Artificial Intelligence to improve the availability of Mission Critical communication

Stan Mulder

Artificial Intelligence to improve the availability of Mission Critical communication

Thesis report

by

Stan Mulder

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on August 25, 2023, at 10:30

Thesis committee:	Dr.ir. Eric Smeitink,	TU Delft, KPN, chair
	Prof.dr.ir. Fernando Kuipers,	TU Delft
	Edwin Bron,	KPN
Supervisors:	Dr.ir. Eric Smeitink,	TU Delft, KPN
	Edwin Bron,	KPN
	Dr. Jacob Groote,	KPN
Place:	Faculty of Electrical Engineering, Delft	
Project Duration:	November 2022 - August 2023	
Student number:	5655307	

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Faculty of Electrical Engineering · Delft University of Technology

Preface

I am delighted to present this master's thesis, Artificial Intelligence to improve the availability of Mission Critical communication. This research marks the end of my studies at the Delft University of Technology for the Master of Science in Electrical Engineering with the specialization track of Wireless Communication and Sensing.

The motivation for this thesis arose from my interest in mobile networks and public safety. With this research, I could delve deeper into this subject and obtain valuable insights in this field. In addition to my interest with mobile networks and public safety, I have always been interested in the potential of artificial intelligence (AI) to address real-world challenges. This research makes it possible to look at a real-world problem KPN is facing.

I would like to express my deepest gratitude to my thesis supervisors, Edwin Bron and Eric Smeitink, for their invaluable guidance, support, and expert advice throughout this research journey. I would also like to thank Jacob Groote for all the help in the first phase of this research. I am also grateful to Fernando Kuipers for being part of my thesis committee. Last but not least, I want to thank KPN for this opportunity and all colleagues that I worked with for all the help and, most important, the unforgettable times. Without all these people, it would not have been possible to achieve this result in the set period.

I hope this thesis makes a valuable contribution, and I am honored to present this work. I sincerely hope it sparks further interest, discussion, and exploration in this subject.

Stan Mulder
Delft, August 2023

Abstract

In modern society, critical operations, such as emergency response and public safety, rely on communication systems, in this context also referred to as mission critical systems. These systems must meet strict availability requirements, since any failure can lead to severe consequences, including loss of life. Traditionally, dedicated private communication systems, like local Push-to-Talk systems, were used for such critical operations, but there is a growing shift towards utilizing public 4G and 5G mobile networks to achieve better coverage, higher data speeds and more innovative features at a lower cost.

With the increasing complexity and vast amount of data from the network, automation and artificial intelligence (AI) are now becoming essential tools in these communication systems to efficiently manage data, configure networks in real-time, and effectively handle alarms. The use of AI can improve the end-to-end availability of mission critical systems, ensuring communication during critical situations.

The main goal of this research is to investigate whether and how the use of AI can improve the end-to-end availability of mission critical systems, with a specific focus on the Mission Critical Push-to-Talk (MCPTT) system of KPN, which is using the public 4G and 5G network. Currently, the KPN MCPTT system is being used with a relatively limited number of users. However, the vision for MCPTT extends beyond its current implementation, aiming to scale up this service. With an increasing number of users, using automation and AI is essential for optimizing and managing the complexities of this mission critical communication system.

The implementation of AI in the MCPTT system follows a systematic approach, starting with the independent analysis and monitoring of system specific elements. By focusing on these elements and using data such as Call Detail Records (CDR) and log data, insights into the system's behavior can be obtained. Through collaboration with system experts, AI algorithms can be trained to effectively detect anomalies, thereby enhancing the overall availability of the MCPTT system. Looking ahead, the integration of real-time data becomes crucial for proactive monitoring. Establishing a streamlined data pipeline facilitates the flow of real-time information, offering a comprehensive overview of system performance and enabling swift anomaly detection. It is concluded that the monitoring of individual system elements with the use of AI is a first step towards improving the end-to-end availability.

In order to ensure the correct use of AI throughout the complete cycle, it is crucial to look at explainability, safety, and data quality. These points should be included at each stage of the AI process. By ensuring explainability, system experts can gain insights into the decision-making process of the AI algorithms. By using safety mechanisms, potential risks and vulnerabilities can be mitigated. Maintaining data quality is essential to achieve accurate outcomes.

Contents

List of Abbreviations

3GPP	3rd Generation Partnership Project
4G	4th Generation Mobile network
5G	5th Generation Mobile network
ACB	Access Class Barring
AF	Application Function
AI	Artificial Intelligence
AI HLEG	European Union High-Level Expert Group on Artificial Intelligence
AI RMF	AI Risk Management Framework
AIOps	Artificial Intelligence for IT Operations
AIVD	Dutch General Intelligence and Security Service
APN	Access Point Name
ARP	Allocation and Retention Priority
CDR	Call Detail Record
E2E	End-to-End
eMMB	enhanced Mobile Broadband
ENISA	European Union Agency for Cybersecurity
GBR	Guaranteed Bit Rate
GCSE	Group Call System Enablers
GDPR	General Data Protection Regulation
HIC	Human-In-Command
HITL	Human-In-The-Loop
HOTL	Human-On-The-Loop
HSS	Home Subscriber Server
IMS	IP Multimedia Subsystem
KPI	Key Performance Indicator
LTE	Long Term Evolution
MB	Metro Bridge
MC	Metro Core
MCPTT	Mission Critical Push-to-Talk
ML	Machine Learning
MLOps	Machine Learning Operations
MME	Mobility Management Entity
mMTC	massive Machine-Type Communications
MSC	Mobile Switching Center
MTTR	Mean Time to Repair
NCSA	National Communications Security Agency
NIST	US National Institute of Standards and Technology
NLP	Natural Language Processing
NOC	Network Operation Centers
NS	Dutch Railways
NSA	Non-Stand Alone
NSI	Near Silent Issue
OTN	Optical Transport Network
OTT	Over-the-Top
PCRF	Policy and Charging Rules Function
PDN-Gw	Packet Data Network Gateway
ProSe	Proximity Services
PTToC	Push-to-Talk over Cellular
QCI	Quality of Service Class Identifier
RCA	Root Cause Analysis
RDI	Dutch National Department for Digital Infrastructure
S-Gw	Serving Gateway
SA	Stand Alone
SLA	Service Level Agreement
SMSC	Short Message Service Center
SPoF	Single Points of Failure
TETRA	Terrestrial Trunked Radio
UE	User Equipment
URLLC	Ultra-Reliable Low-Latency Communication
VoLTE	Voice over LTE
XAI	Explainable AI

List of Figures

List of Tables

1

Introduction

In today's society, critical operations such as emergency response and public safety depend on communication systems, also referred to as mission critical communication systems. Mission critical communication systems have become indispensable, and these systems demand an extremely high level of availability to ensure uninterrupted use of critical services. Currently, there is a shift going from dedicated mission critical communication systems to the use of mission critical over the public 4G and 5G networks. This increases the complexity, scale, and dynamic nature of these systems. This results in significant challenges in maintaining the high availability with constantly changing threats, faults, and vulnerabilities.

To overcome some of these challenges, artificial intelligence (AI) can be used as a tool to revolutionize the way systems are monitored. By using the power of AI algorithms, it becomes possible to proactively identify and mitigate potential disruptions and thereby improving the availability of mission critical systems.

The use of AI has the potential to improve the availability of mission critical systems, but it also introduces risks that cannot be ignored. In the context of mission critical systems, malicious actors can exploit the presence of AI algorithms to manipulate network operations, thereby introducing significant threats to their availability. Attacks targeting AI models within these systems can alter the characteristics of the models, leading to incorrect decisions and outages. Which, in the context of mission critical systems, can lead to loss of lives.

To address these risks, it becomes crucial to incorporate robust mitigation strategies when using AI within mission critical systems. One approach is Explainable AI (XAI), which allows operators to gain understanding of how AI models reach their outcomes. By enabling this interpretability, XAI enables operators to detect anomalies and identify potential attacks. The integration of XAI not only makes AI mission critical systems more secure, but it also makes them more accountable and trustworthy. With the ability to understand the decision-making processes of AI models, operators can identify differences between expected and observed outcomes.

This thesis aims to research whether and how the application of AI in mission critical systems can improve the end-to-end availability, while keeping a focus on explainability and safety. The primary focus of this research lies in the practical implementation of the Mission Critical Push-to-Talk (MCPTT) system, which currently operates within a major corporation, which has approximately one hundred users. As a result, there exists a scarcity of data relating to this system, including limited log data that can be employed to explore the utilization of AI. But it is expected that the MCPTT system will undergo future scaling, making the system more complex. By utilizing the current MCPTT system as a real-world case study, this thesis intends to bridge the gap between theoretical concepts and practical implementation.

To accomplish this, an analysis of the MCPTT system will be conducted and the use of AI algorithms with the available data will be implemented. This research will place significant focus on the aspects of explainability and safety, acknowledging their importance in mission critical systems. Despite the limitations imposed by the limited data availability, this thesis will use the existing data to explore the possibilities of AI within the MCPTT system.

The central research question in this study is formulated as follows: *'How can AI be applied in the Mission Critical Push-to-Talk system to increase the availability of the system?'* This research question sets the foundation for an examination of the specific challenges and opportunities presented by AI implementation in the MCPTT context, aiming to identify mechanisms that can improve the overall availability of this mission critical communication system. To investigate the use of AI inside the MCPTT system, this thesis addresses the following sub-questions:

1. How is the MCPTT system implemented within the network of KPN?
2. Which elements in the system can influence the availability?
3. Where in the MCPTT system can AI be implemented, where is the highest potential to improve the availability?
4. What type of AI models are most suitable to improve availability of the MCPTT system?
5. How can AI systems be designed to mitigate risks, and ensuring security and explainability?
6. How to measure the impact of AI in network management?
7. How to best implement AI in the MCPTT system of KPN?

This thesis consists of nine chapters. Beginning with an introduction in chapter 1, chapter 2 introduces the aspects of mission critical communication. The implementation of MCPTT of KPN, together with the definition of availability and the implementation of priority mechanisms is discussed in chapter 3, after that the fundamentals principles of AI are provided in chapter 4. Chapter 5 delves into the critical aspects of explainability, security, and risks associated with the utilization of AI in mission critical systems. Chapter 6 investigates how telecommunications companies can use AI in their network management. Chapter 7 focuses on the current implementation of AI within KPN for network management, which lays the foundations of chapter 8, which conducts, with the limited amount of data that is available, a first analysis of log files to discover the first findings about how AI can be implemented in the MCPTT system. Chapter 9 presents the concluding remarks of the research, summarizing the key findings and presenting recommendations for the implementation of AI in the MCPTT system. It also identifies possibilities for future research.

2

Mission Critical communication

Mission critical communication systems are characterized by their requirement for availability, as any failure or interruption can have severe consequences. This section explores the concept of mission critical communication systems, and looks into the significance of these systems.

2.1. What is Mission Critical

Communication systems have become indispensable in life, also for businesses and governments. Businesses and governments are relying on these systems for their critical operations, and the unavailability of a system can cause serious problems for these organizations. It is necessary that businesses and governments are able to access a robust communication system that is always available. These systems are called “Mission critical” and “Business Critical” communication systems. Mission and business critical communication are used in different sectors, industries and businesses, but the most crucial case is public safety. Public safety asks for a specific set of requirements that have different features than normal communication systems. Such features are: design, target audience, communication availability, impacts in case of failure as shown in table 2.1 [1].

Table 2.1: Specification of an MC and BC system [1]

Specifications	**Business Critical**	**Mission Critical**
Design	Profitability, competitiveness and sufficient reliable	Redundant, with focus on maximum availability. Priority and preemption must be used
Target audience	Big commercial/industrial organizations	Public safety actors and specific organizations
Availability	Under reasonable or normal critical conditions	At all times
Impacts in case of failure	Operations termination with economic value at stake	Risk of live and important material damage

There is no clear standard definition on mission and business critical systems. In this thesis, the following definition of mission critical used: “A mission critical system is one whose failure or disruption would cause an entire operation or business to grind to a halt. It is a type of task, service, or system that is indispensable to continuing operations [2]”. Business critical is described as a system whose failure causes significant tangible or intangible loss of business or damage to reputation. Unavailability of a system frequently leads to service interruptions. Business critical only refers to a single company or organization. In practice, the term mission critical is often used for both definitions.

Traditionally, mission critical systems for public safety were created from government initiatives which were specifically designed for availability and security with, for example, the TETRA (Terrestrial Trunked Radio) standard which is used to build the C2000 network in The Netherlands. The capacity and throughput of these systems are not comparable to the new generation mobile

networks [3]. These new generation mobile networks (i.e., 4G and 5G) were originally designed with commercial applications in mind. However, they are now proving to be increasingly usable for mission critical purposes due to the standardization of mission critical in the 3rd Generation Partnership Project (3GPP) standard [4]. Mission critical systems should be more robust and available than normal mobile communication systems.

2.2. Mission Critical Push-to-Talk

There are different types of mission critical communication, such as Mission Critical Data, Mission Critical Video and Mission Critical Push-to-Talk. Before these systems were defined in a 3GPP standard, there were different types of systems. First there was the TETRA standard, which were specially designed for safety and security and is used by public services such as the police. Next, the Push-to-Talk over Cellular (PTToC) standard was defined. This standard can be seen as an Over-the-Top (OTT) application, the devices could directly communicate with each other over the mobile network without any quality assurances or priority in the system. MCPTT is based on this idea of PTToC, but it has additional features, ensuring data priority and quality of service, making it suitable for critical applications. MCPTT is defined in 3GPP release 13 and later improved in releases 14 and 15 [5].

MCPTT enables quick voice calls to specific individuals, predefined groups, or all users. It is also possible to send images and text messages. MCPTT can be used in emergency response and critical situations where traditional voice is not sufficient. Release 13 introduced several features such as Group Call System Enablers (GCSE) for dynamic group communications, such as one-to-many calling, and Proximity Services (ProSe) enabling direct device-to-device communication when network signal is unavailable, ensuring high availability. Other supported features include various call types, groups, dynamic regrouping, late entry to groups, floor control, priority-based override, and real-time location capabilities. Release 14 introduced better security features, providing better encryption for MCPTT systems, and release 15 added support for interworking with older systems such as TETRA [4].

Figure 2.1 shows the logical architecture of a MCPTT communication system according to the 3GPP standard. The connection between the mobile core and the User Equipment (UE) is done by the e/g-NodeB. The e/g-NodeB is the base station in the Long Term Evolution (LTE) or 5G network. The mobile core is the central part of the is responsible for the communication. The e/g-NodeB set up and maintains the radio connection with the UE, while the core network components handle the routing of data and signaling messages between the UE and external networks. There is a clear separation between data and signaling messages. The control plane is responsible for handling signaling messages and control information, while the user plane deals with actual user data transmission.

The user plane starts from the UE through the e/g-NodeB, Serving Gateway, PDN Gateway, and finally reaches the MCPTT server. The control plane function is used for setting the appropriate parameters and registering the UE with both the mobile network and the MCPTT server, ensuring service quality and seamless connectivity. The mobile core consists of numerous components:

- Mobility Management Entity (MME): The MME is responsible for managing the mobility of UE devices. It handles tasks such as tracking the UE's location, authenticating and authorizing users, and managing handovers between different e/g-NodeBs.
- Home Subscriber Server (HSS): The HSS serves as the main user database, storing subscriber information, service profiles, and authentication credentials.
- Serving Gateway (S-Gw): The Serving Gateway is responsible for routing and forwarding user data packets to and from the UE. It acts as the anchor point for the user plane during the UE's active session.

- Packet Data Network Gateway (PDN-Gw): The PDN-Gateway serves as the interface between the LTE network and external networks, such as the internet. It handles the UE's IP address allocation and manages the mobility of the UE when moving between different networks.
- Policy and Charging Rules Function (PCRF): The PCRF is responsible for policy control and charging in the network. It applies QoS policies and manages how network resources are allocated to different services, including MCPTT.
- Short Message Service Center (SMSC): The SMSC is responsible for handling SMS messages in the LTE network. The SMSC is employed for user registration and management within the platform.
- Mobile Switching Center (MSC): The MSC handles the SMS delivery to the UE.

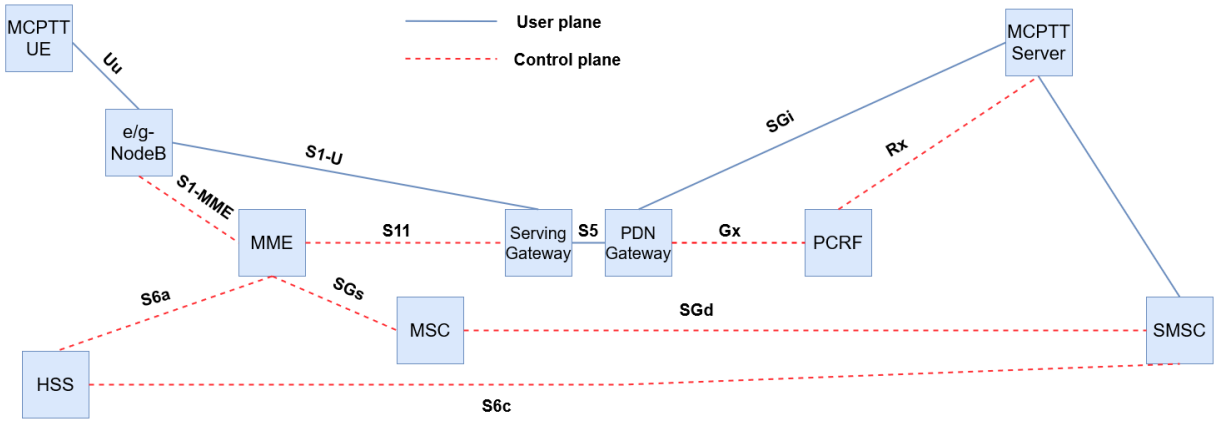

Figure 2.1: Logical architecture of MCPTT with user plane and control plane

2.3. Performance and availability requirements for MCPTT

Two concepts that are often used to characterize the dependability of a system are system availability and system reliability. Availability measures the ability of a system to be ready for immediate use. Reliability measures the ability for a system to perform its function in a specific time. For example, airplanes must be reliable but aren't flying 24 hours a day, so they are not always available. In terms of MCPTT systems, the availability is the most important factor, while the components that are used in the system must be reliable [6].

MCPTT is specifically designed for emergency services and public safety and requires another performance when looking at availability. Public safety users need seamless group communication to carry out their missions, so MCPTT must guarantee successful group calls and immediate talk burst transfer to avoid communication delays. To meet these requirements, MCPTT relies on technologies such as Quality of Service Class Identifier (QCI) values, Access Class Barring (ACB) and Allocation and Retention Priority (ARP) [5].

QCI values are used for every individual bearer in LTE and 5G. It defines the resource type (Guaranteed Bit Rate (GBR) or non-GBR), latency target, packet loss rate and which level of priority the bearer gets. Congestion at the signalling level, which happens when too many User Equipment (UE's) try to get a radio resource, is mitigated by ACB. ARP prevents congestion when the radio resource is already assigned. There are different levels of priority, in case of congestion an existing lower priority bearer can be preempted by a higher priority bearer, and in case of congestion at the signalling level, UEs can be blocked from accessing the network [7].

Figure 2.2 shows a general MCPTT communication scenario within a mobile network. It consists of a MCPTT UE and a MCPTT server. The mobile network, which is generalized in as one cloud, provides dedicated bearers for voice and signalling, each with predefined QCI parameters for specific traffic types. Communication flows through these data bearers, and the MCPTT server communicates with the mobile network to request the necessary QCI parameters for the data bearers used in MCPTT communication.

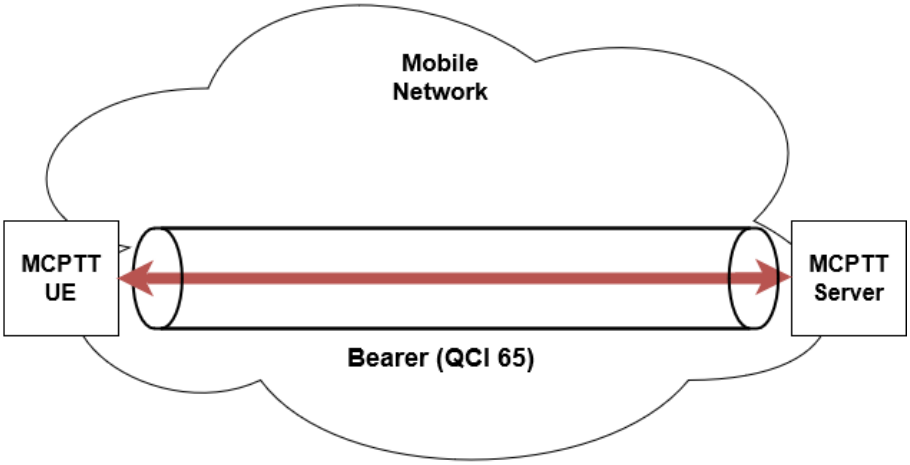

Figure 2.2: MCPTT user plane bearer

The 3GPP specification TS 23.179 specifies how the UE connects to a designated Access Point Name (APN) for MCPTT services. Different applications have different bearers according to the application QCI. Mission critical applications have the highest priority in the system according to their QCI [8]. Table 2.2 shows the requirements for all QCI values. In MCPTT, the communication signals are sent through SIP and HTTP with non-guaranteed bit rate bearers and specific QCI values. SIP utilizes QCI 69 and HTTP QCI 8 or higher. Media transmission is secured through RTP, either via unicast or multicast GBR bearers, with QCI 65 applied. Floor control is also performed on the same bearer.

Table 2.2: QoS class identifiers [8]

Class	Resource Type	Priority	Delay Budget	Error Loss	Example Services
1	GBR	20	100 ms	10^{-2}	Conversational voice
2	GBR	40	150 ms	10^{-3}	Conversational (streaming) video
3	GBR	30	50 ms	10^{-3}	Real time gaming
4	GBR	50	300 ms	10^{-6}	Non-Conversational Video
65	GBR	7	75 ms	10^{-2}	Mission Critical user plane Push to Talk voice
66	GBR	20	100 ms	10^{-2}	Non-Mission Critical user plane Push to Talk
75	GBR	25	50 ms	10^{-2}	Vehicle to everything
5	non-GBR	10	100 ms	10^{-6}	IMS signaling
6	non-GBR	60	300 ms	10^{-6}	Buffered video, TCP-based (www, email...)
7	non-GBR	70	100 ms	10^{-3}	Voice, streaming video, gaming
8	non-GBR	80	300 ms	10^{-6}	Buffered video, TCP-based (www, email...)
9	non-GBR	90	300 ms	10^{-6}	Buffered video, TCP-based (www, email...)
69	non-GBR	5	60 ms	10^{-6}	Mission Critical delay sensitive signalling
70	non-GBR	55	200 ms	10^{-6}	Mission Critical Data
79	non-GBR	65	50 ms	10^{-2}	Vehicle to everything
80	non-GBR	66	10 ms	10^{-6}	Low latency eMBB, augmented reality
81	Delay Critical GBR	11	5 ms	10^{-5}	Remote control
82	Delay Critical GBR	12	10 ms	10^{-6}	Intelligent transport systems
83	Delay Critical GBR	13	20 ms	10^{-5}	Intelligent transport systems
84	Delay Critical GBR	19	10 ms	10^{-4}	Discrete automation
85	Delay Critical GBR	22	10 ms	10^{-4}	Discrete automation

The 3GPP standard suggests three Key Performance Indicators (KPIs) to ensure the quality of MCPTT. Each KPI has a time threshold and a likelihood ratio that must be met, these are shown in table 2.3. Figure 2.3 illustrates the KPIs. The KPIs are as follows [9]:

- **KPI 1 - MCPTT access time:** The access time is the time between the moment a user presses the speak button on the MCPTT UE and receives a signal indicating the user can speak.
- **KPI 2 - End-to-end MCPTT access time:** The E2E access time is comparable to KPI 1 MCPTT access time, but it includes more elements. This includes not only the MCPTT call establishment, but also the potential need for an acknowledgement from the first receiving user before voice data packets can be transmitted.
- **KPI 3 - Mouth-to-ear latency:** The Mouth-to-ear latency refers to the interval between when a transmitting user speaks and when the voice is played back at the receiving user's speaker.

Table 2.3: Threshold and QCI for MCPTT according to the 3GPP standard [9]

MCPTT KPI	Threshold	Likelihood	QCI
KPI 1	<300ms	95%	69
KPI 2	<1000ms	N/A	69
KPI 3	<300ms	95%	65

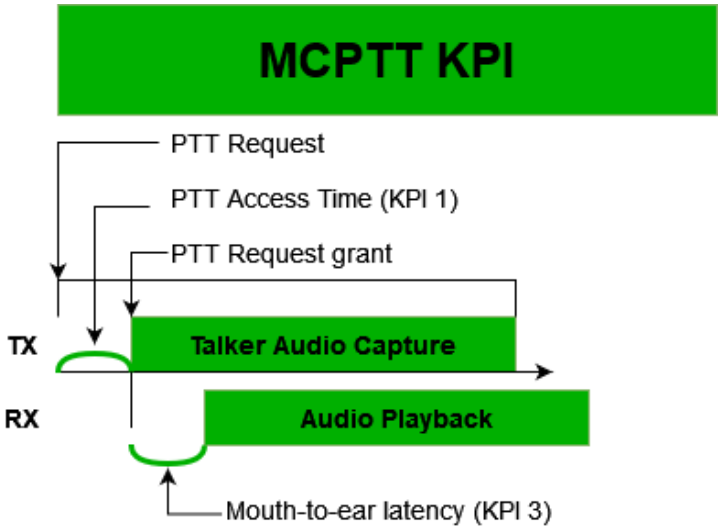

Figure 2.3: The Key Performance Indicators for MCPTT [9]

In the telecommunications industry, availability is typically expressed as a percentage. The golden standard for availability is known as the five-nines, or 99.999% availability. This is the end-to-end (e2e) availability in the viewing point of the end-user. This means that a system can experience a maximum downtime of 316 seconds per year [10]. Achieving five-nines availability is also a primary objective for mission critical applications. With this level of availability, mission critical customers can confidently rely on the system to be consistently accessible whenever required. Reaching 100% availability is not realistic due to the substantial cost when implementing complete redundancy throughout the system. This five-nines availability makes a nation-wide disruption, involving key components of the MCPTT system, almost unacceptable. Resolving such a major disruption within a 5-minute timeframe is nearly impossible. To reach this level of availability, the implementation of AI becomes necessary.

In order to ensure uninterrupted availability, systems are under constant 24/7 monitoring to quickly respond to alarms. However, with the growing complexity of these systems, it has become increasingly difficult for operators to manually monitor the entire infrastructure and identify and address potential outages within a five-minute window. The volume of signals generated daily adds to the complexity and challenge. Therefore, integrating automation, by using artificial intelligence (AI), becomes crucial in effectively managing and resolving such situations.

AI plays a crucial role in preventing and quickly addressing potential disruptions. However, since nation-wide disruptions are rare events, AI systems need to be trained with underlying KPIs and status messages to identify patterns and anomalies associated with failures. By analyzing vast amounts of data, AI can proactively identify potential weak points in the system and predict issues before they escalate into outages.

Another aspect of the high availability for the end-user lies in ensuring that MCPTT calls comply with the performance requirements. Disruptions in signalling messages or an overload of regular, non-MCPTT, traffic, together with suboptimal prioritization settings, can impact the performance of MCPTT calls. To overcome such problems, AI monitoring can be used to handle the large volume of calls and detect any potential performance degradation in real-time. For this, access to real-time data from the mobile core, such as the PCRF, becomes necessary.

Local disruptions can also impact the performance of MCPTT calls. Insufficient indoor coverage or a base station outage can lead to disruption in the critical communications. Here, AI algorithms

can detect local performance degradation, using real-time information from the network, such as Call Detail Records. Quick identification of such issues makes it possible to quickly find and fix the problem. Also, deployment of alternative solutions like a network-in-a-box, providing complete core and radio coverage within the affected area, can be used to quickly provide MCPTT communication in the affected area.

2.4. System monitoring

Preventing errors in a system is not always possible, but these errors can significantly impact system availability. System monitoring plays a vital role in minimizing the mean time to repair (MTTR). KPIs, anomaly detection, Root Cause Analysis (RCA), and human errors are essential aspects of system monitoring.

KPIs are used to monitor system performance and ensure compliance with service level agreements (SLAs). They provide insights into the performance of system entities and are used for anomaly detection. But KPIs often rely on simple threshold values that are not changed to specific environmental conditions [11]. Anomalies are typically detected when a KPI exceeds its threshold, triggering alerts [12].

To identify the underlying causes of errors, RCA is performed. RCA focuses on understanding why, how, and when an incident occurs, aiming to find the source of the problem. It differs from problem-solving, which focuses on corrective actions to resolve incidents. RCA tries to identify all factors that contribute to the problem and establish connections between these events. By understanding the root cause, it becomes possible to know how, when, and why the problem arose [13].

The most unpredictable type of error are human errors, because all organizations work with humans, all organizations are vulnerable. Insufficient knowledge or lack of training can lead to human errors. Systems with poor design, such as those featuring multiple Single Points of Failure (SPoF), are more susceptible to human errors. Automation can help reduce the likelihood of human errors [14].

Although various factors are used for system monitoring to ensure availability, several challenges still exist. Often, simple anomaly detection based on static KPI values is used, no matter external events or changes. Also, manual RCA is a time-consuming process that can be automated [12]. When an error occurs and a KPI threshold is breached, a cascade of events can follow, flooding operations teams with excessive alerts, making it difficult to identify the actual problem.

As mentioned before, with communication systems becoming increasingly complex and interconnected, obtaining a clear system overview becomes challenging, which makes it difficult to response to issues within mission critical systems. Addressing these challenges can be accomplished through automation with the aid of Artificial Intelligence. The next section will focus on the specific implementation of MCPTT within KPN.

3

KPN implementation of MCPTT

This chapter researches three areas of the MCPTT implementation of KPN. First, the MCPTT topology is shown, which delves into network architecture and infrastructure design. Next, the MCPTT availability is mentioned, focusing on ensuring uninterrupted communication. As last, the implementation of the previous mentioned priority mechanism is described. This chapter answers the two sub-questions: 'How is the MCPTT system implemented within the network of KPN?' and 'Which elements in the system can influence the availability?'

3.1. MCPTT topology

MCPTT is implemented by KPN to facilitate direct communication among staff members within specific large corporations, for example the Dutch Railways (NS). MCPTT can be implemented either with or without the use of IMS, in this situation it is implemented without IMS, using the Motorola Kodiak platform, which includes the Application Function (AF) and a SIP server. The AF handles tasks such as call setup, call termination, media handling, and user authentication for MCPTT communication. The SIP server acts as the signaling component for establishing, modifying, and terminating the MCPTT communication sessions.

The UE starts communication through a specialized MCPTT app. The MCPTT app allows users to quickly establish voice and data connections. To facilitate communication, the UE uses the mobile network, which provides the infrastructure for transmitting data and voice. The communication between the UE and the MCPTT server is established using data bearers, which are channels allocated within the mobile network for specific types of data traffic. These data bearers come with predefined QCI parameters, which section 2.2 explained. It is the responsibility of the MCPTT server to request the data bearers, so that the appropriate QCI parameters are set. This is crucial to ensure that the MCPTT communication receives the required priority and resources.

Figure 3.1 shows the MCPTT system, which consists of numerous interconnected elements. The system consists of e/g-NodeBs, which are radio sites spread across the country, in total around 5000 sites. These e/g-NodeBs are directly connected to the Metro Bridge (MB) and Metro Core (MC) nodes. There are 1500 MB nodes and 161 MC nodes in the system. The primary goal of the MC nodes are data transport.

The MC is connected to the Transport Core, via high-speed 10G/100G Optical Transport Network (OTN) connections. The Transport Core provides connectivity for various services, including internet, data center, and mobile core services. There are four locations spread across the Netherlands, ensuring redundancy. As the traffic passes through the nodes, it eventually reaches the mobile core. After crossing the mobile core, the traffic is directed to the Filter Layer, which functions as a connectivity and filtering layer. It handles external connections and internal communications between different zones of the MCPTT platform. The end point is the MCPTT

server, this server is a solution from Motorola that uses the 3GPP MCPTT standard. Figure 3.1 provides an overview of the interconnected elements of this system with a generalized mobile core. According to Motorola, the Kodiak MCPTT server reaches a 99.999% uptime with a distributed and geo-redundant architecture of the platform, spread over 2 locations [15].

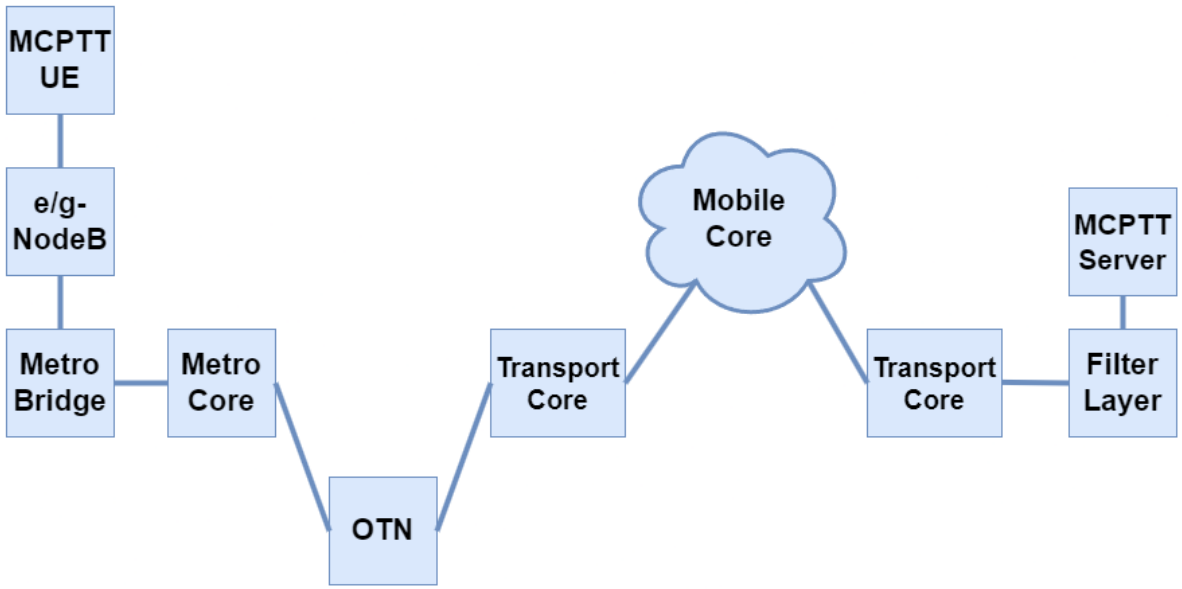

Figure 3.1: Interconnected elements of the MCPTT system

3.2. Availability of the MCPTT system

Within the MCPTT system architecture, there exists a distinction between system specific elements, which include the MCPTT UE and the MCPTT server, and generic elements, which are shared components utilized by various services across the system. Geo-redundancy mechanisms are used for these generic elements, guaranteeing service continuity even in the event of failures. This strategic separation of concerns between MCPTT specific and generic elements allows the MCPTT team to maintain a dedicated focus on their core responsibilities while relying on the generic elements. While the primary responsibility lies with MCPTT specific components, it remains essential to monitor the generic elements, with a focus on their influence on the MCPTT system, such as user plane and control plane traffic, as illustrated in figure 2.1. The monitoring of this traffic enables a full oversight of the MCPTT system.

Previous failures of the MCPTT system demonstrated that there are different elements involved in each failure. This also shows the complexity of the system. There are different interconnected elements, and it can be difficult to quickly find the root cause of the problem also because of external parties involved. This also directly shows the added value AI can have in this system. When analyzing Major Incident Reports of previous failures, it is possible to identify various factors and elements that contributed to the outages. Because these Major Incident Reports are confidential, the details of these reports cannot be described here.

It is important to note that availability should not be viewed solely as a one-dimensional metric. In the context of MCPTT, availability consist of various aspects. For instance, call detail records provide insights into the efficiency of establishing connections and the overall success rate of communication attempts. Local availability focuses on addressing the specific needs of users within areas, ensuring uninterrupted communication in critical locations or during emergencies.

To improve the end-user experience and achieve five-nines availability, it is crucial to have a clear definition of which availability needs to achieve the five-nines. The definition that is used in this research is that in areas with guaranteed coverage, five-nines availability must be assured, measured by the Call Success Rate. The importance of the Call Success Rate as a metric helps to measure the establishing of successful connections. Aiming for five-nines availability in regions with guaranteed coverage ensures an extremely high success rate for call attempts, indicating a robust communication service for mission critical use in these areas.

It is essential to measure the availability of the communication. While there is limited information available regarding MCPTT availability, statistics for VoLTE are more easily accessible. Because of confidentiality, the exact numbers are not mentioned here. In March 2023, the reported LTE availability was a relatively high number. This means that the LTE network was available for almost all the time, according to KPN internal quality reports. This high availability indicates that the LTE network is providing consistent connectivity for various services, including Voice over LTE (VoLTE). When considering VoLTE specifically, two key metrics are commonly used to assess availability: Call Success Rate and Dropped Call Rate. The Call Success Rate for VoLTE was reported as a relatively high number. This indicates that out of all attempted VoLTE call setups, a certain amount was successful, resulting in connected calls. It highlights the high availability and effectiveness of VoLTE in establishing successful connections.

On the other hand, the Dropped Call Rate for VoLTE was relatively high. This metric represents the percentage of VoLTE calls that were successfully completed. With a dropped call rate of this certain number, it suggests that only a miniscule percentage of VoLTE calls encountered premature disconnections. While the reported figures show a relatively high level of availability for VoLTE, it is worth noting that they don't achieve the five-nines availability. Because of differences between VoLTE and MCPTT, such as the priority of MCPTT and the dedicated MCPTT platform instead of IMS, it is not possible to compare the numbers of VoLTE with MCPTT on a one-to-one basis. However, they can provide a general sense of their respective availability levels.

Currently, the availability of MCPTT with respect to the Call Success Rate is not being measured. It is important to continuously monitor this metric to gain an understanding of the systems' availability. The Call Success Rate indicates how the MCPTT service is functioning. By monitoring this number over time, it is possible to identify potential issues, and assess the system's availability. The monitoring of this number ensures that any anomalies or shortcomings are quickly found, leading to improved communication availability.

3.3. Implementation of priority mechanisms

Section 2.2 explained how MCPTT can be given priority in a network using various mechanisms. These mechanisms include Quality of Service Class Identifier (QCI) values, Access Class Barring (ACB), and Allocation and Retention Priority (ARP). To prioritize MCPTT traffic, specific QCI values are assigned. QCI 65 is used for MCPTT voice, while QCI 69 is used for MCPTT signalling data, as defined by the 3GPP standard. These QCI values help differentiate MCPTT traffic from regular data traffic.

Access Class 14 is used for MCPTT, which is originally used for "Emergency Services" according to the standard. In the context of MCPTT, it is used as "High-priority PTT". By using Access Class 14, MCPTT traffic receives priority in the system. ARP is another important factor in giving priority to MCPTT, each bearer gets an ARP value. A priority level of 2 is assigned to a MCPTT bearer, this ensures that, in the case of congestion, a bearer with a lower level will be replaced by a MCPTT bearer, also known as preemption. So MCPTT traffic is given priority over regular traffic.

By combining these methods, MCPTT traffic is guaranteed to have priority in the system, even during congested periods. Normally, voice traffic typically gets priority over data traffic during

congestion. With the implementation of MCPTT, both voice and signalling data and the ability to pre-empt existing connections allows MCPTT to be available during congestion. Figure 3.2 shows the use of these mechanisms in a structured way.

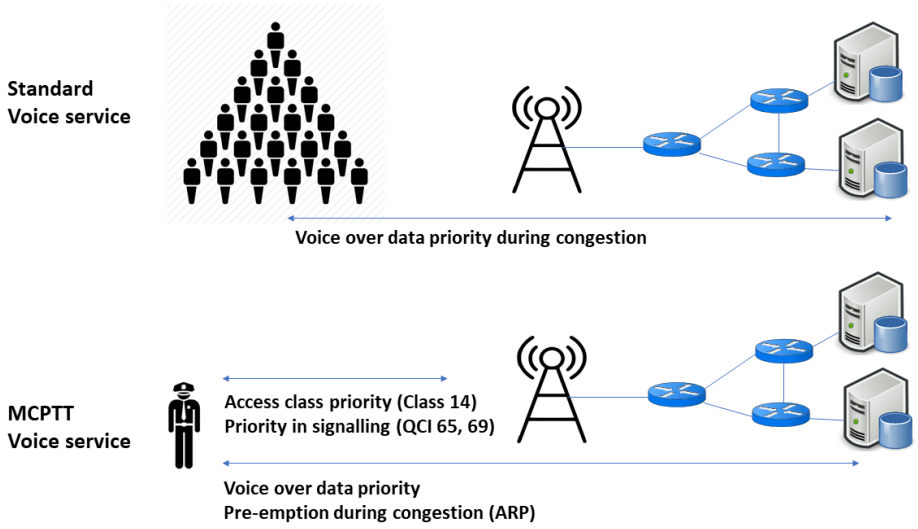

Figure 3.2: Illustration of the three priority mechanisms that are implemented in MCPTT

QCI values are used to distinguish GBR and Non-GBR connections in MCPTT. For each MCPTT call, the MCPTT platform requests a GBR for each user involved in the call. This request is made through the PCRF. The MCPTT platform includes the AF function and utilizes the Rx interface to interact with the PCRF. Through this communication, the MCPTT platform requests GBRs for each user in an MCPTT call. Due to potential delays in the GBR request process, an MCPTT call may begin over the signalling channel and later shift to a dedicated GBR channel when it becomes accessible. This ensures that MCPTT calls can be quickly established while still using the benefits of GBR connections when they are established.

3.4. Conclusions

This chapter provides an overview of how the MCPTT system is implemented within the KPN network and identifies the elements that can influence its availability, addressing the two sub-questions: 'How is the MCPTT system implemented within the network of KPN?' and 'Which elements in the system can influence the availability?.' The network consists of various specific and generic elements, with a clear separation of concerns. Specific priority mechanisms are used to ensure communication during critical situations. The chapter also defines availability definition. Previous failures are analyzed to get an understanding of which elements impact the availability. Also, comparison is made with VoLTE statistics to get a general sense of the availability levels and a recommendation was made to monitor the Call Success Rate of the MPTT system.

4

Artificial Intelligence

This chapter looks at the principles of artificial intelligence, analyzing its various fields, such as machine learning, and highlighting the role played by data quality in using AI algorithms. The objective is to research and identify the most suitable AI algorithms for effectively analyzing the available data from the MCPTT system and answer the research question: 'What type of AI models are most suitable to improve availability of the MCPTT system?' The available data consist of Call Detail Records and log files from the MCPTT server.

4.1. Types of Artificial Intelligence

The difference between automation and AI is crucial to understand. Automation focuses on decisions that are already known, enabling faster execution. On the other hand, AI will help by making a better decision. Tan [16] provides three key factors between AI and automation: the data set, bias, and real-time processing.

- **Data Set:** In automation, it is already known where to look for, and it is possible to accelerate this process. AI is a dynamic process which highlights and prioritizes activities that require attention.
- **Bias:** In automation there is a certain bias in the decisions, when it is known where to look for humans will introduce a bias in the decision. With AI, you are looking for unknowns, this will reduce the bias in the system.
- **Real-time:** AI is a real-time process that will change over time as more data is available. Automation is a static process that is not changed over time, unless human intervention takes place.

Artificial Intelligence consists of various subfields. These are Machine Learning (ML), Neural Networks, Deep Learning, and Natural Language Processing (NLP). Among these, machine learning is the most important in this research, as it enables applications to learn and improve from data without needing explicit human intervention. Figure 4.1 visually shows the relationship of AI, Machine Learning and Deep Learning.

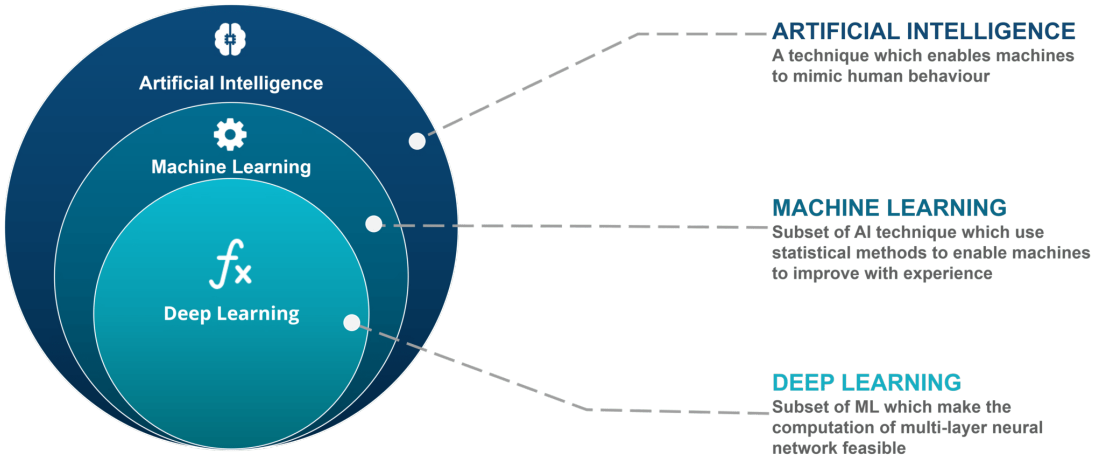

Figure 4.1: Difference between AI, ML and Deep Learning [17]

Machine learning algorithms have a phase of training, during which they learn patterns from data and utilize this knowledge to make decisions. This process is continuous and real-time, as ML algorithms continue to learn from ongoing experiences and collect more data over time. Machine learning has applications in various fields, including fraud detection in financial transactions, real-time language translation, weather prediction, and the improvement of mobile network availability [18].

Machine learning is an analytical subject, using mathematical models to recommend actions. These models help the extraction of knowledge and identification of patterns within the data. ML algorithms are trained using three primary techniques: supervised learning, unsupervised learning, and reinforcement learning. These techniques enable the algorithms to recognize patterns, establish correlations, and develop a deeper understanding of the underlying data structures.

Supervised learning is a machine learning technique where a model is trained using labeled data to make predictions for unseen data. The model is training with a set of input-output pairs, where the correct outputs are provided together with the corresponding inputs, aiming to learn a generalized rule that provides mapping between inputs and outputs [19].

Unsupervised learning consists of training a model on data that hasn't been labeled and focusing on finding patterns or structures in the data. Unlike supervised learning, unsupervised learning does not rely on data with correct outputs. Instead, the algorithm tries to find structures, similarities, or groupings within the data. This learning technique is used in anomaly detection and pattern recognition tasks. The limitation of unsupervised learning is the potential of black box, where it isn't understood how the algorithm reached its outcome [18].

Reinforcement learning is a machine learning technique where an agent provides rewards or punishments to guide the behavior of the model. The objective of this technique is to learn a policy that maximizes long-term rewards. Currently, reinforcement learning remains an active area of research, with potential applications in domains such as supply chain robotics and autonomous vehicles [20].

4.2. Algorithms

Section 4.1 provided an overview of the three machine learning techniques, each driven by different algorithms. Machine learning algorithms use various techniques such as regression, clustering, and classification to analyze data and generate predictions or decisions. Understanding how

these algorithms work and their strengths and limitations is crucial in developing effective machine learning solutions. This section will briefly introduce supervised learning and reinforcement learning algorithms and specific focus will be placed on unsupervised learning algorithms, as they will be used later in this thesis.

4.2.1. Supervised learning algorithms

An important technique for supervised learning is classification, this technique tries to predict the probability of the correct label of the input data based on the training data. The goal is to identify relationships between the input and output and use that relation to classify the new data. The output variable is typically a label or category, such as “spam” or “not spam,” or “dog” or “cat.” The input variables are used to describe the characteristics of the data.

Classification algorithms work by identifying the connection between the input variables and the output variable within the training data. This relationship is then presented into a mathematical model or formula that maps input variables to the output variable. To classify new data, the model uses the input variable values to predict the corresponding output variable. Classification is widely used in industry for image recognition, spam detection and fraud detection [21].

Another technique is regression, this is almost the same as classification but is based on continuous values instead of discrete labels. It predicts the label based on historical data and also the relationship between variables. This helps to predict how variables affect one another [22]. Figure 4.2 illustrates the different outcomes between classification and regression.

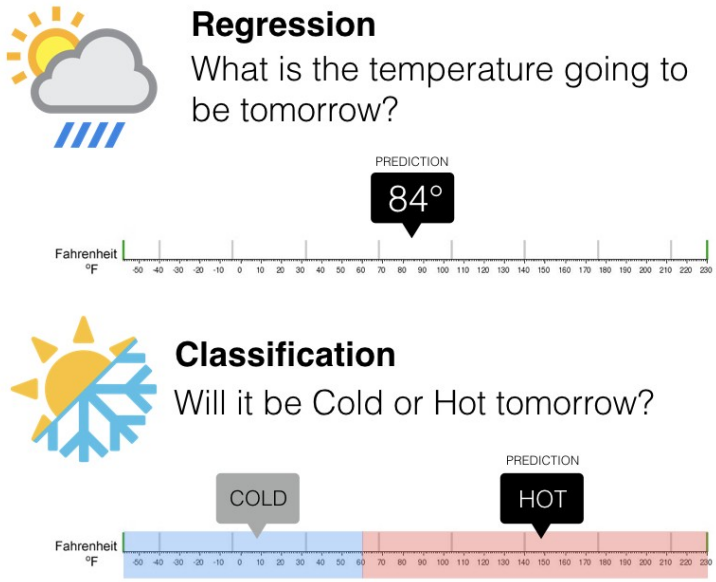

Figure 4.2: Difference of classification and regression [23]

4.2.2. Unsupervised learning algorithms

Techniques that are used in unsupervised learning are clustering and association, which do not rely on predefined labels and try to find patterns and relationships in the data. The goal of association rule learning is to identify patterns of items frequently appearing together in the data, which can be used to make predictions. An application of association rule learning is market basket analysis, where transactions are represented as sets of items, and patterns are searched in these transactions. For instance, in a grocery store’s transaction data, frequently bought items are identified, and candidate item sets are formed. Association rules are generated from frequent

item sets, specifying item correlations. In the end, these rules provide insights, like the likelihood of customers purchasing one item given their purchase of another, helping businesses in product placement and marketing strategies [24].

Clustering algorithms group similar objects together by splitting the data set into subsets. Its goal is to find underlying patterns or structure within a dataset that does not have pre-defined labels. A cluster is formed by looking at the similarity of the data points based on certain criteria. Clustering is an important technique in unsupervised learning for discovering patterns in data, various clustering algorithms, such as K-means, are used [25].

4.2.3. Reinforcement learning algorithm

Reinforcement learning algorithms can be categorized into two main approaches: model-free and model-based learning techniques. In model-free reinforcement learning, the algorithm learns to make optimal decisions through trial and error. It interacts with the environment and receives feedback in the form of rewards or punishments based on its actions. By exploring different actions and observing the consequences, the algorithm learns the optimal decision-making policy. Model-free learns directly from experience and adjusts its behavior accordingly [18].

Model-based reinforcement learning involves building a model of the environment. This model captures the dynamics of the environment, including the possible actions and their associated outcomes. The model can be used to simulate different actions and predict their consequences. By planning and simulating sequences of actions, the algorithm can measure the expected outcomes and make decisions based on this information. Model-based learning can potentially offer more efficient decision-making by using the insights provided by the constructed model, rather than relying entirely on trial-and-error learning [18].

4.3. Data quality

The most important factor in machine learning is the data. With wrong, incorrect or poisoned data, wrong decisions can be made, which can have far-reaching consequences for the availability, particularly in the context of availability of mission critical systems. The performance of a machine learning model is linked to the quality of the data it is trained on. The quote, “garbage in – garbage out”, summarizes this idea. [26].

Numerous issues, such as missing or incorrect values, can be found in real-world data. If these problems aren’t addressed before the data is fed into the machine learning model, it can lead to negative results. When a machine learning model works with such datasets, it can lead to safety issues for mission critical systems [27].

Data quality can be divided into 6 key terms. Completeness, accuracy, timeliness, consistency, validity, uniqueness. These terms are shown in figure 4.3 and are described as follows. Completeness measures the presence of non-null values and ranges from 0% to 100%. Clear explanations should be provided for any missing data. Accuracy verifies that the data accurately represents the reality. Timeliness means the importance of having up-to-date data available when making decisions. Consistency ensures that data remains consistent across various storage locations, software, and formats. Validity ensures that the data fits to predefined limits and is of the correct type. Uniqueness guarantees that features within a dataset do not appear more than once [27].

Figure 4.3: 6 dimensions of data quality [27]

4.4. Anomaly detection

Anomaly detection plays a crucial role in data analysis by identifying abnormal data points within a dataset. Its objective is to find patterns that are infrequent, to discover valuable insights and potential anomalies hidden within the dataset. This task is essential in various domains as it helps in understanding rare events and identifying potential threats. A widely accepted definition by Hawkins is: “An anomaly is an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism” [28]. Figure 4.4 presents a framework illustrating the process of anomaly detection. Anomalies can have critical consequences, since they represent rare events that may lead to mission critical system failures. [29].

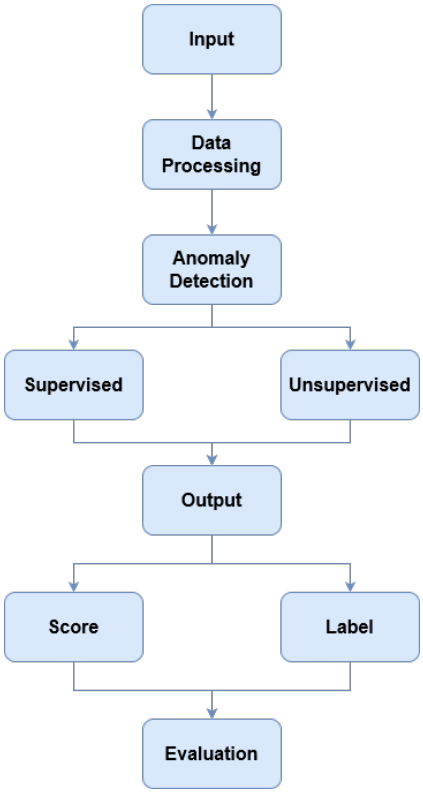

Figure 4.4: Framework for anomaly detection [29]

The primary reason for using AI is to identify unknown issues within a system, particularly in the dynamic context of MCPTT. Fixed thresholds are not suitable due to changing conditions, and manual analysis of frequent logs is time-consuming. AI offers solutions by establishing adaptive thresholds through machine learning, automating anomaly detection in real-time and handling large data volumes. Real-time anomaly detection can be applied to improve availability of mission critical systems

Figure 4.4 shows two types of techniques for anomaly detection, supervised and unsupervised. Supervised and unsupervised anomaly detection are two different approaches used to identify abnormal patterns in data. In supervised anomaly detection, a labeled training dataset with known normal and abnormal data points is required to train the model. The model then uses this information to detect anomalies in new data. However, collecting and labeling the training dataset demands significant manual effort, and the access to sizable training sets can be limited.

Unsupervised learning algorithms can detect anomalies in unlabeled data and use this information to identify anomalies in new data. This approach offers the advantage of effectively uncovering unknown anomalies within the network and also minimizing manual efforts needed for anomaly detection. Unlike supervised learning, where only known anomalies can be labeled, unsupervised anomaly detection avoids bias in the system and can discover unknown anomalies [30].

Considering the specific context of this research on analyzing the available data from the MCPTT system, unsupervised anomaly detection the most suitable choice. The MCPTT system operates in a dynamic and complex environment, making it challenging to predict all possible anomalies. Using unsupervised anomaly detection will enable the system to adapt to anomalies, potentially improving its ability to maintain high availability during critical situations. This approach reduces the need for manual data labeling, making it more practical and efficient for the continuous monitoring and analysis of the MCPTT system.

4.5. K-means

As mentioned in the beginning of this chapter, the available data consists of Call Detail Records and log files from the MCPTT server. For analyzing the available logs from the MCPTT server, K-means clustering is a suitable algorithm for several reasons. The MCPTT server deals with huge amounts of data, with much of unknowns, making unsupervised anomaly detection an appropriate choice. K-means efficiently groups similar data points into clusters without requiring manual labeling. This allows the algorithm to adapt to patterns in real-time. K-means is able to process the large log files generated by the MCPTT server.

However, it's important to note that K-means itself does not make decisions about anomalies. K-means partitions data points into clusters based on their similarity. While K-means is effective at identifying patterns within data, it does not recognize anomalies as being anomalies. Anomalies in log files might not necessarily be the less frequent points. K-means can help in grouping similar data points and present a manageable overview to a system expert, chapter 8.1 will go further into this aspect.

The K-means algorithm aims to put each data point to the nearest cluster based on their similarity. It starts by determining the number of clusters, K, and initializing K cluster centers. Next, each data point is assigned to the cluster whose center is closest to it, based on the Nearest-Neighbor rule. This process is iterative, and after all points are assigned to clusters, the average position of the points within each cluster is calculated to update the cluster centers. The objective is to minimize the sum of distances between data points and their assigned cluster centers [31]. An example of the resulting clustering can be seen in figure 4.5.

K-means clustering has certain limitations that can affect its performance. The algorithm is sensitive to outliers, which can lead to the formation of suboptimal clusters. Another limitation comes from the requirement of specifying the number of clusters, K, in advance, although in many cases, the optimal number of clusters is unknown. Lastly, K-means is only useful to numerical data and cannot handle categorical data. Despite these limitations, K-means is used in various domains due to its simplicity and effectiveness [32].

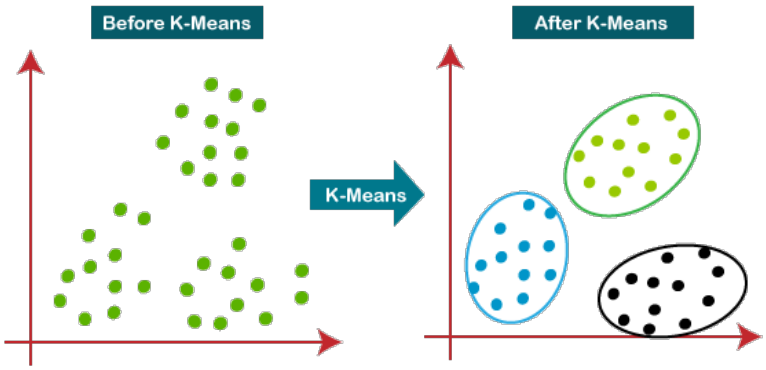

Figure 4.5: K-means algorithm [33]

4.6. Isolation Forest

In the context of Call Detail Records (CDRs) analysis, the Isolation Forest algorithm is a suitable approach for anomaly detection. Because of the dynamic nature of the MCPTT system, the types of anomalies in CDRs are not known in advance, which is why unsupervised learning is necessary. There can be an assumption made that anomalies in CDRs are not frequent. The Isolation Forest algorithm can identify anomalies in data. This algorithm does not require labeled

data, so it is suitable for detecting unusual call patterns, or abnormal behaviors in the CDRs. The Isolation Forest algorithm is able to identify anomalies and adapt to changing patterns in the CDRs. This makes it a valuable algorithm for ensuring the high availability of the MCPTT system.

The Isolation Forest algorithm, or iForest, is suitable for high-dimensional data that has a substantial number of irrelevant attributes. The principle of iForest involves constructing an ensemble of iTrees to a given dataset. Anomalies are detected because of their relatively short average path lengths within the iTrees. The algorithm relies on only two variables: the number of trees to build and the subsampling size. It's important to highlight that each tree within the ensemble, which is a predetermined variable, corresponds to a decision involving an attribute and a split value [34]. The accuracy of the Isolation Forest algorithm improves as the number of iTrees grow because, the algorithm can better differentiate anomalies from normal data points, resulting in improved accuracy and more reliable identification of anomalies. However, increasing the number of iTrees also comes with a trade-off in terms of computational resources.

During the creation of each iTree in the iForest, the algorithm randomly selects attributes and splitting points to divide the data points. Since anomalies are a relatively small proportion of the overall dataset, they are likely to have attribute values that differ significantly from those of normal data points. When the algorithm splits the data points, anomalies are more likely to get separated into their own branches early in the tree's construction. This is because the algorithm chooses random attributes and splitting points, which can often create partitions that capture anomalies efficiently. As a result, anomalies end up closer to the root nodes of the trees, while normal data points are spread out further away from the root nodes [35]. Figure 4.6 shows an example of such an iForest with anomalies, that are close to the root of the tree, and potential anomalies.

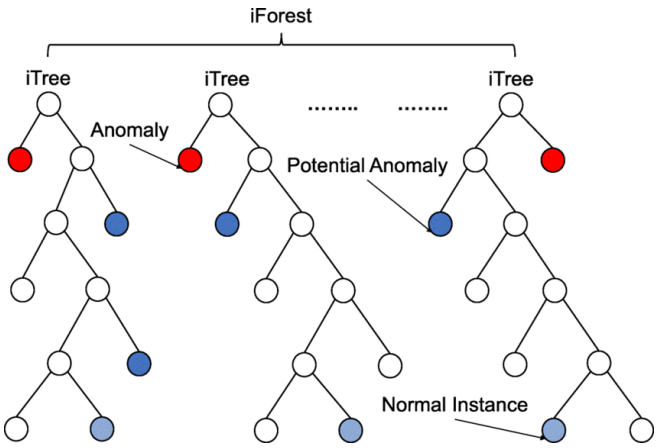

Figure 4.6: Example of an iForest [36]

4.7. Conclusions

The main goal of this chapter was to investigate the most appropriate AI algorithms for analyzing the available data from the MCPTT system, which consist of CDRs and log files. The research question addressed in this chapter was: 'What type of AI models are most suitable to improve availability of the MCPTT system?' The chapter provided an overview of various AI algorithms and looked at the crucial role of data quality. The conclusion is that unsupervised anomaly detection is the best approach for the MCPTT system. Specifically, the K-means algorithm for analyzing the MCPTT server logs, and the Isolation Forest algorithm for the CDR data. By utilizing these unsupervised methods, the MCPTT system can effectively identify abnormal patterns, and ultimately improve its overall availability. These 2 algorithms will be used for anomaly detection in chapter 8.

5

Security, Risks and Explainable AI

The Dutch National Department for Digital Infrastructure (RDI) has presented various risks associated with the implementation of AI algorithms. The primary areas of these risks include the potential for abuse of autonomous systems, the lack of explainability in AI decision-making processes, and the issue of assigning responsibility when things go wrong. Addressing these risks requires careful consideration of the design, transparency, and accountability of AI systems, aiming to establish trust and confidence in their applications. By proactively addressing these challenges, operators can maximize the benefits of AI while minimizing potential negative consequences [37].

The main focus of this chapter is to research the security, risks, and explainability aspects of AI systems. It aims to address the potential vulnerabilities and risks associated with implementing AI systems while ensuring explainability in the decision-making processes. This overview can be used for future implementation of AI in the MCPTT system. The central objective is to investigate and identify ways to design AI systems that effectively mitigate risks and ensure security and explainability. This chapter answers the sub-question: 'How can AI systems be designed to mitigate risks, and ensuring security and explainability?'

5.1. Security issues

In the context of mission critical systems, it is important to look at the security aspects of AI systems. AI systems rely on input data for the decision-making process, which introduces vulnerabilities, as manipulation or bias in the data can significantly impact system outcomes. In the context of mission critical systems, such manipulation can lead to severe consequences. Therefore, understanding the potential risks associated with AI manipulation is crucial.

The European Union Agency for Cybersecurity (ENISA) warns about the abuse of AI by 2030. While unconscious bias in AI is a known concern, intentional manipulation of AI algorithms and training data is becoming as real threat. Corrupted training data can result in incorrect decisions, particularly in high-risk sectors. Threat actors may attempt to exploit AI applications to influence decision-making outcomes or gather information about potential victims [38].

Although there is currently no evidence of such attacks happening, a group of researchers from Google, ETH Zurich, NVIDIA, and Robust Intelligence predict that these attacks will certainly occur in the near future [39]. The objectives of attackers can include espionage, sabotage, and fraud, with attacks occurring during both the training and production stages of AI systems. The National Communications Security Agency (NCSA) of the Dutch General Intelligence and Security Service (AIVD) classifies five types of attacks: Evasion, Inference, Poisoning, Backdoor, and Inversion [40].

Evasion attacks are the most common types of attacks on machine learning models and typically occur after the model is implemented. These attacks involve creating inputs that appear normal

to humans but are misclassified by the ML model. For instance, changing certain pixels in an image before uploading it can cause an image recognition system to misclassify it [41].

Inference attacks are another common type and aims to gather system information for potential future use. There are two main inference models: membership inference and input inference. Membership inference retrieves information about the presence of a specific example in the dataset, which can be used to prepare evasion attacks. Input inference retrieves a specific example by knowing its label. For example, it may be possible to retrieve a photo of a person by only knowing their name, raising privacy concerns [41].

Poisoning attacks involve injecting malicious data into the system, leading to reduced accuracy, misclassification, and unreliable output. These attacks can occur before, during, or after the model training phase, and they can be executed through various means, such as injecting falsified data, manipulating existing data, or disrupting the labeling process. The goal is to let the model make incorrect conclusions. To carry out a poisoning attack, the attacker must have write permissions to at least some parts of the data [40].

Backdoor attacks involve the insertion of a backdoor into an AI model, allowing external parties to influence the model's decision-making process. This means adding a branch to the decision tree that determines the final decision for specific inputs. Backdoors can exist in models that have not been developed internally, such as pre-trained models. Detecting an implemented backdoor in a model can be extremely challenging, if not impossible [40].

The last type of attack is an inversion attack. This type of attack aims to reconstruct the dataset that was used to train the model. This data may contain sensitive information that could be of interest to an attacker, who may use this information for various purposes, such as stealing intellectual property or identifying weaknesses in the model. To carry out an inversion attack, an attacker sends numerous queries to the model to find out how it operates. Once the attacker has a good understanding of how the model functions, they can run the model in their own environment and search for vulnerabilities using input attacks. Additionally, the attacker may test how the model responds to specific attacks to prepare a countermeasure for that response. Unfortunately, these types of attacks are very difficult to detect [40].

Figure 5.1 provides an overview of the different attack types that can occur at various stages of the ML model's lifecycle. This figure emphasizes the importance of considering the risks and potential attacks throughout every phase of the ML model, including development, training, and production [42].

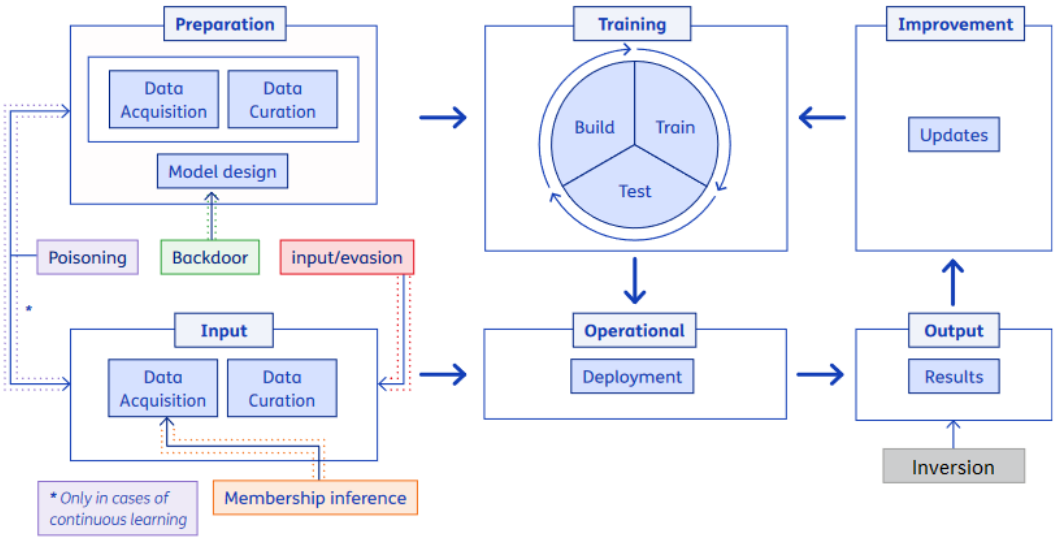

Figure 5.1: Attack possibilities in the ML cycle [42]

In most attacks, the attacker needs knowledge about the systems. The amount of knowledge an attacker has can be classified into four types, black-box, gray-box, white-box and transparent-box. With a black-box the attackers has zero of knowledge about the architecture, model and data. With a gray-box, the attackers have limited knowledge about the model or the training data. With a white-box, the attackers have all knowledge about the model, architecture, and training data. Finally, with the transparent-box, the attacker has the same knowledge about the system as with a white-box, but they also have knowledge about the defensive measures to counter an attack [43].

Now that the attacks are clear, it is important to know what to do to protect the AI systems against these attacks. The NCSA has come up with five principles for protecting the AI systems. Ensuring the quality of the datasets, considering validation of the data, taking supply chain security into account, making the model is robust against attacks and making sure the model is auditable [40].

Ensuring dataset quality is crucial during AI model development. The data should be organized, reliable, and protected against tampering. It is also important to identify elements in the dataset that could negatively affect model performance. By closely monitoring data quality, it is possible to enhance the model's performance and prevent attacks like poisoning and input inference [40].

Validating data becomes crucial when external sources are used. Since the creation process of the dataset may be unknown, validating the data's authenticity and reliability is crucial. Being overdependent on a single source should be avoided, and continuous monitoring of external data is necessary to mitigate risks [40].

When using pre-built models, there can be vulnerabilities in the system. To counteract backdoor attacks, understanding the model is essential. Developing a model in-house makes it more difficult for attackers to introduce backdoors. However, if using external models becomes necessary due to data or computing limitations, trust in the external party must be established through appropriate safeguards [40].

The robustness of an AI model refers to its ability to function properly when there are anomalous

inputs, data changes, or attempted abuse. It's important to train the model against attacks. Adversarial training can help the model recognize and resist modified data. There can also be a detection mechanism to find manipulated input to protect the model and training data from attacks. Creating exercises on the AI model, where a 'red team' tries to discover and exploit vulnerabilities in the model, can also provide insight into the security weaknesses [40].

Often it is unclear how the AI reached a certain prediction. By considering the explainability of the model when building and training it, it is possible to make it less of a black box. This is known as Explainable AI (XAI). For simple models, it's often easy to understand why certain choices are made. For more complicated models, this can also be difficult. Understanding how the model works makes it possible to develop controls and test cases [40]. As more critical systems are relying on AI, it is important to keep the possible risks of the system in mind. Especially with an unsupervised learning algorithm where little is known about the systems or with an AI algorithm from a third party.

5.2. Risks

The US National Institute of Standards and Technology (NIST) has released an AI Risk Management Framework (AI RMF) to mitigate the potential negative impacts of AI systems. Within the AI RMF, risk is defined as the measure of the likelihood and consequences of an event. The consequences of AI can be both positive and negative, leading to opportunities or threats. Risk is quantified by considering the magnitude of harm and the likelihood of occurrence. Effectively managing AI risks can ultimately contribute to the development of more trustworthy systems. There are three main challenges in the process of managing risks: risk measurement, risk tolerance, and risk prioritization [44].

AI risks lack clear definitions, making the measurement of AI risks or failures a challenging task. The use of third-party data, software, or hardware in research and development can accelerate progress but also complicate risk measurement. Risks can come from the third-party components themselves as well as their use. Also, there may be a mismatch between the risk metrics and methodologies used by those developing the AI system and those deploying or operating it. Identifying risks at different stages of the AI lifecycle can differ, and different stakeholders across the AI lifecycle may have other risk perspectives. Working with a baseline for comparing numbers in AI systems that interact with or replace humans, such as in system monitoring, can be difficult to determine [44].

Risk tolerance is not the same across organizations, especially in the context of mission critical systems. The level of risk tolerance depends on the operational context and can be influenced by various factors, such as legal requirements. As AI systems continue to develop, alongside policies and norms, the levels of risk tolerance are expected to change. Organizations may have different degrees of risk tolerance based on their specific needs and customer expectations [44].

The prioritization of risks may change, based on whether an AI system engages directly with humans or not. When dealing with AI systems trained on sensitive or protected data or those that produce outputs with direct or indirect consequences for humans, a higher level of prioritization may be required. It is important to document and address residual risk, which is the risk that exists even after risk mitigation measures are implemented. This documentation is crucial to inform end-users about potential negative impacts associated with interacting with the AI system. By being transparent about residual risk, it can be ensured that users are informed and can make informed decisions regarding their engagement with the AI system [44].

In practice, trying to completely eliminate negative AI risks can be counterproductive, since it is unrealistic to prevent all incidents or failures. Instead, organizations can introduce a risk

management culture that recognizes the need for varying levels of attention and resources for different AI risks. Effective AI risk management involves guidelines to measure the trustworthiness of AI systems and assigning resources based on the level of risk and potential impact. By using this approach, organizations can make informed decisions about risk mitigation, creating a balance between risk reduction and practical considerations. This helps responsible and effective deployment of AI technologies while admitting that a certain level of risk may exist [44].

5.3. Explainable AI

Explainable AI (XA) refers to AI systems that are able to provide clear and understandable explanations for their decision-making processes and predictions. This is important because it allows stakeholders to understand how the AI system arrived at its conclusions, which can help to increase trust and transparency in the technology. Section 4.1 already mentioned that unsupervised learning algorithms can lead to a 'black box' which can be difficult to understand how they arrived at their predictions. Especially with mission critical systems, it is important for the stakeholders to understand how the AI works. XAI can also be used to detect data bias or mistakes in the model, this can help to improve the AI algorithm.

In 2017, the US military's research agency, DARPA, initiated the XAI program with the objective of improving the transparency of AI decision-making processes. At the same time, the Chinese government released plans to promote highly explainable AI systems. In 2018, the European Union introduced the General Data Protection Regulation (GDPR), which allows individuals the right to receive explanations when algorithmic decision-making impacts them. As the use of AI continues to expand, the demand for explainability will increase among users affected by AI decisions and the developers of AI systems. This focus on explainability comes from the need to understand and justify the reasoning behind AI outputs, ensuring accountability and trust between AI systems and their stakeholders [45].

The main focus of XAI is to develop machine learning techniques that are able to create models that are more transparent and easily understood, while also maintaining a high level of performance in terms of prediction accuracy and also allow humans to trust and effectively manage the AI system. Figure 5.2 shows the concept of XAI that allows users to understand the AI's strengths and limitations, give insight into how the system will function in the future, and potentially offer insights on how to correct errors made by the AI [46].

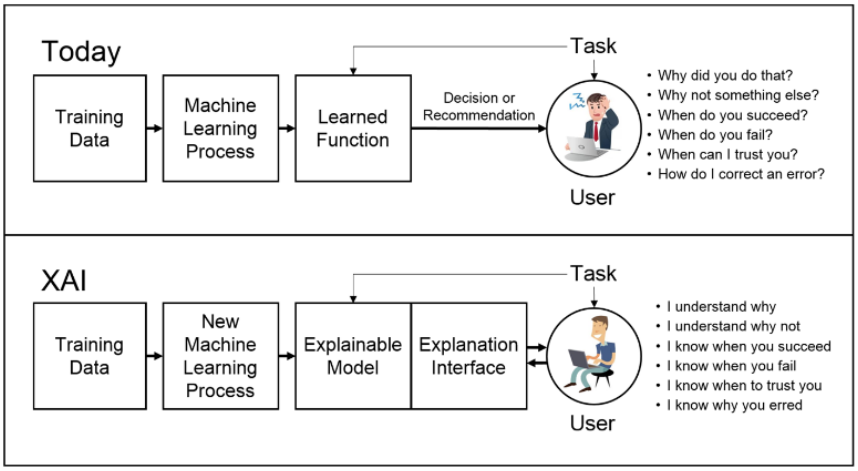

Figure 5.2: The concept of Explainable AI [46]

5.3.1. Responsible AI

XAI is part of the Responsible AI framework. Every stage of the machine learning process is open to biases, safety, security, and privacy issues, which can result in systems that fail, or even cause physical harm. Responsible AI must be viewed as a process, this includes activities such as data acquisition and understanding, model development, deployment, and ongoing operations. By adopting Responsible AI, it becomes possible to address the challenges related with biases, safety, security, and privacy throughout the entire AI lifecycle, ensuring that AI systems are developed and used in a way that aligns with ethical and societal considerations [47].

The top three challenges for AI are the costs, machine learning operations (MLOps) and lack of trustworthy AI. Microsoft developed a responsible AI framework which consist of six key elements; accountability, inclusiveness, reliability and safety, fairness, transparency, and privacy and security. These six principles are guides by two perspectives; ethics and explainability [48].

- **Accountability:** The people who design and deploy the AI should be held accountable for its actions and decisions.
- **Inclusiveness:** The AI should take into account the diversity of all human races.
- **Reliability and safety:** To establish trust in AI systems, reliability, and safety are the most important. It is crucial that these systems operate as intended and respond safely to new scenarios.
- **Fairness:** It is necessary to implement checks to prevent the system's decisions from being discriminatory or creating a bias towards any group or individual based on their gender, race, sexual orientation, or religion.
- **Transparency:** Transparency is crucial in helping teams understand the data and algorithms used to train a model, as well as the transformation processes applied to the data. This knowledge provides insights into the model's development, making it possible to make the model transparently.
- **Privacy and security:** The responsibility of safeguarding the data within an AI system falls upon the data owner, with privacy and security being essential components of this system.

5.3.2. Human oversight

Human oversight is an important aspect of XAI that aims to ensure transparency and understandability of AI systems for human users. The European Union High-Level Expert Group on Artificial Intelligence (AI HLEG) emphasizes the importance of human agency and oversight in the development of AI systems. Respecting human autonomy and decision-making is crucial, and to respect these fundamental rights, making an impact assessment prior to the development of AI systems is essential to identify potential risks [49].

User knowledge is critical for understanding and engaging with AI systems. Users should have sufficient knowledge to assess and question the system's decisions. User autonomy should serve as the foundation of AI systems, enabling users to make informed choices aligned with the objectives of the AI system. Furthermore, appropriate safeguards should be implemented to prevent users from being subjected to fully automated decisions that may have legal implications [49].

In the context of developing AI systems for mission critical purposes, human oversight is important to ensure that AI systems do not compromise human autonomy or lead to harmful effects. Three governance mechanisms to achieve human oversight are: human-in-the-loop (HITL), human-on-the-loop (HOTL), and human-in-command (HIC) [49].

HITL means that the possibility of human intervention at every step of the decision-making cycle, while HOTL allows human intervention during the system's design phase and its operational

monitoring. HIC involves overseeing the entire activity of the AI system, such as its societal impact, economic implications, and ethical considerations. It also involves determining when and how to deploy the system in specific situations. These mechanisms also include the decision of not using an AI system at all [49]. Figure 5.3 illustrates the operation of XAI with HITL, highlighting the need for explainability at each stage of the AI cycle. It also shows the challenges faced in determining what needs to be explained, at which point in the cycle, and to whom. These are ongoing questions that XAI researchers are addressing.

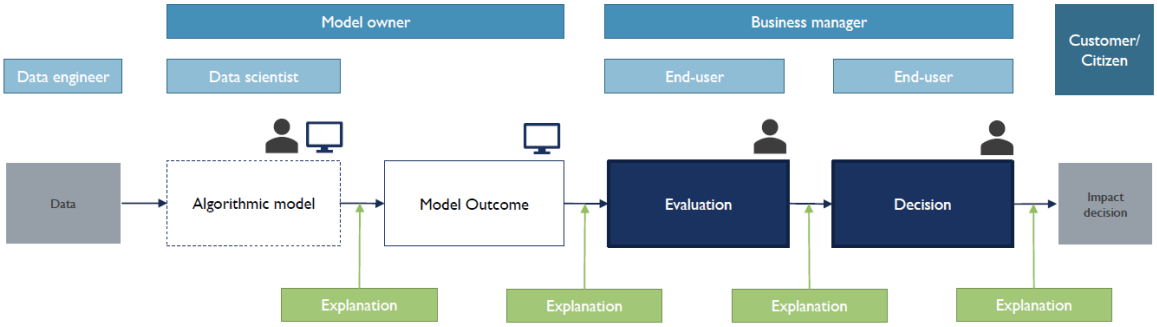

Figure 5.3: Human-in-the-loop during the complete AI lifecycle [50]

The level of oversight required may vary depending on the application and potential risks. Generally, as human oversight decreases, greater importance should be placed on testing and more governance measures. AI systems can effectively support human autonomy and decision-making by prioritizing human agency and oversight. Consideration of human oversight is indispensable when integrating AI into mission critical systems [49].

5.4. Conclusions

The main goal of this chapter was to examine the security, risks, and explainability aspects of AI systems and answer the sub-question: 'How can AI systems be designed to mitigate risks, and ensuring security and explainability?' The chapter highlighted the various types of attacks that can target different each step of the ML model and proposed measures to protect the AI system, including data quality assurance, validation, trust in external parties, and training against attacks. It is important to be aware of these security issues and how to mitigate them in the MCPTT system. The chapter also looked at the importance of measuring and managing the acceptable level of risks. While complete risk mitigation is not possible, the importance of being aware of potential risks and implementing suitable risk management is crucial when working with MCPTT systems. It is recommended that these guidelines are used when implementing AI in the MCPTT system. The chapter also looked at the need of understanding how AI models reach their outcomes. While the field of explainable AI is still relatively new, adopting a Human-in-the-Loop approach is recommended for the MCPTT system to ensure explainability in the decision-making processes.

6

Artificial Intelligence in telcos

The telecom industry is seeing a growing adoption of Artificial Intelligence to improve network efficiency and drive revenue growth. AI is revolutionizing telecommunications companies by enabling data-driven decision-making. This section explores the use and potential benefit of AI in the telecom industry, specifically focusing on Artificial Intelligence for IT Operations (AIOps) and the role of AI in Network Operation Centers (NOCs) to answer the sub-question: 'How to measure the impact of AI in network management?'

6.1. The need for AI

Telecommunication companies are using AI to improve customer service, automate repetitive tasks, and effectively manage the vast volumes of data generated. In the future, AI will play an even larger role in the telecommunications industry, with the use of AI to improve network optimization, real-time network adaptability, predictive maintenance, personalized user services, and proactive issue resolution. The use of AI is driving the telecommunications sector to a new level of intelligence and efficiency [51]. Gartner has reported a significant increase in the number of IT automation initiatives, indicating that 94% of executives are investing or planning to invest in IT operations automation. Although currently less than 20% of the Global 5000 companies have a centralized IT automation function, it is anticipated that this figure will rise to 90% by 2025 [52].

The deployment of 5G networks is currently underway worldwide, transitioning from 5G Non-Stand Alone (NSA) to 5G Stand-Alone (SA), which unlocks the full potential of 5G. 5G networks are designed to support three primary services: enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable low-latency communications (URLLC). This also creates increasing complexity, and creates a challenge in efficiently managing the applications and network configurations while ensuring quality. The key challenge lies in automating and dynamically configuring the network functions for each UE in real-time to optimize performance and meet the specific requirements of various applications.

For improving the availability, AI can be effectively applied in two key areas within the telecommunications network management: network optimization and predictive maintenance. By using AI algorithms, traffic data can be analyzed to optimize network performance. AI can also be used to identify and predict network issues, allowing for proactive resolution before end-users are affected. This improves network quality, reduces customer complaints, and improves overall availability. Predictive maintenance based on previous data allows providers to anticipate future network problems, facilitating quicker RCA during network monitoring. This proactive approach contributes to better network availability [53].

Mobile operators can implement AIOps and Dark Network Operations Center concepts to improve their network management. AIOps lets mobile operators use AI algorithms to analyze large

volumes of network data in real-time. By detecting anomalies, and providing insights, AIOps helps operators to find network issues before they escalate, leading to reduced downtime and improved network quality. In a Dark NOC, the majority of network monitoring components operate autonomously, eliminating the need for human intervention, making it possible 'to turn off the light'. This concept can be seen as the end-goal of the AIOps framework. During the implementation of a Dark NOC, it is crucial to look at the security measures and incorporate XAI features to ensure transparency in decision-making processes and to maintain the integrity of the system, which chapter 5 discussed.

6.2. AIOps

Gartner started with the term AIOps and uses the following definition: "AIOps platforms utilize big data, machine learning and other advanced analytics technologies to directly and indirectly enhance IT operations (monitoring, automation, and service desk) functions with proactive, personal and dynamic insight. AIOps platforms enable the concurrent use of multiple data sources, data collection methods, analytical (real-time and deep) technologies, and presentation technologies" [54].

The process of addressing and resolving disruptions in a network can be divided into three key steps: observation, engagement, and action, as illustrated in figure 6.1. During the observation step, an AIOps platform automatically collects crucial data such as logs, metrics, alerts, and events to identify the specific incident that triggered the disruption. In the engagement step, the platform analyzes this data, detecting patterns and removing the need for manual data gathering and interpretation. Finally, the root cause of the disruption is identified, making it possible to take appropriate actions [55].

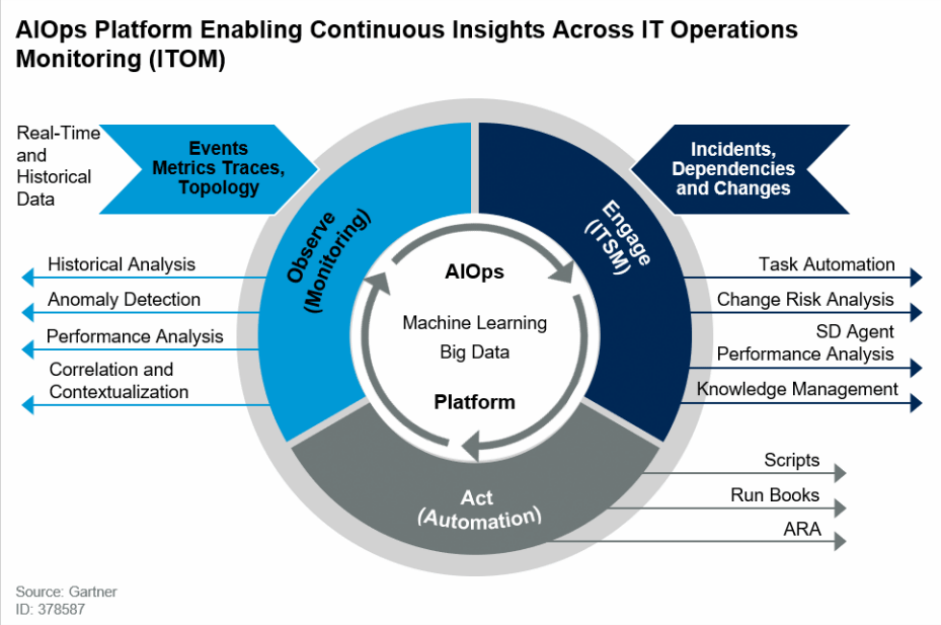

Figure 6.1: AIOps Cycle [55]

The adoption of AIOps brings several benefits to organizations. These include faster identification, resolution, and prevention of outages, compared to manual analysis of alerts from multiple IT operations tools. AIOps benefits include improved resolution time, reduced operational costs, increased observability and collaboration, and proactive and predictive management [56].

AIOps is relevant in various use cases, including root cause analysis, incident management, anomaly detection, capacity planning, and change management. By using machine learning algorithms and big data analytics, AIOps enables the identification of patterns and anomalies in IT systems that may be difficult for humans to detect. Figure 6.2 shows the current and future applications of AIOps, demonstrating the increasing complexity involved in each step of AIOps. It is crucial to maintain an understanding of how AIOps is implemented within the system [56].

For mission critical requirements, AIOps plays a crucial role in improving availability. For example, AIOps can quickly detect network anomalies, identify their root causes, and provide recommendations to prevent similar incidents in the future. AIOps also supports change management by identifying potential risks and impacts of changes before their implementation, enabling informed decisions and minimizing the risk of downtime or other effects. AIOps contributes to improving the availability, and efficiency of IT systems.

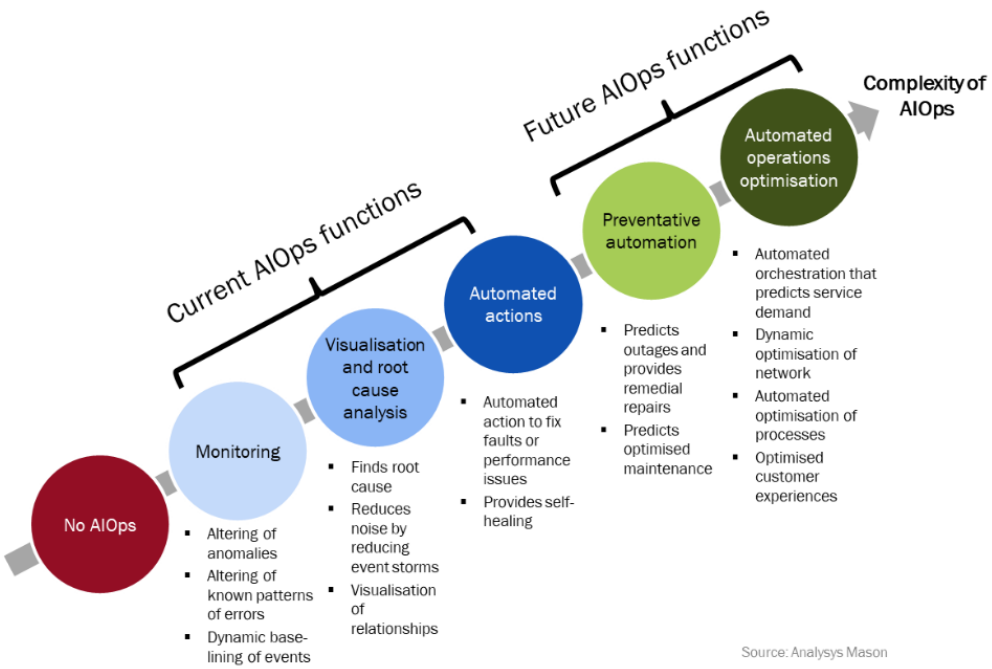

Figure 6.2: AIOps Functions [57]

6.3. Network Operation Center

The Network Operations Center plays a crucial role in the management and operation of networks on a 24/7 basis. The operation of a NOC can result in high costs and resource consumption. To reduce these costs and maintain customer satisfaction, automation has become increasingly important. By centralizing control and reporting interfaces, the NOC creates easy access and visibility, enabling coordinated activities. Traditional NOCs rely on manual monitoring of alarms and KPIs, which may not fully identify or address network issues unrelated to equipment faults. This limitation is particularly significant as networks are dynamic and standard monitoring methods may overlook various causes of degraded performance, resulting in reduced customer experiences [58].

The primary objective for network operators is to deliver exceptional and consistent network availability at a low cost. The adoption of a fully automated NOC, also known as a dark NOC, can help achieve this goal by implementing closed-loop processes and predictive maintenance.

A dark NOC can link alarms from various elements to actions and responses. By using ML techniques, a proactive approach can be taken to maintain high network quality [59].

Telecom operators face the challenge of managing operational expenditure (OPEX), which can account for up to 40% of their total expenses. The rise of new technologies provides opportunities for automation in network operational processes, minimizing human involvement and driving cost reductions. The concept of an automated NOC has emerged as a solution where operational tasks can be performed with minimal staff. The vision of an automated, dark NOC is considered an ideal solution to address the challenges faced by telecom operators [59].

Furthermore, network faults can often be unclear. Multiple alarms triggered by different elements and systems may occur simultaneously, but only one of them may be the root cause of the fault. Identifying the root cause can be time-consuming. Also, certain alarms, referred to as “flapping alarms,” intermittently appear and disappear within a short period, leading to critical alarms if not quickly addressed. The NOC plays a crucial role in quickly assessing the situation, determining the appropriate response, and taking action to minimize the impact on customers. By reducing the need for physical site visits or optimizing the efficiency of these actions, significant operational cost reductions can be achieved [59].

Figure 6.3 illustrates the outcomes of a case study conducted at Elisa, a Finnish mobile operator that implemented a Dark Network Operations Center. The figure shows the impact of different operational approaches on resolution time and customer experience. In the manual operations scenario, the resolution time is highly unpredictable due to human factors, and the point of customer impact is reached. Transitioning to automated reactions results in a significant 79% improvement in resolution time, yet customer impact still exists. With the implementation of predictive automation using machine learning, 69% of alarms are predicted in advance. This proactive approach ensures that the identified alarms are addressed before they impact the customers, effectively preventing any negative consequences on their experience. This shows the power and effectiveness of Dark NOC in mobile operator network management.

The case study focuses on three key areas: automated reactions to alarms, performance automation, and predictive automation. Automated reactions to alarms involve responding to specific alarm signals, such as a base station being out of service, through remote reset procedures and notifications to field maintenance teams or specialized experts. It is essential to integrate sensor alarms related to environmental and physical factors into the automation program to ensure complete automation [59].

Performance automation focuses on continuously monitoring KPIs in the network. Standard NOCs typically monitor equipment alarms to detect service outages or performance threshold breaches. This approach may be limited as it relies on static criteria and fail to detect issues that result in degraded performance. To address this limitation, a more flexible method of monitoring performance can be used. This can involve using ML algorithms to detect anomalies in the network based on historical data. These algorithms can identify trends, sudden changes, or differences that may otherwise go unnoticed [58].

Preventive automation involves using data and machine learning to predict and prevent incidents before they occur. This process consists of analyzing network data, such as KPIs, error patterns, and health status reports, to understand and identify normal and anomalous indicators. ML algorithms can process and analyze large volumes of data, detect patterns, and correlate events. As a result, these algorithms can take actions before an alarm is raised, preventing incidents from happening and ensuring end-user satisfaction [59].

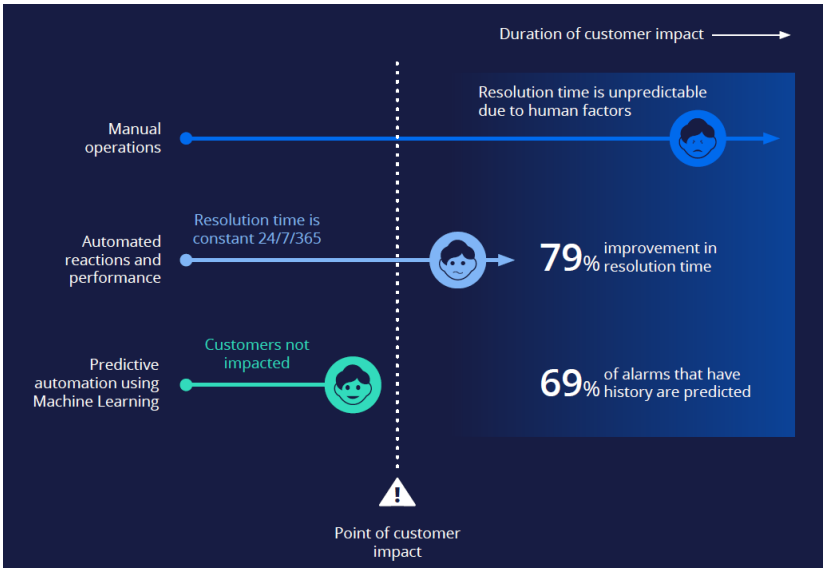

Figure 6.3: Impact on customers with different NOC operations [59]

6.4. Near Silent Issue

For each impactful problem, it is possible to develop an AI algorithm to, ultimately, reach a Dark NOC. The first step in this process is to implement an anomaly detection AI algorithm that can identify potential problems within a system. Once the anomaly detection algorithm is deployed, a domain expert can assess and determine which problems can be effectively addressed using a closed-loop AI algorithm. It is crucial to determine the AI algorithm's performance, comparing it to the existing implementation in terms of accuracy in problem detection and resolution time. This evaluation helps determine the AI's effectiveness in improving the system.

Through interactions with Ericsson Research in Sweden and the USA, I gained insights into their successful implementation of a ML algorithm to tackle a particular challenge, known as the Near Silent Issue (NSI). NSI refers to the sudden drop of the downlink cell throughput. The complex nature of this issue causes that operators do not recognize the problem. Figure 6.4 shows the impact of the NSI. It has been demonstrated that the NSI is associated with a decrease in cell throughput and PRB usage, while the number of RRR users increases. This leads to a significant reduction in throughput for every user.

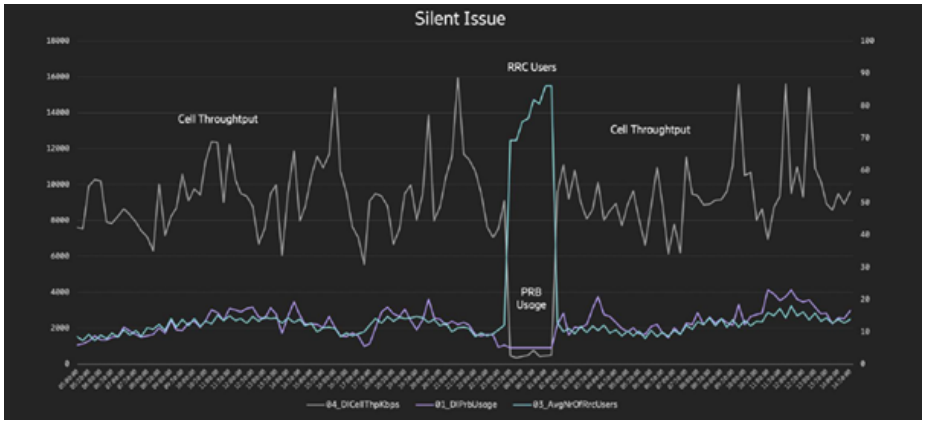

Figure 6.4: Graph of the Cell throughput, RRC Users and PRB usage that show the NSI

The first step Ericsson did, to implement the ML algorithm, was to find the KPIs that relate to this problem. The difficulty in this problem lies in finding the KPIs that relate to the root cause. First, the exact problem needs to be identified. When the problem is found, a domain expert can look at the different KPIs. With the Near Silent Issue, the end-user notices that a web page cannot load or that messages, from Over-The-Top services, are not sent. So the KPIs that are involved are Cell Throughput, UE throughput and Latency. First, KPIs that are related to the end-user needs to be checked. This includes KPIs such as scheduler performance, air interface metrics, uplink/downlink noise. It is also possible to check the KPIs that are related to the Ethernet and core network. After this, it is possible for the domain expert to come up KPIs that can be connected to the issue.

Rule-defined algorithms do not provide the desired accuracy due to their limitations. These algorithms rely on predefined rules and conditions to make decisions and leads to both false positives and false negatives, which results in inaccurate outcomes. To overcome these limitations, a ML algorithm is implemented. The ML algorithm can learn from data and adapt their decision-making based on patterns and examples. Running a ML algorithm for the NSI shows an accuracy of 99.99999%. An illustration of the accuracy of the ML algorithm and the other types of automation for the NSI is shown in figure 6.5. This can be used to explain the difference between a rule-defined algorithm and a ML algorithm.

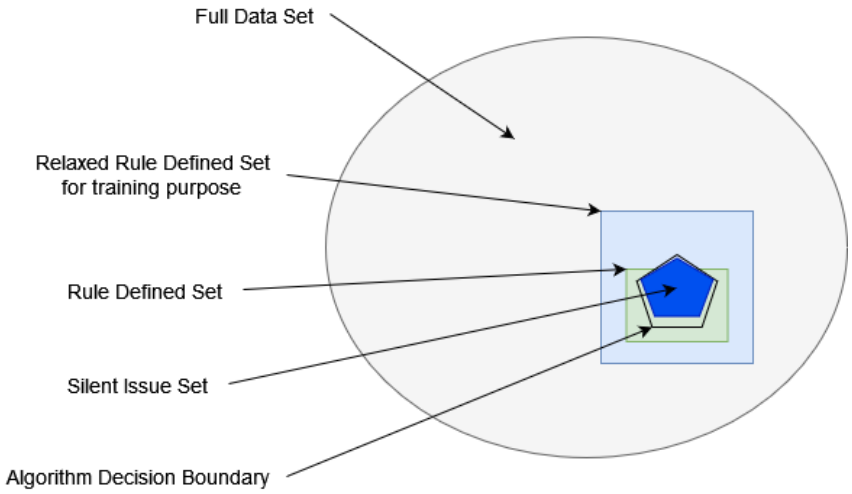

Figure 6.5: Accuracy of a ML algorithm compared to other automation algorithms

6.5. Conclusions

This chapter addressed the sub-question: 'How to measure the impact of AI in network management?' The case study of Elisa's dark NOC demonstrated a 69% of the alarm are predicted in advance, showing the significant impact of AI in network performance. However, it is essential to test the performance of the AI algorithm by comparing them to the existing implementations in terms of accuracy in problem detection and resolution time. This ensures that AI algorithms deliver advantages and help operators make better decisions to optimize network availability. Also, when implementing such a Dark NOC, it needs to comply to the security measures and contain XAI features.

7

Current use of AI for network management at KPN

Chapter 3 introduced the elements of the MCPTT system and showed how they are connected. This chapter will focus on the current use of AI for network management at KPN. Currently, there are already network elements where KPN is experimenting with AI for anomaly detection, but not yet specific for the MCPTT system.

In the current implementation of AI within the network management of KPN, the intelligence of both AI algorithms and human experts are combined to achieve optimal results. AI plays a crucial role in analyzing the huge amounts of data to detect anomalies within a system. When the AI detects a certain anomaly, it requires human expertise to determine the significance of the problem. If there is an ongoing maintenance in the area where AI detects an anomaly, the system expert can decide as this anomaly is expected or requires action.

Currently, the AI monitoring includes several components: Transport core, the Filter layer, Metro Core, Metro Bridge, Loadbalancers and mobile CPE. These elements are integrated using a Kafka data bus, and they are fed with real-time telemetry data. Transport Core, Filter layer, and Metro Bridge and Core are parts of the MCPTT system as well. Figure 7.1 illustrates the same elements as figure 3.1 but in this updated visualization, the components monitored by AI are highlighted in red. This figure shows that the AI monitoring primarily addresses the generic elements. However, the current monitoring is conducted at an individual element level rather than across the entire system. As a result, the algorithm's current focus is on assessing the availability of individual nodes rather than providing insights into the end-to-end system performance. To improve the availability of the MCPTT system, it becomes essential to extract specific MCPTT information, such user plane and control plane traffic, from these nodes.

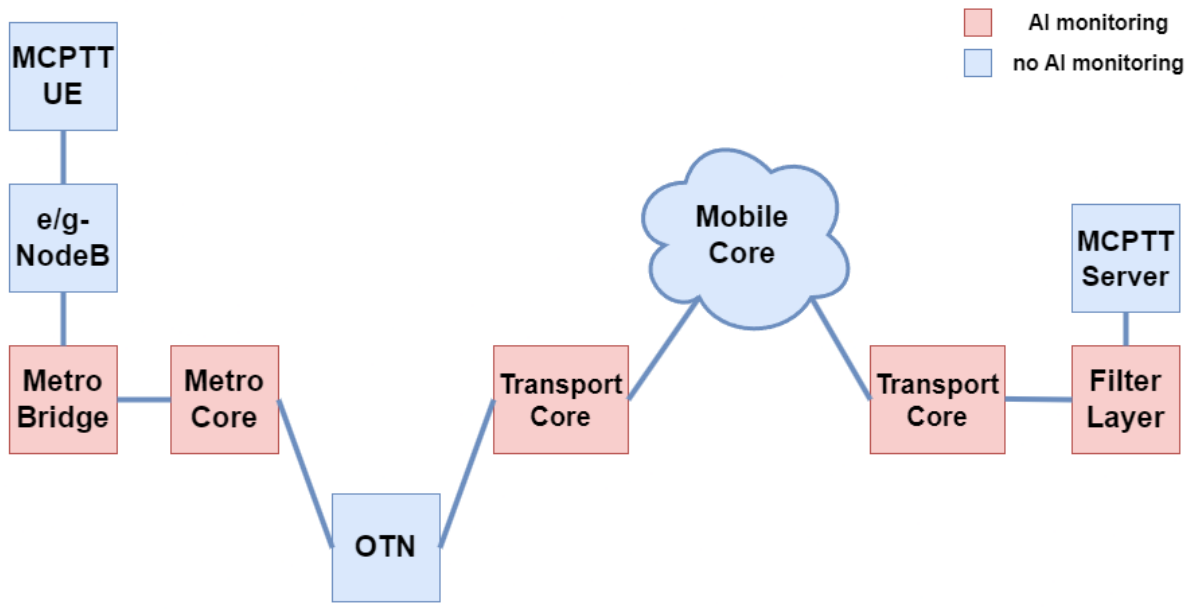

Figure 7.1: Element in the MCPTT system that are currently monitored with AI

To explain the working of the AI algorithm, we dive into the Metro Core and Bridge, because this domain is right now the most worked out and therefore the most concrete example. The working of the AI algorithm can be explained in steps, in the first step, the algorithm in connected to the domain. The system begins with connections to all routers, ensuring up-to-date access to the network topology. It collects hardware and software configurations and parameters from all elements inside this domain. The result is a precise report and detailed maps, presenting the managed network's current inventory, topology, IP address space map, and analysis of Syslog events and SNMP alarms.

In the second step, real-time data is fed into the algorithm, where anomalies are detected. It clusters, with K-means, these anomalies on more than 50 different characteristics. The dashboard interface provides the expert with a graphical 360° overview of these clusters, allowing for real-time visibility, as shown in figure 7.2. The time is visible on the red axis, the blue axis indicates the network elements in an alphabetic order en the height of the spikes indicate the severity of the anomalies. This creates a compact overview for the system expert to monitor the complete domain.

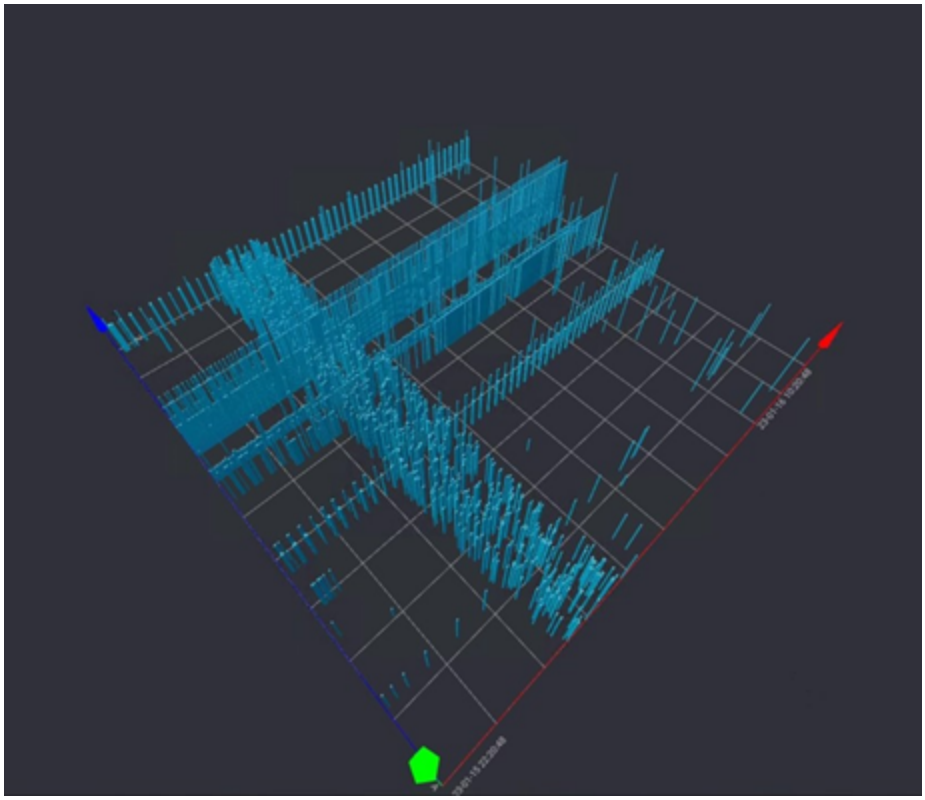

Figure 7.2: Graphical overview of the clusters of the AI algorithm

After the anomalies are clustered and presented in the dashboard interface, the system expert steps in during the third step. The expert assesses the clusters and makes decisions about which anomalies need attention and which can be disregarded as normal network behavior, or planned maintenance. This expert intervention plays a crucial role in the ongoing education of the AI system.

The AI system learns from the expert's labels, enabling the AI system to become improve in identifying, categorizing, and understanding different types of anomalies. This learning process is important for the AI algorithms, making them more accurate and adaptive to the specific domain. Through this collaboration between the AI system and the human expert, the overall effectiveness in anomaly detection improves over time. The AI system becomes more adjusted to the network behaviors within the domain, improving its ability to separate normal network activities and potential anomalies.

With the first implementation of this application within KPN, the AI only detected anomalies parallel to the normal monitoring software. In this time, system experts were able to label clusters and understand the working of this AI application. Recently, different stakeholders sat together and determined which anomalies of the Metro Core and Bridge were crucial and could be sent to the normal monitoring software, as shown in figure 7.3. This is a continuous loop where the anomaly detection is translated into actionable events and put into the ticketing system for further actions by engineers.

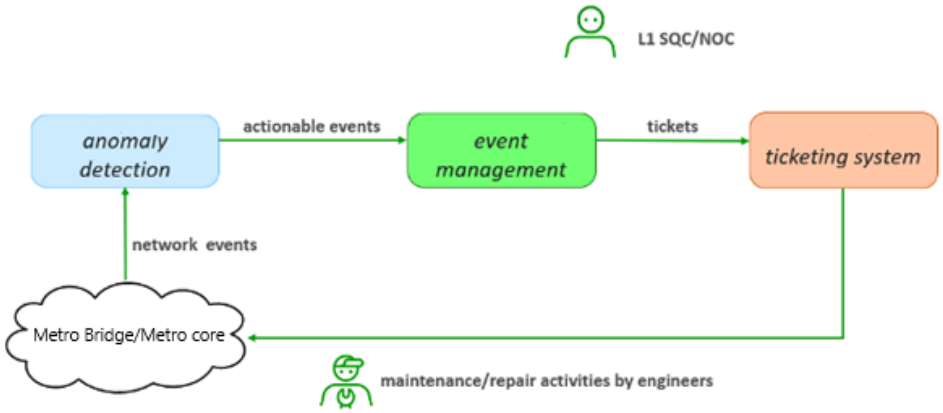

Figure 7.3: AI monitoring of Metro Bridge and Core and sending the events to the ticketing system

Quantifying the exact impact of AI in terms of numbers or percentages can often be challenging. However, one way to express the advantage of AI within a system is by considering the number of anomalies detected that would otherwise go unnoticed without AI. This demonstrates the unique capability of AI to identify and address issues that might not be clear through other means. Evaluating the types of anomalies detected by AI that have the potential to cause failures can provide further insight into the significance of AI. By measuring both the overall number of anomalies detected and the specific anomalies that, if left unaddressed, could lead to failures, it is possible to effectively express the value and advantage that AI brings to the system.

Right now, with this implementation within KPN, the results show that the events from the AI software contain sufficient information, and that the events refer to real issues in the system. The most promising result is that the anomalies that are found and send to the normal monitoring software, were not detected in any other way. So this helps to find unknown errors inside the system and improve the overall availability.

To summarize, this chapter demonstrated that KPN is currently using AI for network management in the generic elements of the network, showing the way for the future implementation of AIOps. The findings presented in this chapter form the foundation for the next chapter, which will focus on conducting experiments to explore the potential and future use of AI in the MCPTT system. The aim is to further extend the implementation of AIOps in the network, using the capabilities of AI to improve the availability of the MCPTT system. The next chapter will further research the practical applications and benefits of AI in MCPTT by using the knowledge from the current AI implementation in the generic elements.

8

Implementation of AI for specific MCPTT system elements

As explained in the previous chapter, AI is currently used in some generic elements of the system. These elements are redundant and handle an enormous amount of traffic every day, so an anomaly doesn't have to lead to an anomaly for the end-user in the MCPTT system. That is why the end-user and system specific architecture needs to be monitored to reach an end-to-end monitoring. This chapter will answer the sub-question: 'How to best implement AI in the MCPTT system of KPN?'

In Chapter 6.2, an analysis of the various stages of the AIOps framework was provided, especially by figures 6.1 and 6.2, which show an overview of the aspects of AIOps. The focus of this research is the use of AI into the MCPTT system. This can be seen as the first phase within the AIOps cycle, recognized as the 'observe' step.

Besides monitoring the system specific elements in the MCPTT system, there is another team responsible for monitoring the generic elements. This separation of concerns allows for specific monitoring. When considering independent monitoring for the MCPTT system, it becomes necessary to use the existing monitoring infrastructure for the generic elements. This approach avoids the need to monitor the generic elements separately for each individual system, which could result in duplicating efforts and increased complexity.

Figure 8.1 illustrates the implementation of AI in the MCPTT system with the concept of separating concerns. The monitoring process in the MCPTT system involves two types of elements: system specific elements and generic elements. System specific elements refer to components that are unique to the MCPTT system, such as the UE and the server. On the other hand, generic elements include more general components that are present in various systems. To ensure effective monitoring, data such as CDRs and logs are collected from both the system specific and generic elements. By analyzing this data, an AI system can identify potential alarms within the elements.

The AI system plays a crucial role in presenting an understandable overview of anomalies and clusters to the system expert. It helps in detecting anomalies and clustering, enabling the expert to focus on potential alarms. This process is for both the system specific elements, where a system expert is involved, and the generic elements, which require a different system expert. This system expert is only needed for the initial training phase. When the potential alarms are known to the system, it becomes a real-time process where the potential alarms are directly sent to the next step.

Next, the potential alarms identified from these different elements need to be combined and linked together for analysis. Here, another AI system serves as a connecting platform, enabling the

alarms to be linked across the entire system. This linkage creates a faster and more effective RCA, allowing the system expert to quickly find the underlying causes of issues. The result of this linkage can be seen as the 'observe' step from figure 6.1.

It is important to emphasize that the final decision regarding further actions based on the linked alarms lies with the system expert. Their expertise and judgment are crucial to guarantee the explainability of the analysis and maintain a Human-in-the-Loop approach in the process. The combination of the AI system's capabilities and the system expert's domain knowledge ensures a collaborative and effective approach to monitoring and managing the MCPTT system.

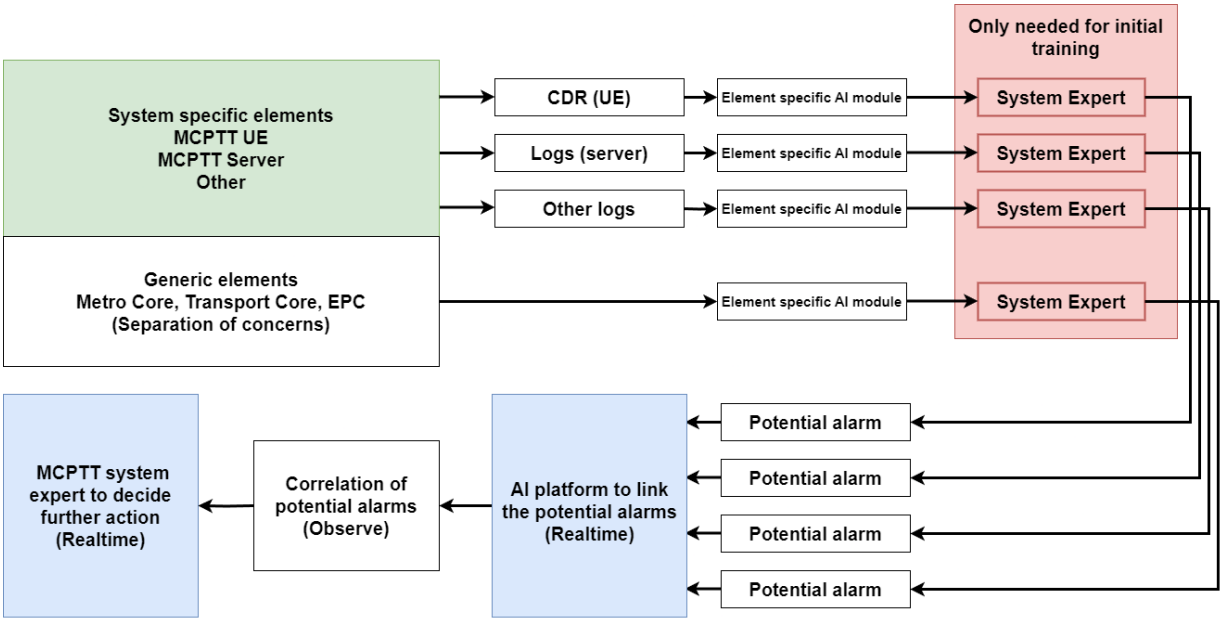

Figure 8.1: Overview of how AI can be implemented in the MCPTT system

In conclusion, monitoring the end-user experience is a crucial aspect of striving for five-nines availability. Traditional approaches that focus only on monitoring individual system elements may not provide a view of the overall user experience. By shifting the focus towards the end-user and actively monitoring them, it becomes possible to detect anomalies that directly impact their satisfaction.

To achieve this, capturing and analyzing end-user log data becomes necessary. This data helps to measure true availability based on the end-user's perception and satisfaction. By integrating end-user log data into the monitoring process, it is possible to gain valuable insights into service quality, identify disruptions, and work towards achieving the level of five-nines availability.

Chapter 2.3 discussed three types of availability to consider when implementing AI for ensuring high availability: Identify weak spots for preventing nation-wide disruptions, monitoring of MCPTT call performance, and monitoring for local disruptions. With the currently available data in this research, it is possible to monitor the MCPTT server to prevent nation-wide disruptions and monitor CDRs to look for local disruptions. By using the information from these log files and CDRs and analyzing their contents, an understanding of the end-to-end system can be achieved. The MCPTT sever log files are analyzed with K-means clustering and the CDRs are analyzed with the isolation forest algorithm.

8.1. K-means clustering to analyze MCPTT server log data

The MCPTT server is a critical component in the system that requires monitoring and analysis. Analyzing this component makes it possible to quickly identify issues that can cause disruptions with the potential to impact the entire nation-wide infrastructure, which is one of the three availability aspects chapter 2.3 mentioned. To gain insights from the MCPTT log data, a K-Means clustering algorithm is used. With this algorithm, it is possible to group similar log events and identify patterns.

In the context of log files, traditional anomaly detection can be difficult to implement because of the nature of log data. Anomaly detection relies on finding outliers in the data. However, log files record the operational activities, such as system events and user actions. As a result, an abnormal data point in log files may not necessarily indicate an anomaly. And in the other way, a common log entry could be an anomaly. The application of traditional anomaly detection methods can lead to a significant number of false positives.

With the approach of using K-means clustering to analyze the log files, it is possible to overcome these disadvantages of traditional anomaly detection algorithms. By applying K-Means clustering to the MCPTT log data, it becomes possible to categorize log data based on their similarities and group them into clusters based on similarity, without making assumptions about anomalies. This clustering analysis provides a clear overview of the MCPTT server and allows the detection of potential issues.

Table 8.1 shows an overview of pros and cons of anomaly detection and K-means clustering. K-Means clustering is chosen for log file analysis because it effectively groups similar log events. Unlike traditional anomaly detection, which focuses on rare occurrences, K-Means is groups together all the log data. By reducing false positives, it provides a more accurate representation of regular system activities, enabling domain experts to optimize the end-user experience. Its interpretable output helps quick decision-making.

Table 8.1: Pros and cons of K-means and anomaly detection in log files.

	K-means	**Anomaly Detection**
Pros	View structure and groupings	Identifies unusual data points
	Categorizes data based on similarities	Suitable for datasets with clear anomalies
	Reduces false positives	Detecting unexpected events
Cons	May not detect rare anomalies effectively	Difficult to apply to log files
	Predefined number of clusters K	Risk of false positives

This subsection will demonstrate the experimental process and outcomes of applying the K-Means clustering algorithm to the MCPTT server. The dataset used consists of logs from the MCPTT server, collected over a three-week period in March 2023. The dataset includes multiple folders containing various log files, making it challenging to identify the specific log files that directly impact the end-user experience. As a result, for the purpose of this study, the focus was narrowed down to the consul logs due to their structured and consistent format. By selecting the consul logs, which follow to a predefined format, it becomes possible to create preprocessing and analysis steps. This decision allows for a more systematized research of the log data, creating a higher degree of confidence in the results and helping the identification of meaningful patterns.

The format consists of three main components: a timestamp, warning level, and informational content. The timestamp indicates the date and time when the log entry was generated. The warning level represents the severity of the logged event, ranging from informational messages

to warnings and errors. Finally, the informational content contains the details of the log entry. Table 8.2 shows an example of a log entry, with the timestamp, warning level and info. The info column can be seen as 4 separate information messages, divided with a colon. The first three messages contain general info and the last part consist of specific info of the server location and the IP-address that is used. Appendix A.1 shows the code that is used to transform the log file to a readable CSV file for further analysis.

Table 8.2: Example of a log entry

Timestamp	Warning Level	Info
2023-03-16T15:40:12.116+0100	[INFO]	agent.client.serf.lan: serf: EventMemberFailed: specific info, location, ip

Encoding the log data is necessary before applying the K-means algorithm, as K-means can only work with numerical values, and log files contain categorical data. There are two common methods for addressing this issue: label encoding and one-hot encoding. Label encoding involves transforming categorical values into numerical values, ensuring that the same categorical values are assigned the same numerical values. This approach encodes the data by creating a unique numerical value for each corresponding categorical value. The code for this technique is shown in Appendix A.2.

In contrast, one-hot encoding creates a table where the categorical values are listed in the first row. Each occurrence of a categorical value in the log file is represented by a 1 in the corresponding row and column of the resulting table. Although this approach creates a larger table, it keeps the distinctness of each categorical value. The code for this technique is shown in Appendix A.3.

To make this concept more clear, an example in presented in table 8.3, it shows the process of label encoding and one-hot encoding for three distinct categories: server, UE, and computer. The table also includes the corresponding number of devices falling under each category, with 5 servers, 100 UEs, and 30 computers.

In the label encoding section of the table, the categorical values, server, UE, and computer, are translated into numerical representations, 1, 2, and 3, respectively. In the one-hot encoding section, a binary column is created for each category in the original data. Each row represents an individual data entry, and a 1 is placed in the corresponding category column if the data entry belongs to that specific category. For example, if a data entry corresponds to a server, the server column will have a 1, and the UE and computer columns will have 0s. It is important to note that numerical values that are already present in the original data are not encoded.

Table 8.3: Example of label encoding vs. one-hot encoding

Label encoding

Devices	Encoded	Number
Server	1	5
UE	2	100
Computer	3	30

One-hot encoding

Server	UE	Computer	Number
1	0	0	5
0	1	0	100
0	0	1	30

To mitigate information loss during the encoding process, a step is taken to split each information column in the log files into four separate columns before proceeding with the encoding step. This additional step is implemented to make sure that the original information present in the log data is preserved as much as possible and accurately represented in the encoded format. This is done

to preserve the data quality, which is as a critical boundary condition during the implementation of AI, as explained in section 4.3.

If every log entry is encoded as a single unit without splitting, the encoding process will treat each log entry as a unique string. As a result, distinct log entries with minor differences, even in a single part of the entry, will receive an entirely separate number during encoding. This can cause that the clustering process may fail to recognize shared information between similar log entries.

By splitting each of this information column into separate columns, specific elements of the log entry can be captured. These attributes are the info column. Splitting the log lines into separate columns ensures that pieces of information are encoded and considered during future analysis. Table 8.4 shows how this can be done with the example from table 8.2. Splitting the log entries into separate columns before encoding allows for more detailed information to be kept about each specific attribute. It also helps to highlight differences in individual parts of the log entries, making the encoded representation more informative.

Table 8.4: Example of splitting the info column into four separate columns

Timestamp	Warning Level	Info1	Info2	Info3	Info4
2023-03-16T15:40:12.116+0100	[Info]	agent.client.serf.lan	serf	EventMemberFailed	specific info, location, ip

The K-means algorithm groups data points into clusters based on their similarity. It creates a multidimensional space, where each dimension is a column in the dataset. Data points are plotted in this space based on their values, and the algorithm calculates the distance between data points. By minimizing the total cluster variance, K-means assigns each data point to the cluster whose center is closest to it.

K-means clustering requires a predefined number of clusters. Ideally, this number should be determined by a system expert or continuously updated using an intelligent AI system. In this research, a cluster number of 10 is selected. The number of 10 is chosen because this number provides enough separation in the data. Each cluster represents a group of data points, making it easier to distinguish between different patterns or behaviors within the dataset. A cluster number of 10 creates a balance between providing a detailed view of the data, as more clusters can lead to smaller groups that only captures one specific log entry, and offering a concise overview of the data, as fewer clusters can lead to more generalization. A total of 10 clusters can help in identifying meaningful patterns without overwhelming the expert with too many clusters.

The results of the K-means with the available log data revealed important findings regarding the use of the 2 different encoding methods. These outcomes highlight the strengths and drawbacks of each strategy, highlighting the need for a domain expert to identify the most effective method for this specific context.

The label encoder demonstrated its effectiveness in grouping together similar logs based on shared features. This encoding method successfully captured common lines within the log data, enabling the identification of clusters with similar log entries. Also, it captured the warning levels as a distinguishing factor. As a result, logs with distinct warning levels were grouped together.

Table 8.5 shows how many log entries were present in each cluster. cluster 1 exclusively contains information-level warning messages, and all entries have identical IP addresses in the information part of the log entry. Cluster 2 consists of one single type of message with an information-level warning. Cluster 3 consists of error messages that share significant similarities with each other, but are not exactly the same. Cluster 4 are general information messages. Cluster 5 is similar to cluster 3, this cluster also contains error messages. However, these errors differ from those

in cluster 3 in some part of the detailed info. Cluster 6 contains infrequent information-level messages. Cluster 7 is another cluster with error messages, but of a different type compared to clusters 3 and 5. Cluster 8 exclusively consists of warning messages. Cluster 9 is similar to cluster 1, this cluster contains information messages, but with a different set of IP addresses. Cluster 10 contains error messages that do not correspond with the other messages observed in other clusters.

Table 8.5: Label encoding, number of logs in each cluster

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
Log entries	105703	9002	598	42974	324
	Cluster 6	**Cluster 7**	**Cluster 8**	**Cluster 9**	**Cluster 10**
Log entries	69	325	16	79284	44

On the other hand, one-hot encoding shows a completely different result. From the 10 clusters that were created, 7 clusters only contained 1 specific log entry. Table 8.6 shows how many log entries were in each cluster. Cluster 1 consists of one single type of message with an information-level warning. Cluster 2 are general information messages. Cluster 3, 4 and 5 contain only 1 log entry which is a specific error message. Cluster 6 is a mix of error, warning and info logs. Again, cluster 7, 8, 9 and 10 only contain 1 specific type of error log.

The clustering with the use of one-hot encoding shows some undesirable results. Specifically, it has resulted in clusters that contain only a single log entry, limiting their meaningful interpretation. Additionally, cluster 6 is a mixture of all types of log entries, making it challenging to identify patterns within this cluster.

Table 8.6: One-hot encoding, number of logs in each cluster

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
Log entries	8996	41597	1	1	1
	Cluster 6	**Cluster 7**	**Cluster 8**	**Cluster 9**	**Cluster 10**
Log entries	187739	1	1	1	1

The difference between label encoding and one-hot encoding in log entry clustering is significant. When using label encoding, there is a concise and informative overview of 10 clusters. Each cluster consists of meaningful log entries that share similar characteristics, making it easier for a system expert to observe and interpret the results. On the other hand, one-hot encoding generates 10 clusters, but its clustering outcome is less meaningful. Seven out of the 10 clusters contain only a single log entry, which provides limited insight and does not offer any meaningful patterns. Furthermore, cluster 6 in one-hot encoding is a mix of all sorts of log entries that are spread out over different clusters in the label encoding. This result of one-hot encoding makes it difficult to interpreter the results, and also makes it challenging for a system expert to gain clear insights from the clustering outcome. Label encoding provides a more clear and concise overview for a system expert to understand the log entries.

Due to the short time period covered by the log files in this study, it is not possible to determine whether there were any disruptions that were visible to end-users based solely on the log data. However, the study demonstrates the use of clustering MCPTT server log files and obtaining a

clear overview of the log data. Determining the meaning and significance of these log files would require domain experts for further analysis.

In conclusion, the application of encoding methods, followed by K-means clustering with a predefined number of clusters, provides a valuable approach for gaining an overview of log files. This shows the possibility of clustering log data and obtaining insights into the underlying patterns and similarities. However, it is important to note that this analysis is just the initial step. Further investigation and domain expertise are necessary to understand the log files accurately. The system expert is crucial for determining the optimal number of clusters and selecting the most suitable encoding method.

8.2. Isolation forest to analyze CDRs

In chapter 2.3, the significance of local disruptions on MCPTT call performance is highlighted, emphasizing the potential impact of factors like inadequate indoor coverage and base station outages. These disruptions can lead to critical communication breakdowns. To address these challenges, CDRs can quickly detect performance degradation at a local level. By harnessing the insights provided by CDRs, disruptions in critical communication due to factors can be identified and resolved, improving the local availability of the MCPTT system.

CDRs contain high-dimensional data and have irrelevant attributes, making them challenging to analyze manually. An effective way of detecting anomalies within the CDRs can be done with the isolation forest algorithm. By using the unsupervised learning of the isolation forest algorithm, it is possible to find anomalies within the CDR data.

The isolation forest, as introduced in chapter 4, works by creating isolation trees. At the root of each tree, a specific feature of the dataset and a split value is chosen. The data is then split into two subsets: one subset with data points having feature values less than the chosen value, and the other subset with data points having feature values greater than the chosen value. If all the trees are constructed, the path lengths are determined. The path length is the number of edges crossed from the root of the tree to the isolated node. Anomalies are expected to have shorter path lengths in the tree because they are easily separable and require fewer splits to be isolated. Figure 8.2 shows this concept, the anomalies, which are easy to isolate, appear higher in the tree.

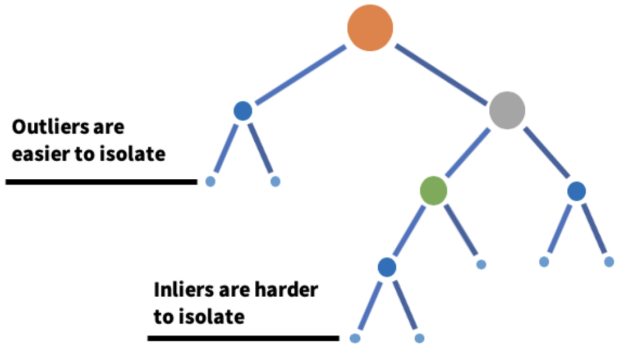

Figure 8.2: Simple example of the isolation forest [60]

By applying the isolation forest algorithm to the CDR data, it is possible to identify unusual patterns. These anomalies can indicate issues that require attention, such as local outages,

service disruptions, or abnormal user behavior. The experimental process and outcomes of applying the isolation forest algorithm to CDR analysis is demonstrated in this subsection.

To begin the analysis of CDRs, a preprocessing step is performed to filter out irrelevant columns and ensure that the data is in a suitable format for further analysis. This step involves removing columns that do not provide relevant information for anomaly detection. Please refer to Appendix B.1 for the corresponding preprocessing code. In the CDR file before preprocessing, there are a total of 37 columns that include various aspects, such as timestamps, phone numbers, media ports, and group IDs.

During the preprocessing, the CDR file is reduced to only 17 essential columns that are considered crucial for the analysis. These maintained columns provide insights into the call characteristics and relevant information for understanding the activities. The columns consist of elements, such as call type, phone numbers of both parties involved, the timestamps of call answer and release, call duration, termination reason, diagnostic information, group ID, location data, media IP address, IMEI, and cell site numbers. By focusing on these attributes, the resulting dataset becomes more manageable.

Once the preprocessing is complete, the iForest algorithm is used for anomaly detection. The algorithm identifies instances that deviate significantly from the normal patterns observed in the data. This method of unsupervised learning has the advantage of detecting anomalies without the need for labeled training data. The implementation of this anomaly detection algorithm can be found in Appendix B.2.

In this study, the first file containing CDRs from February 1, 2023, is used to train the isolation forest algorithm. The algorithm learns the underlying patterns and characteristics of the CDRs during this training phase. Next, the remaining CDR data from the month of February is fed into the trained algorithm for anomaly detection. The output of this process is a list of the first 100 anomalies detected in the file, i.e., 100 anomalies per day. This number can be adjusted depending on the specific requirements of the system. This code can be found in Appendix B.3.

The number 100 is chosen as a threshold for the anomalies detected per day to create a balance between effective anomaly detection and manageable data analysis. This limit ensures that the anomaly detection process remains efficient during this research. It creates a manageable overview of anomalies, which makes it possible to get insights of the working of this algorithm. However, it's essential to emphasize that the choice of 100 anomalies per day is not a fixed value. Depending on the specific requirements and objectives, this threshold can be adjusted. For instance, as the system matures, the threshold can be fine-tuned to align with the needs and the desired level of sensitivity for detecting anomalies.

The results of the anomaly detection process indicate that only CDRs with a call duration of 0 are identified as anomalies. This outcome is desired, as it suggests that the algorithm is capable of effectively filtering out anomalous records that may indicate potential disruptions or abnormal behavior. A call duration of 0 indicates that a call was terminated immediately upon initiation, which is unusual and could be indicative of network disruptions. By focusing on these instances, the algorithm effectively filters out abnormal records that may be caused by system failures, helping to identify potential issues.

Finding cases where the calls were terminated unexpectedly or prematurely is crucial for maintaining the overall quality and reliability of the communication infrastructure. Identifying and addressing these anomalies can contribute to better network performance and enhanced user experience. Therefore, the outcome of the algorithm is desirable as it enables the detection of potential disruptions and abnormal behavior in the network, leading to more efficient troubleshooting and improved overall network stability.

The available CDR dataset contains valuable columns that can provide insights into potential disruptions. For example, the originating and terminating phone numbers can help identify specific users or groups experiencing anomalies. The timestamp data enables the identification of temporal patterns and the detection of unusual call activity during specific time periods. The cell location data allows the identification of cell sites or areas with unusual call behavior. Additionally, the cause for termination and CDR diagnostic columns provides further context and details regarding the reasons for call termination or the specific issues encountered during the call.

The dynamic nature of a mission critical system and the evolving patterns of call behavior in the MCPTT system need adaptive methods for detecting and addressing potential disruptions. AI algorithms are able to handle such complex and dynamic environments, as they can continuously learn and adapt to new patterns and anomalies. By using AI, it becomes possible to perform real-time risk assessments, quickly identifying potential disruptions and taking proactive measures to minimize their impact.

By plotting the anomalies on a geographic map, it would be possible to observe any patterns or clusters that may exist, focusing on specific areas or cell sites that are more susceptible to disruptions or abnormal call behavior. This visualization technique could help in prioritizing maintenance or troubleshooting efforts to improve network availability in this location. It is important to note that this thesis did not include the creation of such a visualization. Instead, the focus was on preprocessing CDR data, training an unsupervised learning algorithm for anomaly detection, and demonstrating the effectiveness of the approach in identifying anomalies.

When the AI system detects a potential disruption or anomaly, it can trigger an alert or notification to a system expert, as explained in the beginning of this chapter. This approach, combining the AI algorithms with human expertise, ensures that the final decisions regarding potential disruptions are made by experts who can assess the situation and consider other factors. The AI system acts as an initial filter, making to possible to efficiently assess the risks.

The alerts from this anomaly detection process can be used to find problems related to network coverage. It becomes possible to identify areas or regions experiencing communication gaps or weak signals. This insight allows to strategically optimize and expand coverage, addressing potential blind spots and enhancing overall network performance. The detection of problematic UE is another crucial aspect of these insights. By identifying UEs associated with call duration anomalies, it is possible to identify devices that might be causing disruptions or experiencing technical difficulties. This targeted approach allows for efficient resolution of device specific issues, ultimately leading to improved service quality for end-users.

With an understanding of network disruptions and coverage issues provided by the anomaly detection process, it is possible to make decisions about future network deployments. For instance, in scenarios where MCPTT is required, the insights gained from the anomaly detection can guide the deployment of dedicated network-in-the-box, a portable communication system that can be rapidly deployed to provide local network coverage, solutions to address coverage challenges in specific areas, such as providing seamless and reliable MCPTT communication for emergency services. Chapter 2 introduced Proximity Service of MCPTT, utilizing this mode can extend network coverage, providing a portable communication system that ensures seamless and reliable MCPTT communication, especially in emergency situations or areas with limited primary network coverage. Right now, this feature is not yet supported by manufacturers.

This research, which involves preprocessing CDR data, using the isolation forest algorithm for anomaly detection, offers a model for identifying and addressing potential disruptions in networks. By using the available data, including call duration, originating and terminating phone numbers, timestamp, cell location, cause for termination, and CDR diagnostic, valuable insights can be

gained into potential disruptions and abnormal call behavior. The combination of AI algorithms and human expertise allows for effective analysis in a changing environment, ensuring the continuity of critical communication systems in an explainable and safe manner.

8.3. Conclusions

Figure 8.1 highlighted the importance of both system specific elements and their interconnections for achieving end-to-end monitoring. The aim was to demonstrate the feasibility of actively monitoring individual components and their relationships within the system. However, due to the limitations of the available data, the desired link between the elements could not be established in the analysis. Although the desired link between system specific elements could not be established, the capability to actively monitor and analyze individual components is a valuable result itself.

The approach to implement AI in the MCPTT system involves the monitoring and independent analysis of system specific elements as the initial step. By focusing on individual elements, with data such as CDR or MCPTT server log data, a compact overview of the systems' behavior can be achieved. When collaborating with system experts, AI algorithms can be trained to detect anomalies. When these anomalies are effectively addressed, this leads to a more available end-to-end MCPTT system. The clear separation of concerns is crucial, as it eliminates the need to monitor generic elements separately for each system, allowing a specific focus on MCPTT elements.

In the future, real-time data is crucial for effective monitoring and decision-making in the MCPTT system. Creating a data pipeline enables the seamless flow of real-time data into the system, facilitating quick identification and resolution of potential errors. This approach provides a comprehensive, end-to-end real-time overview of the entire system's performance, leading to improved operational efficiency and enhanced user experience.

While the current research focused on monitoring specific elements of MCPTT, expanding the monitoring scope to include critical components, like the PCRF, is important. Real-time monitoring of the PCRF is vital because it plays an important role in allocating network resources and giving priority to mission critical communication. By closely monitoring the PCRF's performance, it can be ensured that the necessary QoS levels are maintained, guaranteeing high availability and reliable communication for end-users, particularly during emergencies. Also, this was mentioned in chapter 2.3 as one aspect of availability. With the monitoring elements, like the PCRF, it is possible to monitor if the MCPTT calls comply with the performance requirements.

Incorporating elements like the PCRF into the overall data pipeline will provide a better overview for network management. This will give operators information they can use to make informed decisions, improve network performance, and make sure mission critical communication is available.

Also, stressing the importance of supporting Proximity Services in MCPTT to manufacturers is crucial. In mission critical situations, network infrastructure may become compromised or overloaded, hindering communication capabilities. Proximity Services can act as a valuable fallback mechanism, allowing devices to establish direct peer-to-peer connections in such scenarios, ensuring continuous communication even when traditional network connectivity is limited or disrupted. Manufacturers should integrate Proximity Services into their MCPTT solutions. Implementing this feature will create a resilient communication system.

Although there are certain limitations, there is a significant impact of this research. The results demonstrate that actively monitoring the end-user experience is achievable. Traditional AI algorithms often focus on individual system elements, whereas using user data and AI techniques allows for improving the end-user experience and moving closer to achieving five-nines availability.

9

Conclusions and Recommendations

The main goal of this thesis was to investigate whether and how the use of AI can improve the end-to-end availability of mission critical systems, with a specific focus on Mission Critical Push-to-Talk (MCPTT), which is using the public 4G and 5G network. Therefore, the central research question addressed was: 'How can AI be applied in the Mission Critical Push-to-Talk system to increase the availability of the system?'

Implementing Artificial Intelligence (AI) in the Mission Critical Push-to-Talk (MCPTT) system involves the monitoring and analysis of system specific elements as the first step. By focusing on individual elements, with data such as Call Detail Records (CDRs) or MCPTT server log data, valuable insights can be gained regarding the behavior of the system that can impact the availability of the end-user. Collaborating with system experts, these AI algorithms can be trained to detect anomalies. When these anomalies are effectively addressed, this will lead to a more available end-to-end MCPTT system.

The findings of this study highlight the potential of AI in improving the availability. The use of specific algorithms in this research demonstrated the potential of AI in effectively observing the MCPTT system. The outcomes show that AI monitoring could offer advantages over existing techniques. AI makes it possible to analyze huge quantities of data in a structured way. This makes it possible to identify potential system alarms that might otherwise remain unnoticed. Further research and validation are required to determine the full extent of benefits of AI.

A crucial aspect in this research is the implementation of a clear separation of concerns between system specific and system generic elements. By doing so, it becomes possible to monitor each element individually using specialized AI algorithms. Potential alarms can be evaluated by system experts to decide which potential alarms can influence the end-user.

By monitoring all elements within the system and merging the results, the interdependencies between different potential alarms can be identified. Collaboration with system experts is needed in determining which alarms require further action, ensuring end-to-end monitoring and improving the overall end-user satisfaction. This system expert must take the final decision on further action based on the result of the AI algorithm.

Throughout the entire research, the importance of explainability and safety within mission critical systems was crucial. To ensure explainability, a Human-in-the-Loop approach is essential at each step of the AI model's implementation. This integration of human expertise ensures accountability and transparency in the processes, as human operators are able to understand and provide explanations for the outcomes generated by the AI model.

Furthermore, the implementation of AI in mission critical systems introduces potential vulnerabilities that needs to be recognized and mitigated. To address these concerns, an assessment of

risks was conducted, showing that all stages of the AI model are vulnerable. When implementing AI in mission critical context, measures need to be employed to identify any potential weaknesses in the AI model.

In addition, this research focused on the role of data quality in AI implementation. An AI model's effectiveness is highly dependent on the quality of the available data. Therefore, it is important to identify the essential data and ensure the integrity throughout the entire process to achieve optimal AI integration in mission critical systems.

The findings of this research represent a first step in advancing the use of AI in mission critical systems. To build on this progress, there are some practical recommendations for the KPN MCPTT system. Implementing these recommendations can improve both the availability and the integration of AI within the MCPTT system.

First, this research demonstrated the use of two specific AI algorithms for the MCPTT system, further investigation is needed to explore other AI algorithms. By using system experts to assess the performance of various algorithms for system specific elements will provide valuable insights into finding the most suitable algorithm for every element.

Second, the integration of a Human-in-the-Loop is necessary at this stage of AI. This ensures the crucial element of explainability in the decision-making process. Integrating human oversight into AI operations can enhance the trustworthiness and transparency of outcomes, addressing concerns related to algorithmic accountability.

Third, there needs to be a clear guideline in using AI algorithms in mission critical systems. There are several types of attacks that can influence the performance of the algorithm. Before implementing AI, there needs to be risk assessments and mitigation strategies to identify any risks. This is an ongoing process once the model is implemented.

Fourth, data quality is a boundary condition in introducing an effective algorithm. There needs to be a clear overview of available data and the real-time availability of these data with the use of a data pipeline. Next, there needs to be decided if this data is suitable of improving the availability of the end-to-end system. Keeping a focus on the data quality increases the effectiveness and reliability of AI models.

Fifth, to effectively assess the availability of MCPTT communication systems, the Call Success Rate of the MCPTT calls should be continuously monitored. This critical metric serves as an indicator of the performance, providing real-time insights into the establishment of MCPTT calls during mission critical operations. By consistently monitoring and analyzing the Call Success Rate, it becomes possible to quickly identify potential issues.

Sixth, to improve the MCPTT system's availability, the retrieval of specific MCPTT information from the monitored generic elements is needed. By focusing on acquiring detailed user plane and control plane traffic data from these nodes, a better understanding of the system's performance can be achieved.

Last, while 100% availability is currently not possible, stressing the importance of supporting Proximity Services in MCPTT to manufacturers is crucial. Proximity Services can act as a valuable fallback mechanism, allowing devices to establish direct peer-to-peer connections when the network is not available. Manufacturers should integrate Proximity Services into their MCPTT solutions. Implementing this feature will create a resilient communication system.

By using these recommendations, future research can build upon the foundation established by this research, further advancing the field of AI in mission critical systems. These efforts will contribute to achieving higher levels of availability for the MCPTT system.

Future research can delve into a wider spectrum of AI algorithms that can be used in the MCPTT system. This research includes close collaboration with system expert to find the most fitting algorithm for the MCPTT system. With this collaborative approach, the research aims to find the algorithms that best align with the specific demands of the MCPTT system, contributing to improving the availability of the MCPTT system.

Next, as Explainable AI continues to progress, further investigation can be undertaken to explore the possibility of implementing a dark Network Operations Center scenario. This would include a study of the advantages and benefits of using AI in situations where human intervention and monitoring are limited.

Third, future research can delve into the exploration of the correlation between MCPTT sever log data and CDR data. This research can gain insights about log entries in the MCPTT server that have direct impact on the end-user. With such insights, the potential to identify and respond to potential issues or alarms at an earlier stage could be significantly improved and thereby improving the availability of the MCPTT system.

References

[1] Pierre Fortier et al. *Towards a future-proof Mission Critical communication ecosystem for public safety*. Tech. rep. Capgemini, 2021.

[2] Will Kenton. *Mission Critical*. Sept. 2021. URL: https://www.investopedia.com/terms/m/mission-critical.asp.

[3] Vladimir Atanasovski et al. *Future Access Enablers for Ubiquitous and Intelligent Infrastructures*. Vol. 159. Cham: Springer International Publishing, 2015, pp. 212–215. DOI: 10.1007/978-3-319-27072-2.

[4] GSMA. *Network 2020: Mission Critical Communications*. Tech. rep. London, 2020.

[5] Sang Won Choi et al. "A Feasibility Study on Mission-Critical Push-to-Talk: Standards and Implementation Perspectives". In: *IEEE Communications Magazine* 57.2 (Feb. 2019), pp. 81–87. DOI: 10.1109/MCOM.2018.1700886.

[6] *Network availability: How much do you need? How do you get it?* Tech. rep. Cisco Systems, 2004.

[7] The Critical Communcations Association. *Public Safety prioritisation on commercial networks*. Tech. rep. Newcastle, June 2019.

[8] Marko Höyhtyä et al. "Critical Communications Over Mobile Operators' Networks: 5G Use Cases Enabled by Licensed Spectrum Sharing, Network Slicing and QoS Control". In: *IEEE Access* 6 (Mar. 2018). DOI: 10.1109/ACCESS.2018.2883787.

[9] Aitor Sanchoyerto et al. "Analysis of the Impact of the Evolution Toward 5G Architectures on Mission Critical Push-to-Talk Services". In: *IEEE Access* 7 (2019), pp. 115052–115061. DOI: 10.1109/ACCESS.2019.2930936.

[10] Attila Hilt et al. "Availability Prediction of Telecommunication Application Servers Deployed on Cloud". In: *Periodica Polytechnica Electrical Engineering and Computer Science* 60.1 (2016), pp. 72–81. DOI: 10.3311/PPee.9051.

[11] *What are network KPIs?* URL: https://www.technichegroup.com/determining-kpis-for-network-monitoring/.

[12] Hongcheng Wang et al. *Using AI/ML for Improving IT Operations*. Tech. rep. Washington: Comcast, Sept. 2022.

[13] Robert Sheldon. *Root Cause Analysis*. July 2022. URL: https://www.techtarget.com/searchitoperations/definition/root-cause-analysis.

[14] *Human Error and Root Cause Analysis*. URL: https://www.qualitygurus.com/human-error-and-root-cause-analysis/.

[15] Motorola Solutions. *Driving development of the mission critical PTT standard*. Tech. rep. Chicago, Sept. 2018.

[16] Germaine Tan. *Finding the Right Cyber Security AI for You*. Dec. 2022. URL: https://darktrace.com/blog/finding-the-right-cyber-security-ai-for-you.

[17] Atul. *AI vs Machine Learning vs Deep Learning*. Dec. 2022. URL: https://www.edureka.co/blog/ai-vs-machine-learning-vs-deep-learning/.

[18] *Machine Learning – Thematic Intelligence*. Tech. rep. Global Data, Dec. 2022.

[19] Ethem Alpaydin. *Introduction to Machine Learning*. 2nd ed. Massachusetts Institute of Technology, 2010, pp. 21–47.

[20] Richard S Sutton et al. *Reinforcement Learning*. 2nd ed. Cambridge: MIT Press, Nov. 2018, pp. 2–25.

[21] Mayank Banoula. *Classification in Machine Learning: What it is & Classification Models*. Feb. 2023. URL: https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning.

[22] Zoumana Keita. *Classification in Machine Learning: An Introduction*. Sept. 2022. URL: https://www.datacamp.com/blog/classification-machine-learning.

[23] Sakshi Gupta. *Regression vs. Classification in Machine Learning: What's the Difference?* Oct. 2021. URL: https://www.springboard.com/blog/data-science/regression-vs-classification/.

[24] Jiawei Han et al. *Data Mining: Concepts and Techniques*. 3rd ed. Elsevier, 2012, pp. 243–495.

[25] T. Soni Madhulatha. "An Overview on Clustering Methods". In: (May 2012).

[26] Lukas Budach et al. "The Effects of Data Quality on Machine Learning Performance". In: (July 2022). DOI: https://doi.org/10.48550/arXiv.2207.14529.

[27] Andrew McDonald. "Data Quality Considerations for Petrophysical Machine-Learning Models". In: *PETROPHYSICS* 62 (Jan. 2021), pp. 585–613. DOI: 10.30632/PJV62N6-2021a1.

[28] Douglas M Hawkins. *Identification of outliers*. Vol. 11. Springer, 1980.

[29] Mohiuddin Ahmed et al. "A survey of network anomaly detection techniques". In: *Journal of Network and Computer Applications* 60 (Jan. 2016), pp. 19–31. DOI: 10.1016/j.jnca.2015.11.016.

[30] Yulia Gavrilova. *What Is Anomaly Detection in Machine Learning?* Dec. 2021. URL: https://serokell.io/blog/anomaly-detection-in-machine-learning.

[31] Zhe Zhang et al. "Improved K-Means Clustering Algorithm". In: *2008 Congress on Image and Signal Processing*. IEEE, 2008, pp. 169–172. DOI: 10.1109/CISP.2008.350.

[32] Abhishek Patel et al. "New Approach for K-mean and K-medoids Algorithm". In: *International Journal of Computer Applications Technology and Research* 2.1 (2013), pp. 1–5.

[33] Shubhang Agrawal. *To Start with K-Means Clustering*. Jan. 2021. URL: https://medium.com/analytics-vidhya/to-start-with-k-means-clustering-1c6ee3cb840f.

[34] Fei Tony Liu et al. "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining*. 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

[35] Dong Xu et al. "An Improved Data Anomaly Detection Method Based on Isolation Forest". In: *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*. Vol. 2. 2017, pp. 287–291. DOI: 10.1109/ISCID.2017.202.

[36] Julien Lesouple et al. "Generalized isolation forest for anomaly detection". In: *Pattern Recognition Letters* 149 (2021), pp. 109–119. DOI: https://doi.org/10.1016/j.

patrec.2021.05.022. URL: https://www.sciencedirect.com/science/article/pii/S0167865521002063.

[37] Tommy van der Vorst et al. *Managing AI use in telecom infrastructures*. Tech. rep. National Department for Digital Infrastructure (RDI), June 2020.

[38] Rossella Mattioli et al. *Identifying Emerging Cyber Security Threats and Challenges for 2030*. Tech. rep. European Union Agency for Cybersecurity (ENISA), Mar. 2023. DOI: 10.2824/117542.

[39] Danny Palmer. *The next big threat to AI might already be lurking on the web*. Mar. 2023. URL: https://www.zdnet.com/article/the-next-big-threat-to-ai-might-already-be-lurking-on-the-web/.

[40] General Intelligence and Security Service. *AI systems: Develop them securely*. The Hague, Feb. 2023.

[41] Alex Polyakov. *How to attack Machine Learning (Evasion, Poisoning, Inference, Trojans, Backdoors)*. Aug. 2019. URL: https://towardsdatascience.com/how-to-attack-machine-learning-evasion-poisoning-inference-trojans-backdoors-a7cb5832595c.

[42] Niels Brink et al. *Adversarial AI in het cyberdomein*. Tech. rep. TNO, Feb. 2023.

[43] Ishai Rosenberg et al. "Adversarial Machine Learning Attacks and Defense Methods in the Cyber Security Domain". In: *ACM Comput. Surv.* 54.5 (May 2021). DOI: 10.1145/3453158. URL: https://doi.org/10.1145/3453158.

[44] Elham Tabassi. *AI Risk Management Framework*. Tech. rep. National Institute of Standards and Technology, 2023. DOI: 10.6028/NIST.AI.100-1.

[45] Feiyu Xu et al. "Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges". In: *Natural Language Processing and Chinese Computing*. Cham: Springer International Publishing, 2019, pp. 563–574.

[46] David Gunning. *Explainable Artificial Intelligence (XAI)*. Tech. rep. Arlington: DARPA, Aug. 2016.

[47] Ritu Jyoti. *Empowering Your Organization with Responsible AI*. Tech. rep. Framingham, MA: IDC, June 2020.

[48] Microsoft. *Responsible and trusted AI*. Dec. 2022.

[49] European Commision for Communications Networks Content and Technology. *Ethics guidelines for trustworthy AI*. Publications Office, 2019. DOI: 10.2759/177365.

[50] Tamara Thuis. *Explain What, to whom?* June 2021.

[51] Innovile. *How will the integration of generative artificial intelligence and machine learning shape the future of mobile telecom network orchestration and management automation?* URL: https://www.innovile.com/resources/insights/the-future-of-mobile-telecom-network-orchestration-and-management-embracing-generative-artificial-intelligence-and-machine-learning/.

[52] Resolve. *In Pursuit of the Dark NOC*. Campbell, 2021.

[53] Daniela Levi. *6 Common Uses of AI in Telecommunications*. Mar. 2018. URL: https://techsee.me/blog/artificial-intelligence-in-telecommunications-industry/.

[54] Andrew Lerner. *AIOps Platforms*. Sept. 2017. URL: https://blogs.gartner.com/andrew-lerner/2017/08/09/aiops-platforms/.

[55] PureStorage. *What Is AIOps? Artificial Intelligence for IT Operations*. URL: https://www.purestorage.com/knowledge/what-is-ai-ops.html.

[56] IBM. *What is AIOps?* URL: https://www.ibm.com/topics/aiops.

[57] Justin van der Lande. *Using AIOps solutions in the telecoms industry: a market assessment*. Tech. rep. London: Analysys Mason Limited, Aug. 2021.

[58] Thomas Nilsson. *AI-Powered Anomaly Detection*. Tech. rep. Elisa Polystar, 2022.

[59] Elisa Polystar. *Automate the Network Operation Center*. Tech. rep. 2020.

[60] James Verbus. *Detecting and preventing abuse on LinkedIn using isolation forests*. Aug. 2019. URL: https://engineering.linkedin.com/blog/2019/isolation-forest.

A

Python scripts MCPTT server clustering

The following appendix provides a collection of Python scripts for the MCPTT log server developed and utilized in the course of this research. These scripts serve various purposes, ranging from data preprocessing and analysis to algorithm implementation.

A.1. Preprocess log files

```
# -*- coding: utf-8 -*-
"""
@author: Stan Mulder
"""

import csv
import datetime
import re

def process_log_file(input_file_path, output_file_path):
    # Open the input and output files
    with open(input_file_path, 'r') as input_file, open(output_file_path, 'w', newline='') as output_file:
        # Create a CSV writer
        writer = csv.writer(output_file)

        # Write the headers to the output file
        writer.writerow(['Timestamp', 'Warning Type', 'Info'])

        # Loop through each line in the input file
        for line in input_file:
            #Some lines do not correspond to the standard, so first try. If fail, skip
            try:
                # Split the line into its individual components
                parts = line.split(' ')
                parts = list(filter(None, parts))

                # Extract the timestamp, warning type, and info from the line
                timestamp = datetime.datetime.strptime(parts[0], '%Y-%m-%dT%H:%M:%S.%f%z')
                warning_type = parts[1]
                info = ' '.join(parts[2:]).strip()

                # Remove the colon from the time offset
                offset = timestamp.strftime('%z')
                offset = re.sub(r'[:-]', '', offset)

                # Format the timestamp with the modified offset
```

```
                formatted_timestamp = timestamp.strftime('%Y-%m-%dT%H:%M:%S.%f') +
    offset

                # Write the data to the output file
                writer.writerow([formatted_timestamp, warning_type, info])

                # Write the data to the output file
                writer.writerow([timestamp.replace(microsecond=0).isoformat(timespec=
    'milliseconds'), warning_type, info])
            except Exception:
                continue
```

A.2. K-means Label encoder

```
    # -*- coding: utf-8 -*-
"""
@author: Stan Mulder
"""

import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import silhouette_score
import pickle


def kmeans_label(input_file, output_file):
    # Read the train log file into a pandas dataframe
    df_orig = pd.read_csv(input_file)
    df = df_orig.copy()

    # Convert timestamp column to datetime type
    df['Timestamp'] = pd.to_datetime(df['Timestamp'], utc=True)

    # Split the 'Info' column into four separate columns
    info_split = df['Info'].str.split(':', n=3, expand=True)
    df['Info1'] = info_split[0]
    df['Info2'] = info_split[1]
    df['Info3'] = info_split[2]
    df['Info4'] = info_split[3]

    # Convert timestamp column to datetime type
    df['Timestamp'] = pd.to_datetime(df['Timestamp'])

    # Label encode categorical columns
    label_encoder = LabelEncoder()
    df_encoded = df.copy()
    df_encoded['Warning Type'] = label_encoder.fit_transform(df_encoded['Warning Type
    '])
    warning_type_mapping = {i: label for i, label in enumerate(label_encoder.classes_
    )}

    df_encoded['Info1'] = label_encoder.fit_transform(df_encoded['Info1'])
    df_encoded['Info2'] = label_encoder.fit_transform(df_encoded['Info2'])
    df_encoded['Info3'] = label_encoder.fit_transform(df_encoded['Info3'])
    df_encoded['Info4'] = label_encoder.fit_transform(df_encoded['Info4'])

    # Extract features from log file
    X = df_encoded.iloc[:, [1] + list(range(3, df_encoded.shape[1]))].values
```

```
45
46      # Standardize the features
47      scaler = StandardScaler()
48      X = scaler.fit_transform(X)
49
50      n_clusters = 10
51
52      # Train the k-means model with the optimal number of clusters
53      kmeans = KMeans(n_clusters=n_clusters, init='k-means++', max_iter=300, n_init=10,
         random_state=0)
54      kmeans.fit(X)
55
56      # Add cluster labels to dataframe
57      df_encoded['cluster'] = kmeans.labels_
58      df_encoded['Warning Type'] = df_encoded['Warning Type'].map(warning_type_mapping)
59
60      # Save the trained model to a file
61      model_filename = "kmeans_model.pkl"
62      with open(model_filename, "wb") as file:
63          pickle.dump(kmeans, file)
64
65      # Save the clusters to a text file
66      clusters_file = open(output_file, "w")
67      for cluster in range(n_clusters):
68          cluster_data = df_encoded[df_encoded['cluster'] == cluster].iloc[:, [0, 1,
        2]]
69
70          clusters_file.write(f"Cluster {cluster}:\n")
71          clusters_file.write(cluster_data.to_string(index=False))
72          clusters_file.write("\n\n")
73      clusters_file.close()
```

A.3. K-means One Hot encoder

```
1       # -*- coding: utf-8 -*-
2   """
3   @author: Stan Mulder
4   """
5
6   import pandas as pd
7   import numpy as np
8   from sklearn.cluster import KMeans
9   from sklearn.preprocessing import StandardScaler
10  from sklearn.metrics import silhouette_score
11  import pickle
12
13
14  def kmeans_onehot(input_file, output_file):
15      # Read the train log file into a pandas dataframe
16      df_orig = pd.read_csv(input_file)
17
18      # Convert timestamp column to datetime type
19      df_orig['Timestamp'] = pd.to_datetime(df_orig['Timestamp'], utc=True)
20
21      # Split the 'Info' column into four separate columns
22      info_split = df_orig['Info'].str.split(':', n=3, expand=True)
23      df_orig['Info1'] = info_split[0]
24      df_orig['Info2'] = info_split[1]
25      df_orig['Info3'] = info_split[2]
26      df_orig['Info4'] = info_split[3]
```

```
27
28      # Convert timestamp column to datetime type
29      df_orig['Timestamp'] = pd.to_datetime(df_orig['Timestamp'])
30
31      # Perform one-hot encoding for categorical columns
32      df_encoded = pd.get_dummies(df_orig, columns=['Warning Type', 'Info1', 'Info2', '
        Info3', 'Info4'])
33
34      # Extract features from log file
35      X = df_encoded.iloc[:, 2:].values  # Exclude timestamp and cluster columns
36
37      # Standardize the features
38      scaler = StandardScaler()
39      X = scaler.fit_transform(X)
40
41
42
43      n_clusters = 10
44
45      # Train the k-means model with the optimal number of clusters
46      kmeans = KMeans(n_clusters=n_clusters, init='k-means++', max_iter=300, n_init=10,
         random_state=0)
47      kmeans.fit(X)
48
49      # Add cluster labels to dataframe
50      df_encoded['cluster'] = kmeans.labels_
51
52      # Save the trained model to a file
53      model_filename = "kmeans_model.pkl"
54      with open(model_filename, "wb") as file:
55          pickle.dump(kmeans, file)
56
57      # Save the clusters to a text file
58      clusters_file = open(output_file, "w")
59      for cluster in range(n_clusters):
60          cluster_data = df_encoded[df_encoded['cluster'] == cluster].iloc[:, :5]
61
62          clusters_file.write(f"Cluster {cluster}:\n")
63          clusters_file.write(cluster_data.to_string(index=False))
64          clusters_file.write("\n\n")
65      clusters_file.close()
```

B

Python scripts CDR anomaly detection

The following appendix provides a collection of Python scripts for the CDRs developed and utilized in the course of this research. These scripts serve various purposes, ranging from data preprocessing and analysis to algorithm implementation.

B.1. Preprocess CDR

```
"""
@author: Stan Mulder
"""
import os
import pandas as pd
from sklearn.preprocessing import LabelEncoder
import joblib

def preprocess_cdr(input_csv_file):
    # Load the CSV file into a pandas DataFrame
    data = pd.read_csv(input_csv_file)

    for column in ['cdr.seizure_time', 'cdr.answer_time', 'cdr.release_time']:
        # Apply the to_datetime() function only to non-zero timestamps
        mask = data[column] != '0'
        data.loc[mask, column] = pd.to_datetime(data.loc[mask, column], format='%y%m%d%H%M%S%z').apply(lambda x: x.timestamp())

    # Create LabelEncoder objects for each non-numerical column
    le_group_id = LabelEncoder()
    le_pani = LabelEncoder()
    le_rat_changes = LabelEncoder()
    le_media_ip = LabelEncoder()
    le_calling_party = LabelEncoder()
    le_called_party = LabelEncoder()

    # Fit and transform each non-numerical column using its corresponding LabelEncoder
    data['cdr.group_id'] = le_group_id.fit_transform(data['cdr.group_id'])
    joblib.dump(le_group_id, 'group_id_labelencoder.pkl')

    data['cdr.pani'] = le_pani.fit_transform(data['cdr.pani'])
    joblib.dump(le_pani, 'pani_labelencoder.pkl')

    data['cdr.rat_changes'] = le_rat_changes.fit_transform(data['cdr.rat_changes'])
    joblib.dump(le_rat_changes, 'rat_changes_labelencoder.pkl')

    data['cdr.media_ip'] = le_media_ip.fit_transform(data['cdr.media_ip'])
    joblib.dump(le_media_ip, 'media_ip_labelencoder.pkl')
```

```
38
39      data['cdr.calling_party'] = le_calling_party.fit_transform(data['cdr.
        calling_party'])
40      joblib.dump(le_calling_party, 'calling_party_labelencoder.pkl')
41
42      data['cdr.called_party'] = le_called_party.fit_transform(data['cdr.called_party'
        ])
43      joblib.dump(le_called_party, 'called_party_labelencoder.pkl')
44
45      # Select only the desired columns
46      selected_columns = ['cdr.record_type', 'cdr.calling_party' , 'cdr.called_party',
        'cdr.seizure_time', 'cdr.answer_time',
47                          'cdr.release_time', 'cdr.call_duration', 'cdr.
        cause_for_termination',
48                          'cdr.diagnostic', 'cdr.group_id', 'cdr.location', 'cdr.
        media_ip',
49                          'cdr.media_port', 'cdr.imei', 'cdr.pani', 'cdr.
        internal_corporate_id',
50                          'cdr.rat_changes'] #, 'cdr.partitiondate']
51      data = data[selected_columns]
52
53      # Replace NaN values with 0 for all columns
54      data.fillna(0, inplace=True)
55
56      # Save the preprocessed data to a new CSV file
57      output_csv_file = os.path.splitext(input_csv_file)[0] + '_preprocessed.csv'
58      data.to_csv(output_csv_file, index=False)
59
60      return output_csv_file
```

B.2. Train CDR anomaly

```
1  # -*- coding: utf-8 -*-
2  """
3  @author: Stan Mulder
4  """
5
6  import pandas as pd
7  import pickle
8  from sklearn.ensemble import IsolationForest
9  from sklearn.model_selection import train_test_split
10 from preprocces_cdr_def import preprocess_cdr
11
12 from datetime import datetime
13 import joblib
14 import datetime
15
16 # Load the training CSV file into a pandas DataFrame
17
18 data = preprocess_cdr('cdr/cdr_20230201.csv')
19 data = pd.read_csv(data)
20
21 # Instantiate an Isolation Forest model
22 isolation_forest = IsolationForest(n_estimators=5000, contamination='auto',
       random_state=42)
23
24 # Fit the model to the training data
25 isolation_forest.fit(data)
26
27 # Save the trained model to a file
```

```
with open('trained_model.pkl', 'wb') as f:
    pickle.dump(isolation_forest, f)

 # Load the label encoder from file
label_encoder = joblib.load('labelencoder.pkl')

# Load the label encoder object from the pickle file
le_group_id = joblib.load('group_id_labelencoder.pkl')
le_pani = joblib.load('pani_labelencoder.pkl')
le_rat_changes = joblib.load('rat_changes_labelencoder.pkl')
le_media_ip = joblib.load('media_ip_labelencoder.pkl')
le_calling_party = joblib.load('calling_party_labelencoder.pkl')
le_called_party = joblib.load('called_party_labelencoder.pkl')


# Predict the anomaly score for each CDR in the test data
anomaly_scores = isolation_forest.decision_function(data)

# Reverse the encoding for the label encoded columns in test_features
data['cdr.calling_party'] = label_encoder.inverse_transform(data['cdr.calling_party'
    ])
data['cdr.group_id'] = le_group_id.inverse_transform(data['cdr.group_id'])
data['cdr.pani'] = le_pani.inverse_transform(data['cdr.pani'])
data['cdr.rat_changes'] = le_rat_changes.inverse_transform(data['cdr.rat_changes'])
data['cdr.called_party'] = le_called_party.inverse_transform(data['cdr.called_party'
    ])
data['cdr.media_ip'] = le_media_ip.inverse_transform(data['cdr.media_ip'])


for column in ['cdr.seizure_time', 'cdr.answer_time', 'cdr.release_time']:
    # Apply the to_datetime() function to the non-zero timestamps in test_features
    mask = data[column] != 0
    data.loc[mask, column] = pd.to_datetime(data.loc[mask, column], unit='s')

# Identify and output the test CDRs with the highest anomaly scores
n_anomalies = 100
anomalies = data.iloc[anomaly_scores.argsort()[:n_anomalies]]
anomalies.to_csv('anomalies.csv', index=True)
print(anomalies)
```

B.3. Anomaly detection on CDR

```
# -*- coding: utf-8 -*-
"""
@author: Stan Mulder
"""

import pandas as pd
import pickle
from sklearn.ensemble import IsolationForest
from preprocces_cdr_def import preprocess_cdr
from sklearn.model_selection import train_test_split
from datetime import datetime
import joblib
import datetime
import os

def predict_anomalies_cdr(input_file):

    # Load the preprocessed data from the input file
```

```
    # data = pd.read_csv(input_file)

    #First call the preprocess function to preprocess the data so the labels are
     correct
    data1= preprocess_cdr(input_file)
    processed = input_file.replace('.csv','_preprocessed.csv')
    data = pd.read_csv(processed)


    # Load the trained Isolation Forest model from a saved file
    with open('trained_model.pkl', 'rb') as f:
        isolation_forest = pickle.load(f)


    # Predict the anomaly score for each CDR in the data
    anomaly_scores = isolation_forest.decision_function(data)

    # Load the label encoder objects from the pickle files
  # label_encoder = joblib.load('labelencoder.pkl')
    le_group_id = joblib.load('group_id_labelencoder.pkl')
    le_pani = joblib.load('pani_labelencoder.pkl')
    le_rat_changes = joblib.load('rat_changes_labelencoder.pkl')
    le_media_ip = joblib.load('media_ip_labelencoder.pkl')
    le_calling_party = joblib.load('calling_party_labelencoder.pkl')
    le_called_party = joblib.load('called_party_labelencoder.pkl')

    # Reverse the encoding for the label encoded columns
    data['cdr.group_id'] = le_group_id.inverse_transform(data['cdr.group_id'])
    data['cdr.pani'] = le_pani.inverse_transform(data['cdr.pani'])
    data['cdr.rat_changes'] = le_rat_changes.inverse_transform(data['cdr.rat_changes'
    ])
    data['cdr.calling_party'] = le_calling_party.inverse_transform(data['cdr.
    calling_party'])
    data['cdr.called_party'] = le_called_party.inverse_transform(data['cdr.
    called_party'])
    data['cdr.media_ip'] = le_media_ip.inverse_transform(data['cdr.media_ip'])

    for column in ['cdr.seizure_time', 'cdr.answer_time', 'cdr.release_time']:
        # Apply the to_datetime() function to the non-zero timestamps in data
        mask = data[column] != 0
        data.loc[mask, column] = pd.to_datetime(data.loc[mask, column], unit='s')



    # Identify and output the CDRs with the highest anomaly scores
    n_anomalies = 100
    anomalies = data.iloc[anomaly_scores.argsort()[:n_anomalies]]

    # Get the input file name without extension
    file_name = os.path.splitext(input_file)[0]

    # Set the output file path with the same name as input file + "anomalies.csv"
    output_file = file_name + '_anomalies.csv'

    anomalies.to_csv(output_file, index=True)
```