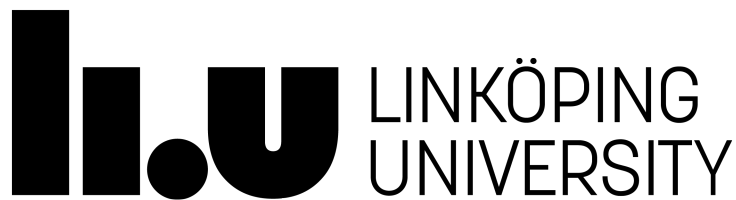


Master Thesis in Statistics and Machine Learning
(LIU-IDA/STAT-A--21/040--SE)

**Anomaly detection and causal inference
in 5G mobile networks**

Mohsen Pirmoradiyan



Division of Statistics and Machine Learning
Department of Computer and Information Science
Linköping University

2021

This is copy write

Abstract

Of vital importance for any system is to continually monitor the functionality of the system to ensure performance. Monitoring the system enables us to detect, if there is, any kind of inconsistency and irregularity that has occurred in the system. An irregularity in a system that deviates it from normal behavior is referred to as an anomaly. 5G network is characterized by extremely low latency, high-speed data transfer, and the vast growing nodes connected via the network. The challenge for such a complex system is to assure the network performance and maintain the quality of the service (QoS) requirements of different services at the standard levels. One of the key aspects of quality management is fault management in which the attempts are focused on detecting the abnormal behavior of the network so a corrective measure can be taken accordingly. This study aims to detect anomalies and diagnose the main causes of behavior changes in a 5G mobile network communication. The latency is the characteristic of interest in this study and considering the abnormal behavior, we addressed the high latency problem in the 5G networks communication. To tackle the problem, two approaches were implemented. Machine learning classification, specifically random forest classifier, was applied to detect the anomalies since the labels of the data instances were available. At this step, we only focused on training the classifier to be able to detect the high-latency instances. To perform the causal inference, the causal Bayesian networks were implemented. The PC algorithm was applied to discover a causal model from the data and then the causal effect of the main causes according to the causal structure was estimated. Considering the anomaly detection, the trained classifier resulted in a precision about 70%. The causal inference study result indicated 5 features which can be considered as the main causal factors affecting the latency variable.

Acknowledgments

I would like to express my gratitude to all those who played a role in the way to accomplish this thesis.

First and foremost, I would like to thank my supervisors deeply.

It is my pleasure to express my deep sense of thanks and gratitude to my academic supervisor **Jose M Peña**, and my supervisor at Ericsson **Caner Kilinc** for all of their support, invaluable feedback, comments, and guidance that brought my work to a higher level.

I am also really grateful to my examiner **Oleg Sysoev** for his valuable comments regarding completing this thesis and for all the time and efforts he has devoted to managing the whole master program as the programme director.

A special thanks to my opponent, **Martin Svensson**, for giving a thorough opposition and providing valuable comments.

I am extremely thankful to all the teachers who put huge efforts over the past semesters into transferring knowledge and science. I learned a lot and I am so grateful.

I also extend my gratitude to **Jörgen Gustafsson**, sector manager, and **Andreas Johnsson A**, master researcher, at Ericsson AI Research without whom this project would not have been possible.

It is my privilege to thank my lovely wife for her constant patience, support, and encouragement throughout my study period.

Last but not least I would like to express my great appreciation to my parents for their constant source of encouragement and inspiration throughout my life.

Contents

1	Introduction	1
1.1	Problem	1
1.2	Objective	1
1.3	Literature review	2
1.3.1	Rule-based methods	2
1.3.2	Classification	3
1.3.3	Clustering	4
1.4	Questions to be answered	5
1.5	Thesis organization	5
1.6	Ethical considerations	5
2	Background	6
2.1	Telecommunication network	6
2.1.1	5G network architecture	6
2.2	QoS - Quality of service	7
2.2.1	More about latency	8
3	Data	10
3.1	Raw data	10
3.2	Preprocessing and preparation	11
4	Theory	13
4.1	Anomaly detection	13
4.1.1	Machine learning approach	13
4.2	Causal inference	17
4.2.1	Correlation vs. causation	18
4.2.2	Different levels in causal inference	19
4.2.3	Causal graphical models	21
5	Method	31
5.1	Feature set	31
5.2	Anomaly detection by classification	32
5.3	Causal inference	33
6	Results	35
6.1	Anomaly detection by classification	35
6.2	Causal inference	37
7	Discussion	39
8	Conclusion	41
	References	42

List of Figures

1	Fifth-generation communication system.[22]	7
2	Latency contribution in E2E delay of a packet transmission.[31]	8
3	A few samples from end-to-end log.	10
4	Highly correlated variables which are figured out through EDA.	11
5	Difference in sampling resolution.	12
6	A merged dataset resulted from the preparation process.	12
7	Decision tree corresponding to the partitioning of two-dimensional feature space(right) and the resulted output regions(left).[37]	14
8	Confusion matrix for a binary classification with two classes, positive and negative.	16
9	Relationship between chocolate consumption and the number of Nobel Laureates per country.[40]	18
10	The causal hierarchy. Questions at level i can be answered only if information from level i or higher is available.[44]	20
11	A directed acyclic graph over the variables A, B, C, and D.	21
12	a) B is an unshielded collider. b) B is a shielded collider.	22
13	A DAG depicting a causal structure to clarify the d-separation concept.[40]	23
14	A causal structure to illustrate the back-door criterion.[43]	26
15	Illustration of how the PC algorithm works. (A) Original true causal graph. (B) PC starts with a fully-connected undirected graph. (C) The X-Y edge is removed because X and Y are independent. (D) The X-W and Y-W edges are removed because of conditional independence. (E) After finding v-structures. (F) After orientation propagation.[46]	29
16	The workflow followed to select the feature set.	31
17	The distribution of the binary target variable. Experiment 2 seems to have the most balanced target variable.	32
18	Histograms showing the typical distribution of the features.	34
19	The classification performance reports for the training and test sets.	35
20	The classification performance for the training and test sets.	35
21	Comparing the trained model performance and the dummy classifier performance.	36
22	The overall performance scores aggregating all test sets.	36
23	Feature importance resulting from the random forest classification.	37
24	The causal structure discovered using the PC algorithm. Drawn in [53]	38
25	Estimated total causal effects.	38
26	Total causal effects bounds.	38

1 Introduction

An anomaly can be defined as any pattern within the data that deviate from a well-defined notion of normal behavior, and so anomaly detection refers to the problem of identifying those patterns in data.[1]

Ensuring the performance of a system in accordance with the predefined standards and requirements necessitates continuous monitoring of its functionality in order to keep track of all the corresponding aspects of the performance. Monitoring the system enables us to detect, if there is, any kind of inconsistency and irregularity that has occurred in the system.

Causal inference can be defined as the process of investigating the causes of events with safety, health, environmental, quality, reliability, and production impacts. Any occurrence in the system resulting in the mentioned ramifications is identified as an “event”.[2]

Identifying an abnormal event in a system is just a preliminary step and additional actions need to keep the system running normally. Having detected an irregularity in a system, determining the main factors contributing in generating that behavior will pave the way to specifying a corrective measure not only to restore the system, but also to prevent the recurrence of such failure.

Considering the fact that a mistake or a failure in a system does not just happen, the “root causes” can be defined as the underlying reasons that lead up to the occurrence of an event and that can be reasonably identified and specified.[2] Identifying such a reason is the key to determining how effective a causal inference process can be in achieving its goal. Improving recommendations for restoring a system and preventing event recurrences can be developed if the reason for the occurrence of the event is accurately identified and specified.

A precisely designed anomaly and its main cause detection can be implemented as part of an automated management system enabling disclosing the abnormal behavior along with their cause(s) which can then be used as an input for developing a corrective measure to improve the system.

1.1 Problem

5G PERFECTA as an international project launched in 2018 aiming to develop automated processes, tools, and mechanisms ensuring 5G service quality, based on data processing and analytics approaches. The main objective of the project is to develop a 5G performance compliance testing assurance model to show the real behavior of the 5G network and services, with the main focus on the provisioning of QoS (Quality of Service) and QoE (Quality of Experience). The vast growth of the devices communicating through the network and also the users connecting to it has considerably increased the network diversity and complexity which necessitates the network automation and optimization for 5G and beyond networks as a must to be taken into account.

5G technology operates in a wide range of frequency bands, including very high-frequency bands. This implies for example lower diffraction and higher outdoor/indoor penetration losses, which means that signals will have less ability to propagate around corners and penetrate walls. Such characteristics might give rise to some anomalies and problems on the network coverage and availability. Anomaly detection is the main focus in this thesis in which detecting an anomaly related to a characteristic and the main cause(s) contributing to the behavior variations of that characteristic are of interest.

1.2 Objective

As aforementioned, the objective of this study is to detect anomalies and diagnose the main causes of behavior changes in a 5G mobile network communication. The *latency* is the characteristic of interest in this study and thus, by anomaly we will address the high latency problem in the 5G networks communication.

In telecommunication networks *end-to-end latency* is defined as follow: “ The time that takes to transfer a given piece of information from a source to a destination, measured at the communication interface, from the moment it is transmitted by the source to the moment it is successfully received at the destination. “ [3] From the QoS perspective, it is of crucial importance to detect such anomalies in 5G networks since the ultra-low latency for communication is one of the most important requirements for 5G communications.

With developing the new generations, the mobile networks are meant to perform within the more strict requirements and standards in order to address the existing applications and demands as well as providing the newly introduced services and use cases. Central to those requirements is the transmission time of a message or packet through the network which has been attempted to shorten by implementing the new technologies. The aim is to decrease the latency as one of the main factors indicating the quality of the service. Considering such importance, an increase in the value of the latency beyond the allowed threshold will definitely degrade the predefined standards and the quality of the services, hence must be addressed as soon as detected. High latency in the network, therefore, is defined as an anomaly that must be taken into account seriously.

The standard value of end-to-end latency required to be fulfilled varies depending on the domain application (remote surgery, online gaming, intelligent transport Systems, ...), however, for most of the applications, it should be below 20ms.

To tackle the problem in this study, two approaches were implemented. Machine learning classification, specifically random forest classifier, was applied to detect the anomalies since the labels of the data instances were available. At this step, the focus was on training the classifier to be able to detect the high-latency instances. To perform the causal inference, the causal Bayesian networks were implemented through which the relevant algorithms were applied to discover a causal model from the data as well as estimating the causal effect of the main causes according to the causal structure.

1.3 Literature review

The problem of anomaly detection and root cause analysis has been a central subject of researches in various application domains for many years. [1] and [4] provide a comprehensive overview of the research and different techniques that have been developed aiming on anomaly detection and root cause diagnosis. In general, the existing techniques can be grouped into three dominant categories:

- 1) rule-based methods
- 2) classification
- 3) clustering

1.3.1 Rule-based methods

A rule-based model [5] is simply a collection of IF... THEN rules statements designed based on all available knowledge about the might-be happened faults and their relations to all possible causes. In other words, the previously known problem is related to specific conditions and criteria which as soon as they are fulfilled the recommended corrective workflow will be triggered according to the rules. As the description suggests, such a system can reliably work on anticipated faults whose nature and causes are entirely known by the experts. As a newly unspecified criterion occurs in the system, such a model fails to recognize it, and therefore the tool will remain ineffective in unforeseen circumstances.[6] Moreover, considering a very complex system creating and maintaining such a model would be a time-consuming and expensive task. [5] proposed a rule-based system in order to identify the cause of disconnection at the user level in 4G network. At the data collection phase, a number of key performance indicators such as signal power and signal quality were recorded. For the next step, a jumper of potential cause resulting in a connection release were specified including coverage hole, lack of dominant cells, cell edge, mobility problems, and interference. In order to specify the relation between each radio cause and abnormal values of the indicators, thresholds were defined for each of the

indicators discretizing them into low, normal, and high. The discretization consisted of designing the threshold of each key performance indicator based on their Xth percentile defined by the experts. Finally, based on the relation between radio causes and indicators a set of IF...THEN rules were designed for the criterion of each radio cause.

1.3.2 Classification

The second category of the methods involves implementing machine learning algorithms to build and train a model from the available labeled data, enabling the model to recognize the learned patterns in unseen data. [7] trained and implemented a decision tree model in order to perform QoS anomaly detection and root cause tracking in 5G networks. The data used in this study were a collection of key performance indicators (KPIs) collected from different levels of a communication path. These KPIs considered as the features and a target variable defined for each sample categorizing them into normal and abnormal labels. A decision tree model was then built and learned from the data to automatically detect the anomalies in the system. In fact, such a system is a rule-based system in which the rules are learned from the data, rather than specified by the experts. According to their suggestion, the root cause of the detected anomaly can be reversely tracked based on the analysis of the dependence between the network data and the network events.

[8] also proposed a method in which a supervised machine learning algorithm was applied in order to automatically identify dependencies between system events to help network operators recognizing the root causes of errors. During a period of time, the occurred events and the device in the network on which the event happened were recorded. For each event type, a dataset was created in a way that the dependency between this specific event and all other events in the network can be indicated. After collecting and preparation of the data, a random forest model was trained on each data set. For each event, a model was built to investigate if its occurrence could be predicted using other event types. Having fitted all the models, an influence matrix was created containing as many rows as the number of trained models and as many columns as the number of distinct events present in the observation windows. The feature importance was then assigned to each cell indicating the influence of each event on the emerging of each other. This influence matrix can then be analyzed by different visualization and summarization methods to identify the important causes of each event.

1.3.2.1 Bayesian networks One of the important subcategories of supervised learning used in causal inference is Bayesian networks. A Bayesian network, also known as Bayes network, belief network, decision network, Bayesian model, or probabilistic acyclic graphical model is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG).[9] Bayesian networks provide a compact representation of joint probability distribution of a large number of variables, which are difficult to be probabilistically described using the traditional specification.[10]

Bayesian networks have been employed in [11] to automate the process of finding the most likely failure mechanism for a given failure mode experienced by a complex hardware system. The workflow starts by collecting all the most relevant parameters across all different types of failure modes. At the next step, a graph structure must be constructed. The domain expert knowledge was employed to lay out the topology of the network. Having constructed the network, the conditional probabilities were then learned from the data. Once the parameters are learned, the joint probability distribution of the variables can be factorized and quantified. Then, given a set of known or so-called evidence variables it is possible to perform probabilistic inference to obtain the posterior probability distribution of a set of unknown or so-called query variables.

[12] implemented Bayesian network to present a methodology for automating troubleshooting tasks in mobile networks. The objective in that study was to detect a fault and relate the KPIs as a number of symptoms to the faults in order to identify the causes of the dysfunctioning in the system. First, a list of possible faults that could occur

and a list of symptoms (key performance indicators) was considered. Then assuming conditional independence between the symptoms given the cause, a Naive Bayesian network was built for each fault and the set of the symptoms. Then, the conditional probability distribution for the symptoms given the cause can be learned from the data. Finally, the joint distribution of the cause and the symptoms was computed. Having trained the models, once a dysfunction occurs, the most probable cause may be selected as the root cause in order to develop the required measures to heal the system.

[13], [14], [15] also proposed an automated troubleshooting system for mobile communication networks on the basis of the Naive Bayesian network. More or less a similar approach was followed to relate the KPIs to a list of faults in order to specify the probabilistic relationships between the KPIs and the faults.

An application of Bayesian networks in diagnosing the faults in Virtual Private Networks can be found in [16]. In this study the previous knowledge and Chi-Square independence test were used to construct the causal structure. Chi-Square test determines whether any two variables are dependent. The existing knowledge from previous experience also helps to determine if there is any causal relation between two variables. Using such a hybrid approach helps to constructing a causal model, which is much more closer to the true one.

In [17] an implementation of a causal Bayesian network to perform a causal inference study in biology can be found. The objective was to know which genes play a role in a certain phenotype, a disease status for instance. Having assumed that there is an underlying causal graph generating the available data, the authors firstly tried to discover that graph from the observational data and then estimate the causal effect of the genes accordingly. To relate this study to our case, we may assume that the KPIs in a network might be considered as a collection of genes, each of which describes the performance of a certain function of the network. We are interested in estimating their causal effect on a characteristic of network communication.

1.3.3 Clustering

Unsupervised models are algorithms to build models from unlabeled training data. In an unlabeled dataset, the target variable is not known. Clustering methods are categorized under the unsupervised models. Similarity between the points is the key based on which clustering is performed. The data is categorized into a number of clusters in a way that the within-cluster similarity of the points is maximized and between-clusters similarity is minimized. [6] presented an automated anomaly detection and diagnosis in 5G networks based on clustering methods. The workflow in this study starts with profiling the normal behavior. Having learned the normal behavior, any deviations from that can be detected and analyzed. The cross-correlation of the KPIs was considered to capture the joint distribution of them. Then using density-based and K-means clustering, the joint distributions were clustered into a number of partitions. Each cluster was specified by a normal distribution. For any new observed value p , the distance of p from all clusters can be calculated and aggregated, then by defining a threshold for the distance, this can be investigated if p is a normal value or not. For the diagnosis phase, the insights and experiences of the human experts were applied to create a diagnosis knowledgebase storing the already analyzed and labeled anomalies. This knowledgebase can then be used for finding and presenting the closest match between the detected anomaly and the labeled anomaly patterns.

Depending on the complexity of the system, one or a combination of the aforementioned methods can be implemented to build an automated tool in order to detect the anomalies in the system and to diagnose the most probable root causes. Supervised classification techniques can be implemented to detect the errors in the system assuming that enough data of all possible fault scenarios are available. In that case, the algorithm can learn effectively the important patterns from the data. In cases that labeled data is not available clustering can be taken into account. Clustering techniques rely on computing the similarities between the data points and so the outliers can be distinguished in

most of the cases. Bayesian networks can provide us a full picture of the dependence and independence relationships between variables. Given that a correct structure can be built, the probabilistic relationships between the different variables can be learned effectively from the data. In cellular networks, relationships among different probable causes of faults are uncertain and thus, they can be expressed in terms of probability.[18] Therefore discovering the joint probability distribution of the variables enables us to probabilistically describe the relationships between a fault and symptoms and select the most probable cause for observed abnormal behavior.

1.4 Questions to be answered

Considering the problem and the objective required to be addressed in this study and also bearing the important points extracted from the literature review in mind, the most important questions to be answered in this study may be listed as follow:

1. Are the key performance indicators (KPIs) of a 5G network able to predict the high latency values in a telecommunication network?
2. Which of those KPIs can be considered as the main causes that affect the latency?

1.5 Thesis organization

The report is organized into seven chapters following the introductory one. In chapter 2 a review of the mobile network configuration and an explanation of the technical background is briefed. The data structure and preprocessing steps are reported in chapter 3. Chapter 4 is dedicated to the outline and deeper explanation of the theory of the methods followed during the research work, providing the reader with an overview of the process. Chapter 5 presents the description of the implementation of the explained methodology in the previous chapter, and the obtained corresponding results are reported in chapter 6. Chapter 7 gives a discussion around the work and chapter 8 concludes the thesis highlighting the conclusions and answering the research questions.

1.6 Ethical considerations

The experiments to record the data were designed and carried out by Ericsson AI Research, and the whole data used in this thesis is considered confidential, so is not allowed to be published publicly. To avoid disclosing confidential information, variables are renamed and only general aspects of the data are reported.

2 Background

The intention of this chapter is to provide a basic understanding of a telecommunication network, in particular 5G mobile network, as well as present a brief description of the quality of services in the network.

2.1 Telecommunication network

A telecommunication network is a group of nodes that are used to exchange messages between the nodes. Depending on the technology used for the message passing, the data may be transmitted through either a circuit-switched network or a packet-switched network.[19]

In Circuit-switching transmission a dedicated communication channel (circuit) is established through the network by which the source-receiver ends before the communication starts. This circuit guarantees the full bandwidth of the channel and remains connected until the data transferring is accomplished.

Packet-switching, on the other hand, uses a different method. In this method, the data is split into smaller segments called packets. Each packet is assigned a header containing the destination address and a sequence number. The addressed packets are then routed to the destination through any channel on the network. The channel is occupied only during the transmission of the packet and will be made available for the other traffics to be transferred as soon as the packet transmission is completed. Examples of telecommunications networks include computer networks, the Internet, the public switched telephone network (PSTN).[20]

Each node in the network is assigned a network address for identification and locating it on the network. Examples of network addresses include the telephone number in PSTN, IP, and MAC addresses on the Internet. An Internet Protocol address (IP address) is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.[21]

While the IP address is the network software identifier, the media access control address (MAC address) uniquely identifies the devices on a network. Network Interface Card, known as NIC is essentially a computer circuit card that makes it possible for a computer to connect to a network. NIC Card's Manufacturer provides the MAC Address.

2.1.1 5G network architecture

Figure 1 depicts at a very basic level a 5G communication system deployed in a typical topology. A 5G system consists of user equipment (UE) connecting over the radio interface and via a radio access network (RAN), a core network (CN), and a transport network either to a peer UE or to a server application that typically is located on the internet. The RAN consists of functions mainly related to accessing and managing the radio interface resources. RAN functions are deployed on local (near antenna site) and in some cases on a regional level in dedicated RAN nodes. The CN holds functions for managing the UE connectivity to its UE peer or server application, including mobility management, gateways to external networks, authentication, authorization, accounting, etc. The transport network provides the routing/switching and aggregation between the network nodes in the RAN and the CN.[22]

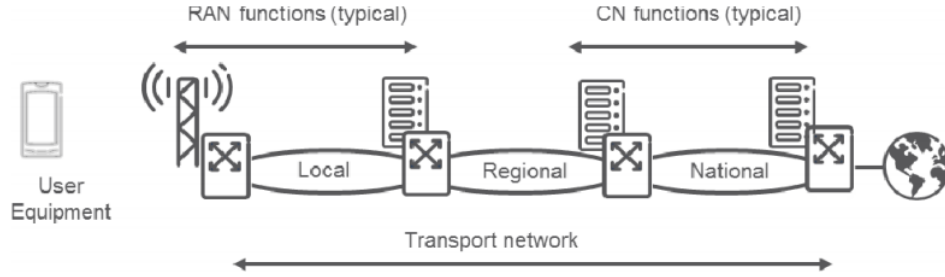


Figure 1: Fifth-generation communication system.[22]

2.2 QoS - Quality of service

Quality of service refers to any performance measures that manages and controls data traffic to reduce packet loss, delay, latency, and jitter on the network. Network resources management is achieved by setting priorities for specific types of data on the network.[23] During traversing the data across a network, some predictable and measurable services are required to ensure the quality of the data transmission. QoS is utilized by organizations to meet the traffic requirements and to prevent quality degradation due to packet loss, delay, and jitter.

Several parameters are to be considered to measure the QoS quantitatively:

Packet loss: Packet loss occurs when one or more data packets traveling across a network fail to reach the destination. Packet loss can take place due to either occurring errors in data transmission or network congestion. Network congestion happens when a network node or link is carrying more data than it can handle. Once network links become congested, the routers and switches start to drop the packets. Dropping the data packets during real-time communication such as a voice or video call will result in quality degradation.[23],[24]

Jitter:

In electronics, jitter refers to the deviation from the true periodicity of a periodic signal. In telecommunication networks, it happens as a result of network congestion and is the variation in arrival time of data packets. The timing drift between packets arrivals causes real-time communications to experience gaps between the transmission. The longer the data packets take to travel, the more jitter affects the quality. The standard jitter measurement is in milliseconds (ms). If receiving jitter is higher than 15-20ms, it can increase latency and result in packet loss, resulting in quality degradation.[25],[26]

Latency:

Latency is a measure of delay that is measured in milliseconds (ms). It is defined as the time it takes a packet to travel from its source to its destination.

Bandwidth:

In computing, bandwidth is the maximum rate of data transfer across a given path. For a network, it describes the maximum data transfer rate of a network or Internet connection. In other words, a network bandwidth defines the maximum capacity of a network communication link to transmit the data from a source point to a destination end within a given amount of time. Typically, bandwidth is expressed as a bit rate and measured in bits per second (bps). The term should not be confused with the speed with which bits of data move from one location to another. Bandwidth is used to describe network speeds and it measures how much data can flow through a specific connection at one time.[27],[28]

Reliability:

Reliability takes the ability of a network to carry out the communication into account. It can be thought of as delivering legitimate packets of data from source to destination within a reasonable amount of time. The reliability of a network depends on how successful the network is to transmit the data according to its predefined standards. This parameter is affected by the all aforementioned parameters.[29]

Availability:

Availability is defined as the probability that a given service is available (i.e., coverage). For instance, 99.99% availability means that one user among 10000 does not receive proper coverage.[30]

2.2.1 More about latency

The need to support different kinds of UEs (e.g., for the Internet of Things (IoT)), services, and technologies is the driving force to move towards a high-performance and highly efficient system. Based on this specification, the key objective in 5G development is to deploy scenarios that enable the system to address diverse market segments.[3] Such specified requirements gave rise to three main characteristics to be considered as the major design aspects in the 5G network, namely enhanced mobile broadband (eMBB), massive machine type communications (mMTC), and ultra-reliable low-latency communications (URLLC).

eMBB focuses on better system performance in terms of higher capacities and improved energy efficiency to support new user experiences such as those enabled by augmented reality/virtual reality.

mMTC provides an enhancement in the density of the devices that connect and communicate with each other through the network to support machine-to-machine (M2M) services with massive numbers of end devices.

URLLC attempts to improve network communications in terms of latency and reliability.

Low latency requirement is defined under the URLLC services. Latency is highly critical in some applications such as automated industrial production, control/robotics, transportation, health-care, entertainment, virtual reality, education, and culture. In particular, IoT(Internet of Things) is quickly becoming a reality which connects anything to any other thing anytime, and anywhere.[31]

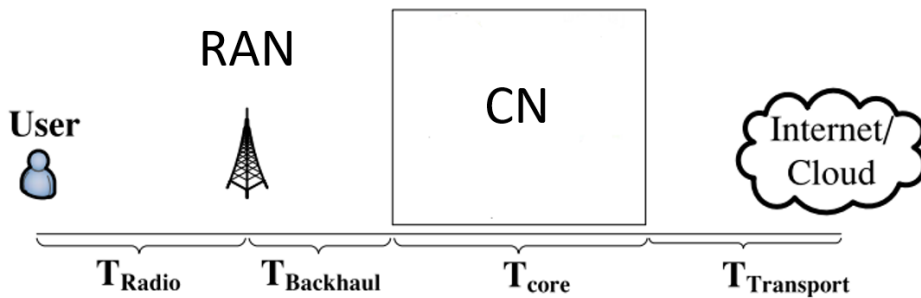


Figure 2: Latency contribution in E2E delay of a packet transmission.[31]

The requirements for URLLC services are end-to-end (E2E) characteristics, where the endpoints are the user equipment (UE) and the server issuing the service. The E2E latency is defined as the required time to transmit the data between the UE and the interface to the CN.[29]

The latency and reliability performance of the Core Network (CN) is clearly a key element to consider when E2E service-level requirements need to be addressed.

Figure 2 illustrates the total one way transmission time of a communication path.[31]

As the figure suggests, the total transmission time can be written as:

$$T = T_{Radio} + T_{Backhaul} + T_{core} + T_{Transport} \quad (1)$$

T_{Radio} is the packet transmission time between the access network and UEs.

$T_{Backhaul}$ is the time for building connections between access network and the core network.

T_{Core} is the processing time taken by the core network.

$T_{Transport}$ is the delay to data communication between the core network and Internet/cloud.

The E2E delay is then approximately given by $2 \times T$. [31]

3 Data

3.1 Raw data

The dataset employed in the research work has been provided by Ericsson AI Research and consists of end-to-end log files and radio traces recorded at the different parts of the network infrastructures. The 5G radio traces are used to store radio measurements and signaling messages in order to reflect user-level performance. [5]

The simplified illustration of a 5G network depicted in figure 2 is used in order to explain the sources from which data have been recorded. It depicts a path for communication that takes place in a network. Source and destination of the communication are shown at the endpoints and in between there exist access network and core network facilities and infrastructures.

During a message transmission, at different levels of the path, a number of logs and traces are generated, measuring the different key performance indicators (KPIs). Overall, the logs can be grouped into two categories:

1. End-to-end metrics: The log that is generated at the UE levels. The most important variable in these logs is RTT, stands for round trip transmission, which is an indicator of Round-trip packet delays.
2. Radio metrics: The logs generated at the radio interface and core network levels. The indicators at these levels measure the different aspects of the performance such as signal quality, signal power, signal to noise ratio, etc.

Figure 3 shows a few instances from an end-to-end log. As it can be seen there are a number of attributes in this log, however, **rtt** is the most important one for our study. RTT is defined as: *“The amount of time it takes for a signal to be sent plus the amount of time it takes for an acknowledgment of that signal to be received. This time delay includes the propagation times for the paths between the two communication endpoints.”* [32]

The RTT values are used to label the data as normal/anomaly. The required threshold for such categorizing is provided by a domain expert.

	timeStamp	icmp_seq	ttl	rtt
1	2021-02-19 12:25:51.648163080	2	60.0	9.19
2	2021-02-19 12:25:52.653563023	3	60.0	9.00
3	2021-02-19 12:25:53.658473969	4	60.0	9.21
4	2021-02-19 12:25:54.664808035	5	60.0	9.80

Figure 3: A few samples from end-to-end log.

Radio traces are a set of indicators measuring the QoS perceived by the user. They may be thought of as the performance indicators characterizing the network functioning. Any event that imposes a reduction in the quality of communication in terms of increasing the transmission time of a message affects the key performance indicators. The effects could be an abrupt increase or a rapid decrease in the values of those indicators which will eventually impact the value of latency. Therefore, these indicators are used as predictors to develop the anomaly detection and cause-diagnosing tool. As examples of such traces, RSRP and SINR will be explained.

- RSRP: Reference Signal Received Power, is a measurement of the received power level in a cell network.[33]
- SINR: Signal to Interference plus Noise Ratio. This KPI is defined as the ratio between the power of the desired data signal to the sum of the inter-cell interference powers plus noise. [15]

As the explanations convey, these radio measurements indicate the performance of the radio interfaces and the signaling facilities of the network.

3.2 Preprocessing and preparation

Similar to any other data science project, the workflow started with the so-called EDA. EDA stands for Exploratory Data Analysis and refers to the initial process of investigating the data set which provides preliminary insight into the data. The process involves summarizing the different variables statistically and visualizing them so that the structure of the data may be discovered and understood. Different tools to visualize the variables were employed such as histogram plot, line plot, box plot, etc. In addition, the different statistics such as mean, variance, correlation coefficient, etc. were used to describe and summarize the features statistically. The most important achievement obtained through this analysis was a reduction in dimensionality as the results enabled us to drop some of the attributes. It was figured out that a number of attributes in each dataset are zero-variance and so do not convey any information from the pattern recognition perspective. Moreover, it was found that there is a strong correlation between some of the attributes so it was enough to keep one among those strongly correlated features. Figure 4 illustrates an example of such a strong correlation. As it can be observed, three features are strongly correlated and all of them show the same variations over the time period.

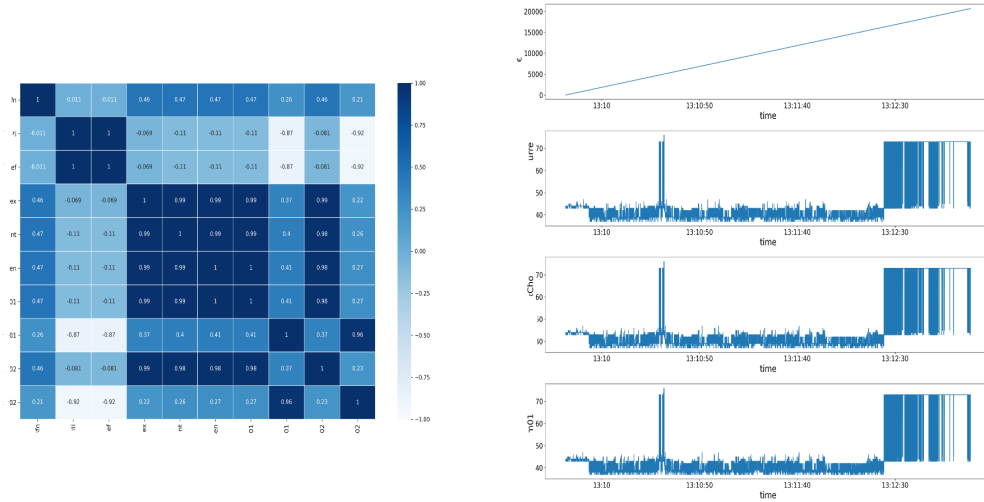


Figure 4: Highly correlated variables which are figured out through EDA.

Merging the different datasets together is the next step in data preparation in this study. In order to be able to assign the RTT value to each instance of the radio traces datasets, it is necessary to merge the end-to-end log to the other files. In addition, it is required to merge different datasets together in order to make use of all important indicators of different datasets. The original data were recorded in microsecond levels which makes it difficult to match the different logs based on time as each source of data has its own sampling strategy so the timestamps do not exactly match. Figure 5 shows such a disparity between the two datasets.

In order to overcome this difficulty, a floor operation was performed to change the resolution of all timestamps in all datasets to the millisecond level. Then the data was resampled to have a uniform sampling interval. To fill the produced NaN as the result of resampling, backward and forward filling were implemented and since no significant difference was observed, backward filling was implemented in which the next valid observation was used to fill the gap. At the end of this step, the RTT values are provided for each instance of the radio traces logs, hence enable us to label them as normal or abnormal according to a predefined threshold by a domain expert. Figure 6

	timeStamp	icmp_seq	ttl	rtt
0	2021-02-19 12:25:50.767848969	1	60.0	134.00
1	2021-02-19 12:25:51.648163080	2	60.0	9.19
2	2021-02-19 12:25:52.653563023	3	60.0	9.00
3	2021-02-19 12:25:53.658473969	4	60.0	9.21

	timeStamp	ver	ef	st	cd
0	2021-02-19 12:25:10.079904	1	90	0	1
1	2021-02-19 12:25:10.079924	1	90	0	2
2	2021-02-19 12:25:10.079943	1	90	0	3
3	2021-02-19 12:25:10.079962	1	90	0	4

Figure 5: Difference in sampling resolution.

illustrates an example of a merged dataset that resulted from the aforementioned process.

	Pro	CIndex	rs	m1	m2	m3	m4	rtt
timeStamp								
2021-02-19 12:38:50.094	0.0	38.0	81.0	2.0	1.0	3.0	9.0	16.185776
2021-02-19 12:38:50.095	0.0	38.0	81.0	2.0	1.0	3.0	9.0	16.093811
2021-02-19 12:38:50.096	0.0	38.0	81.0	2.0	1.0	3.0	9.0	16.001846
2021-02-19 12:38:50.097	0.0	38.0	81.0	2.0	1.0	3.0	9.0	15.909881
2021-02-19 12:38:50.098	0.0	38.0	81.0	2.0	1.0	3.0	9.0	15.817915
2021-02-19 12:38:50.099	0.0	38.0	81.0	2.0	1.0	3.0	9.0	15.725950

Figure 6: A merged dataset resulted from the preparation process.

Pandas [34], Matplotlib[35], and Seaborn[36] libraries were used to manipulate, analyze, and perform the preprocessing and preparation of the data.

4 Theory

This chapter aims to provide the description and the background theory of the important concepts and methods upon which the thesis work is built. It starts with introducing the anomaly detection followed by explaining the classification methods implemented in this study. Causal inference comes to be presented afterward, and finally, the employed evaluation metrics will be introduced.

4.1 Anomaly detection

The definition of normal behavior varies depending on the application domain. Every system is defined, designed, and created under several specific standard metrics and descriptions for the services which it is meant to provide. Such standards specify a normal behavior in the system that any significant deviation from that would give rise to an abnormality in the services. Anomaly detection, therefore, refers to the problem of identifying the patterns in the data that do not conform to normal behavior. [1]

Traditionally human experts and their knowledge and experiences were employed to perform troubleshooting in a system.

The data-driven approaches, on the other hand, rely on the pattern that can be learned from the data to establish a tool to identify any abnormality within the data. This group of approaches utilizes the different statistical and machine learning techniques trying to learn from the data and build a model to predict the normal or abnormal patterns in the data.

4.1.1 Machine learning approach

As alluded to above, the data-driven methods of anomaly detection involve implementing machine learning algorithms to build and train a model from the available data, enabling the model to recognize the learned patterns in unseen data.

Depending on the availability of the labels of the target variable, different algorithms can be implemented to recognize the abnormal samples in a dataset. In the problem under-discussion, the notion of a label for a data sample denotes whether the sample is normal or abnormal. Based on the availability of the labels, anomaly detection techniques can operate in either supervised or unsupervised machine learning methods.

Since the data in this study is considered as the labeled data, we mainly focus on the classification which is categorized under the supervised techniques.

4.1.1.1 Classification Classification lies at the heart of both human and machine intelligence. The problem involves a wide variety range of tasks such as image labeling, face recognition, spam detection, anomaly detection and voice recognition. The underlying task in all of these cases is to assign a class label to input data. Depending on the number of unique labels per target variable the classification task could be binary or multi-class. In cases in which the task is to divide the inputs into two categories is referred to as a binary classification. In cases the class labels are more than two, the task is called multi-class classification.

The classification technique involves learning a model, referred to as a classifier, from a set of labeled data instances and then, classify a test instance into one of the classes using the learned model. The dataset which is used to train a classifier is known as the training set, and the one employed to evaluate the model is called the test set. Building an automated system starts with choosing and developing a machine learning algorithm that can be trained on the training dataset. The trained model can then be implemented as an automated system to predict the target variable values for future data.

Deciding on the choice of the classifier is not a straightforward task. The best choice always can be made by trying evaluating several models to find the best one for the data at hand. Having tried several algorithms including decision trees, naive Bayes, nearest

neighbors, and random forest, the latter turned out to perform an acceptable task which motivated us to move forward with that.

4.1.1.1.1 Random forest classifier Random forest classifier is categorized under the ensemble methods in machine learning. Ensemble methods address a class of methods that create and combine several base estimators in order to improve the performance. In fact, instead of predicting based on a single estimator, a number of estimators are built, and then the predictions of those individual models are combined to obtain the final result. These methods are categorized into two main families: Averaging methods, and boosting methods. In averaging methods, several estimators are built independently, and then their predictions will be aggregated by taking an average. The intuition is that, on average, the combined estimator is usually better than any of the single base estimators because its variance is reduced. In boosting methods, the base estimators are no longer independent but are built sequentially each of which tries to reduce the bias of previous estimators.

Random forest is one of the main representatives of averaging methods. The base estimators in this method are the decision trees.

A decision tree is a supervised algorithm that attempts to learn a bunch of decision rules from the data to classify an instance to its proper category. As the name suggests, it is a hierarchical process starting at a root node and trying to divide the data based on an appropriate threshold. Each node represents a predictor or feature. The algorithm tries to find the best threshold in each feature which enables the model to divide the feature space into two regions. Each region should be as pure as possible. By successively splitting the feature space, space will be decomposed to distinct and non-overlapping regions, $R_1, R_2, \dots, R_j$. For every observation that falls into the region R_j , we then predict the most commonly occurring class of training observations in that region for the observation.[37]

Figure 7 illustrates the decision tree process for a dataset with two predictors, x_1 , and x_2 .

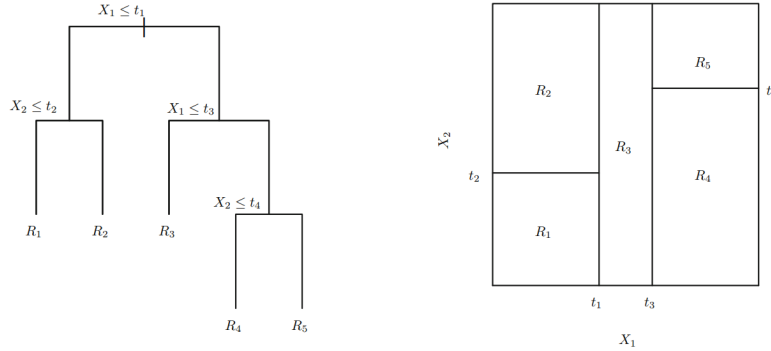


Figure 7: Decision tree corresponding to the partitioning of two-dimensional feature space(right) and the resulted output regions(left).[37]

As it was mentioned earlier a new observation in a given region is assigned to the most commonly occurring class of training observations in that region[37], therefore a classification error measure must be used when training the model. In practice, the Gini index and cross-entropy are the two main measures that are used as the error measures during the training.

The Gini index is defined by

$$G = \sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk}) \quad (2)$$

and is a measure of total variance across the K classes. $\hat{p}_{mk}$ represents the proportion of training observations in the mth region that are from the kth class. As it is noticeable, the Gini index value will be small if a region R_m contains data that is mostly from a single class. For this reason, the Gini index is referred to as a measure of node purity.

The cross-entropy is given by

$$D = - \sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk} \quad (3)$$

Similar to the Gini index, the cross-entropy will take on a small value if the mth node is pure. When building a classification tree, either the Gini index or the cross-entropy are typically used to evaluate the quality of a particular split which basically can be tuned as one of the hyperparameters.

One of the main problems associated with the decision trees is that they are prone to overfit the data. In fact, they suffer from the high variance issue. In other words, when building a decision tree classifier, it always tries to detect every variation within the training dataset which some of them are just random fluctuations and not a repeatable pattern within the data. Therefore it may not generalize properly. Here is the point random forest comes in to improve the performance of the decision trees. As the name suggests, a random forest is a forest of multiple trees learned and built to several sets of datasets taken randomly from the original dataset. In other words, a number of decision trees on bootstrapped training samples are built and then the predictions of those trees are aggregated as the final result. Each tree in the ensemble is built from a subset drawn from the training set. Bootstrap sampling is used for the sampling in which the samples are sampled with replacement. In addition, when splitting each node during the construction of a tree, the best split is found either from all input features or a random subset of them. These random samplings attempt to decrease the variance of the forest estimator and hence improving the performance.

Having trained the model, a prediction, in the case of regression, for an unobserved sample x' can be made by averaging the predictions from all the individual trees on x' [38]

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x') \quad (4)$$

where B is the number of trees in the forest, f is the prediction from each tree, and $\hat{f}$ is the resulted prediction from the forest.

In the case of classification the majority vote is taken as the result.

Hyperparameters

The most important hyperparameters to be tuned while training a random forest model are the *number of base estimators* and the *maximum features* used when learning a tree. The number of base estimators, in random forest is the number of trees in the forest. It is basically the number of individual trees which have to be learned to build up the forest. *Max_features* is the size of the random subsets of features to consider when splitting a node. It could be simply all the available features or just a random subset of them. Usually the lower number of features can result in reduction in variance, however the best choice must be made for each problem after performing a grid search and cross validation. Which *criterion* to be used for measuring the quality of a split is also another parameter that may be tried to select the best one. As alluded to earlier, *Gini index* and *cross-entropy* are the two criteria that can be selected. The *maximum depth* of a tree is another parameter to be mentioned as a hyperparameter. This parameter determines the number of nodes along the longest path of a tree from the root node to the farthest leaf node.[38]

Feature importance

Random Forest is able to rank the features according to how powerful they are in predicting the target variable. As mentioned before the features are used as the decision nodes in each tree, thus their relative importance with respect to the predictability of the target variable can be assessed. The aforementioned impurity measures are used to rank the features and it is called the impurity-based feature importance. In other words, the ability of each feature to decrease the impurity is taken into account. The reduction in impurity can be calculated by computing the difference between the successive nodes impurities. The features selected as the top decision nodes of a tree result in a greater reduction in impurity. The importance of a feature is computed as the total reduction of the impurity. For each tree in the forest it is computed for each feature. So the average over all trees in the forest is the measure of the importance of a feature. The values are normalized and the relative importance for each feature is reported.[38],[39]

4.1.1.2 Evaluation metrics The test set used to evaluate the model is imbalanced, therefore, considering the anomaly detection objective the Precision and Recall scores chosen to be reported. Consider the confusion matrix in figure 8.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive(TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Figure 8: Confusion matrix for a binary classification with two classes, positive and negative.

Precision

Precision can be measured for each class label. For the positive label it is defined as:

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

It determines the proportion of correctly classified samples into the positive class.

Recall

Recall or sensitivity is also defined for a particular class. It returns the percentage of the samples belong to an actual class which have been correctly assigned into that class. For the positive class it is defined as:

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

F1

In case both Precision and Recall are on interest, F1 may be considered. F1 is a harmonic mean of them.

$$F1 = \frac{2.Precision.Recall}{Precision + Recall} \quad (7)$$

4.2 Causal inference

A cause may be defined as any factor that if it exists, will be followed by a consequent effect. David Hume defined “cause” as:[40]

“We may define a cause to be an object, followed by another, [. . .] where, if the first object had not been, the second had never existed.”

When observing and recording a number of variables, what we do is just collecting the values which those variables take on. In most studies, the final goal is to understand the underlying mechanism by which the variables took on their values.[40]

Learning about the underlying mechanisms of the generation of the values of a variable gives us a deep insight into the system, and thus more effectively will we be able to define a generative model.

Causal inference is to study and understand the causal relationships between a set of variables. Discovering the cause-effect relationships goes beyond the dependence relationship which is at the heart of statistical inference. In causal inference, we would like to answer the question about the mechanisms by which a variable comes to take on values.[41]

The main difference between a causal relationship and a dependence relationship is that causality is asymmetric. In causality there exists a “cause” that has an impact on the “effect”. The roles of the variables cannot be switched.

Consider two variables X and Y as cause and effect respectively. The graphical representation of this relationship would be as:

$$X \longrightarrow Y$$

In fact, the direction of the arrows will be important in a causal relationship. It cannot be reversed.

On the other hand, the dependency relationship is symmetric. In this case, the direction of the arrow does not transmit any causal information. In other words, the only information obtained in this regard is the correlation of X and Y .

A causal model tells us about the cause-effect relationship, but also it reveals the dependence and independence relationships as well.

The outstanding characteristic in causal inference is “manipulation”. It is interested to see whether any change occurs in a system, having manipulated a variable. Manipulation or intervention is a key term in causal inference. The objective is to answer the question of what will be the value of a variable if the value of another variable is manipulated. Answering this question will give us a deep knowledge of the mechanisms by which variables take on values, thus building the prediction model will be more straightforward.

4.2.1 Correlation vs. causation

Figure 9 [40] illustrates a nice example that conspicuously elucidates the disparity between correlation and causation.

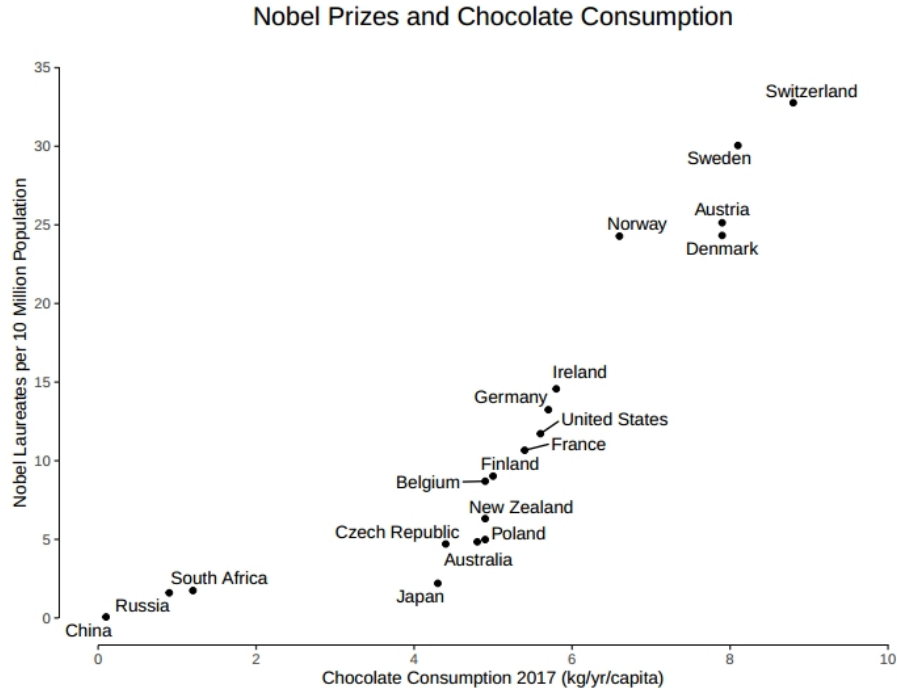


Figure 9: Relationship between chocolate consumption and the number of Nobel Laureates per country.[40]

The figure shows a strong positive correlation between chocolate consumption and the number of Nobel Laureates per country. Although the joint distribution depicted in Figure 1 suggests a strong dependency between two variables, none of the variables can be thought of as a cause impacting the other one. While the strong correlation between chocolate consumption and Nobel Laureates helps to create a prediction model, a causation structure cannot be inferred directly. A common unobserved reason might exist that is a cause in increasing the two variables. Three different possibilities in such a problem can be investigated which stated in the Reichenbach common cause principle:[40],[42]

“ If two random variables X and Y are statistically dependent, $X \not\perp Y$, then either

- a) X causes Y ,
- b) Y causes X , or
- c) There exists a third variable Z that causes both X and Y . Further, X and Y become independent given Z , i.e., $X \perp Y | Z$.”

Conducting an intervention on one of the variables can lead us to break this uncertainty. In the aforementioned example, we could force the population of a country to consume more chocolate and study whether this increases the number of Nobel prize winners.[40] Such an intervention can only be achieved through an experiment. Conducting an experiment, allows to manipulate the value of a variable, setting into a fixed value, and study its consequences on the other variable. In practice, however, collecting experimental data is hardly possible. What is always can be collected is what is being observed and is referred to as observational data. Causal Inference tries to provide the tools to be able to infer the causality from the observational data.

4.2.2 Different levels in causal inference

The relationship between variables can be studied at three levels.[40] At each level, a different problem is aimed to be addressed, and thus three different probability distributions may be inferred.[41] *Association*, *Intervention*, and *Counterfactual* are the levels that correspond to the actions *seeing*, *Doing*, and *Imagining* respectively.

Association

The lowest level of the causal hierarchy is related to the association or observation. The prototypical action corresponding to this level is “seeing”. At this level, we merely observe and record what has occurred. In other words, the values that the variables came to naturally take on is recorded. The information that can be captured at this level tells the story about a subpopulation since we observe a natural behavior of some random variables over a subpopulation, i.e. a particular group of people for instance.[41] There is no manipulation at this level as none of the variables is forced to take on a specific value in order to monitor the effects of that on the other variables.

The probability distribution that is assigned to this level is conditional probability distribution, denoted by $P(Y|X)$. The dependence relationship between X and Y can be captured by this conditional distribution. The question which can be addressed by this distribution is “*what is the probability of $Y=y$ given that X has taken on value x* ”. Therefore this distribution can only be used to predict the value of a variable given the value of other variables.

Intervention

The next level of the hierarchy belongs to the Intervention. The corresponding action at this level is “doing”. By “intervention” a variable is forced to take on a specific value in order to study whether it is a cause of an effect. If possible, an intervention can reveal whether a variable is affected by another variable or not. At this level, population-level questions can be addressed such as “what will happen if we force every patient to take the treatment?”.[40] In this case, we no longer record the natural behavior of a variable, instead we force it to take on a specific value. If so, the population is said to be manipulated.[41]

Manipulated probability distribution [41], denoted by $P(Y|do(X = x))$ [43] is the probability density that can be captured at this level. Depending on the causal relationship between the variables, a manipulated density may or may not be equal to the observational conditional density.

As alluded to, a conditional probability distribution can be estimated from a subpopulation, known as a sample, from a population. In contrast, the gold standard for estimating a manipulated density is by conducting an experiment.[41]

“*What if*” would be the typical question to be answered at this level, such as “*what will be the value of Y , if I set the value of X into x* ?” An action, “*doing*”, takes place at this level to address the problem.

As mentioned, performing an experiment is hardly possible. Consider a study aiming to investigate if smoking is the cause of developing lung cancer. Observation means just recording the data related to those who choose by themselves to smoke, which obviously would be a sample or a subpopulation from the population. Intervention, on the other hand, means to force a large group of people randomly selected from the whole population, regardless of their willingness to smoke or not, to smoke to see if it has any effect on lung cancer. Performing such an experiment is not feasible, even if so it is not ethical to force everyone to start smoking. Therefore, drawing the causality conclusions and building a causal model is the problem that has been focused to be addressed by Causal Inference.

Counterfactual

At the highest level of the hierarchy, counterfactuals come to play.

The typical question at this level is “*What if I had acted differently,*”, hence requiring retrospective reasoning.[44] The corresponding activity to counterfactuals is “*imagining*”.[40] Strong assumptions are required to answer the individual-level questions arisen regarding this level of causal inference. Considering a study on how effective a treatment is to curing a disease, a counterfactual-level question regarding this example could be: “*would the patient have recovered if we had given him/her the treatment, even though he did not recover and has not received the treatment?*”[40]

Counterfactual probability density, denoted by $P(y_x|\acute{x}, \acute{y})$ would be the probability distribution associated with this level. It stands for “*The probability that event $Y = y$ would be observed had X been x , given that we actually observed X to be $\acute{x}$ and Y to be $\acute{y}$* ”.[44]

Figure 10 summarizes the aforementioned causal hierarchy. This hierarchy is directional in the sense that a question at the lower level could be answered given that the upper-level model has been constructed, however, the upper-level queries is not possible to be addressed by the lower-level models.

Level (Symbol)	Typical Activity	Typical Questions	Examples
1. Association $P(y x)$	Seeing	What is? How would seeing X change my belief in Y ?	What does a symptom tell me about a disease? What does a survey tell us about the election results?
2. Intervention $P(y do(x), z)$	Doing, Intervening	What if? What if I do X ?	What if I take aspirin, will my headache be cured? What if we ban cigarettes?
3. Counterfactuals $P(y_x x', y')$	Imagining, Retrospection	Why? Was it X that caused Y ? What if I had acted differently?	Was it the aspirin that stopped my headache? Would Kennedy be alive had Oswald not shot him? What if I had not been smoking the past two years?

Figure 10: The causal hierarchy. Questions at level i can be answered only if information from level i or higher is available.[44]

4.2.3 Causal graphical models

A causal model is a model that represents the causal relationship over a set of variables. In order to be able to describe the causal relationship over a population, a framework is required to describe the causal mechanism of a system. Various techniques have been specified to map the causal relationships into a framework enabling us to quantitatively study the causality over a set of random variables among which the causal Bayesian networks and structural equation models are the most commonly used.[41]

4.2.3.1 Structural equation models(SEMs) Structural equation models specify the value of the effect variable as a function of the values of its causes, typically including some unmeasured noise terms.[41] With this definition, the set of random variables in structural equation models can be categorized into two groups: The error terms, and the substantive variables. The substantive variables are the variables of interest that are not necessarily observed. The value of each substantive variable can be estimated as a function of the value of other substantive variables and a unique error term. Consider the following structure:

$$X \longrightarrow Y$$

The SEM describing the causal relationship between X and Y could be identified as:

- $X = \varepsilon_X$
- $Y = f_Y(X, +\varepsilon_Y)$

This model tells us that X is generated independently of Y , however, Y is a function of its cause, i.e., X .

An SEM can be associated with a graphical model that represents the causality in the form of a directed acyclic graph.

4.2.3.2 Causal Bayesian networks A Bayesian network, also known as Bayes network, belief network, decision network, Bayesian model, or probabilistic acyclic graphical model is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph(DAG).[45]

A DAG denoted as G , consists of a set of nodes or vertices, V , and directed arrows in between them. The nodes represent the variables and the directed edge between them captures the causal relationships between the variables.

Consider the DAG in figure 11 for the variables A, B, C , and D ;

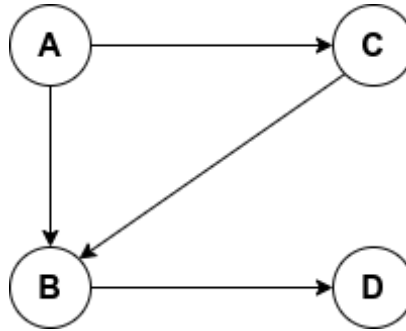


Figure 11: A directed acyclic graph over the variables A, B, C , and D .

This graph consists of four nodes representing the variables and four directed edges between them.

Two nodes are said to be *adjacent* if there is an edge between them. For instance, A and C are adjacent nodes as there is an edge between them, however, A and D are not adjacent.

For the two adjacent nodes A and C , A is said to be the *parent* of C as there is a directed edge from A to C . In this case C is known as the *child* of A .

A *path* between two variables in the graph is a sequence of non-repeating edges (oriented in either direction), starting from one of the variables and ending at the other variable. A *directed path* is a path whose edges all point in the same direction. In figure 11, $B \leftarrow A \rightarrow C$ is a path, and $A \rightarrow B \rightarrow D$ represents a directed path.

A *descendent* of a node A is a node in the graph if there exists a directed path starting from A which ends at that node. In figure 11, D is a descendent of A .

Based on the orientation of edges in a path, three different structures can be distinguished in a graph: *chain*, *common-parent*, and *v-structure*.

A *chain* is a sequence of edges that all of them point to the same direction. $A \rightarrow B \rightarrow D$, and $A \rightarrow C \rightarrow B \rightarrow D$ are the examples of this kind of pattern.

A *common-parent* pattern is the one in which there are directed edges from a common parent to its descendents. $B \leftarrow A \rightarrow C$ in figure 11 represents such a pattern.

A *v-structure* is a structure in which two parents point to the same child. $A \rightarrow B \leftarrow C$ is an example of this kind of pattern. B is referred to as a *collider*. In this case if the parents are adjacent the collider is said to be a shielded, otherwise it is called unshielded collider. The difference is illustrated in figure 12.

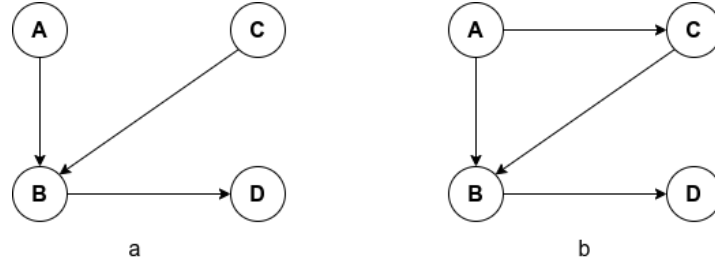


Figure 12: a) B is an unshielded collider. b) B is a shielded collider.

With the aforementioned explanations the Bayesian network is defined as:

A Bayesian Network over a finite set of disjoint random variables is defined as a pair of (G, P) that G is the directed acyclic graph whose nodes are the random variables, and P describes the joint probability distribution capturing the probabilistic distribution of the variables. The graph represents the causal structure and the Bayesian Network represents the causal model of the system.[45]

The so-called *d-separation* criterion [43] allows us to infer the conditional independence relations between variables in a graph.

d-separation

According to [43] d-separation is defined and formalized as follow:

“Let X , Y and Z be three disjoint subsets of nodes in a directed acyclic graph G , and let p be any path between a node in X and a node in Y . Then Z is said to block p if there is a node w on p satisfying one of the following two conditions:

- (i) w has converging arrows along p , and neither w nor any of its descendants are in Z , or,
- (ii) w does not have converging arrows along p , and w is in Z . Further, Z is said to d-separate X from Y , in G , written $(X \perp\!\!\!\perp Y|Z)_G$, if and only if Z blocks every path from a node in X to a node in Y ” [43]

Converging arrows in the definition refers to the v-structure pattern in the graph.

Considering this definition, Z is called a conditioning set which includes the nodes that we condition on (it can be empty). In simple words, the d-separation criterion can be summarized in two rules:[40]

- Conditioning on a non-collider along a path blocks that path.
- A collider along a path blocks that path. However, conditioning on a collider (or any of its descendants)unblocks that path.

If Z blocks a path between X and Y , we say that Z d-separates X and Y . In statistical language we say that X and Y are conditionally independent given Z , denoted by $(X \perp\!\!\!\perp Y|Z)_G$. The subscript G indicates that the conditional independence is inferred from the graph using d-separation definition.

To clarify this definition consider the graph demonstrated in figure 13.

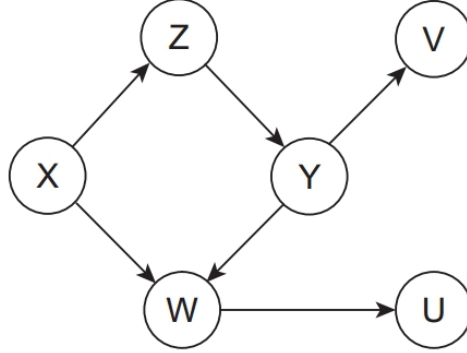


Figure 13: A DAG depicting a causal structure to clarify the d-separation concept.[40]

A number of paths can be distinguished in this graph. First take the path $X \rightarrow Z \rightarrow Y$. As explained earlier, this is considered as a chain pattern. Based on d-separation rules conditioning on an empty set, then X and Y will be dependent, however conditioning on Z , then Z blocks the path and so X and Y will be independent, and it is denoted as $(X \perp\!\!\!\perp Y|Z)_G$. Similar scenario can be explained for the path $Z \rightarrow Y \rightarrow V$, and so $(Z \perp\!\!\!\perp V|Y)_G$ as this is also a chain structure, and so it is about $Y \rightarrow W \rightarrow U$; $(Y \perp\!\!\!\perp U|W)_G$.

A common-parent structure can be noticed in the graph, $W \leftarrow X \rightarrow Z$. The independence relationship for this pattern is quite similar to what was explained for the chain pattern. Z and W will be d-separated having conditioned on X . In other words, X blocks this path between Z and W and it is denoted by $(Z \perp\!\!\!\perp W|X)_G$.

Note that there is another path between Z and W which passes through Y . This path is a chain structure. Therefore, Z and W are conditionally independent if both paths are blocked and this can be achieved by conditioning on X and Y , $(Z \perp\!\!\!\perp W|X, Y)_G$. $W \leftarrow Y \rightarrow V$ is also another common-parent that can be distinguished in the graph.

Finally, a v-structure can be seen which is different from the other two patterns. $X \rightarrow W \leftarrow Y$ can be noticed as a v-structure and W is a collider in this pattern. According to the d-separation rules, conditioning on a collider or its disconten, the path which contains the collider will be unblocked. So in the path $X \rightarrow W \leftarrow Y$ X and Y are not conditionally independent given W and it is denoted by $(X \not\perp\!\!\!\perp Y|W)_G$.

Note that irrespective of the orientation of the edge, two adjacent nodes are always dependent conditioning on empty sets. For example, In figure 13 X and Z are always dependent as they are adjacent, $(X \not\perp\!\!\!\perp Y)_G$.

In summary, among the three different patterns, chain and common-parent share a common independence relationship. V-structure, on the other hand, reflects the relationship in an opposite manner.

Note that the aforementioned conditional independencies between variables were concluded from the graph. In particular, that d-separation provides an independence model $(\perp\!\!\!\perp)_G$ defined on graphs.[40] To connect this to the probabilistic independence relationships *Global Markov condition* is assumed. It states that if the nodes X and Y are d-separated by Z in the graph G , the corresponding random variables X and Y are conditionally independent given Z . This implies that all conditional independencies in the data are represented in the graph.[40]

$$(X \perp\!\!\!\perp Y|Z)_G \Rightarrow (X \perp\!\!\!\perp Y|Z)_P \quad (8)$$

This assumption implies the following factorization of the joint probability distribution over a set of disjoint variables, $(X_1, X_2, \dots, X_n)$:

$$p(X_1, X_2, \dots, X_n) = \prod_{i=1}^n p(X_i | \text{Parents}(X_i)) \quad (9)$$

This factorization states that every variable, X , in a directed acyclic graph, is independent of its non-descendants given its parents.[46]

Graphs that have the same d-separation properties are usually called “*Markov equivalent*” and imply the same conditional independence relations; a collection of all directed acyclic graphs that are Markov equivalent is a *markov equivalence class (MEC)*. [46]

With this explanation, chain and common-parent structures are in the same MEC as they describe the same conditional independence relations. A v-structure, however, is in a different MEC. As a result each MEC contains several DAGs which have the same skeleton and the same v-structures. The *skeleton* of a directed graph G is the undirected graph that is obtained from G by replacing the directed edges with undirected edges.

An equivalence class can be uniquely specified by a completed partially directed acyclic graph (CPDAG)[18]. All the graphs within an equivalence class have the same skeleton, which is represented by the the CPDAG skeleton. The directed edges in CPDAG are those which are in common between all the graphs within an equivalence class, and the undirected edges correspond to edges whose directions are not the same in all graphs.

Although the d-separation properties and Markov assumption is necessary to connect the conditional independencies captured in the graph to the joint probability distribution over the variables, it is not sufficient to make the connection between them completely. The probability distribution may have additional conditional independence relations that are not entailed by d-separation applied to a graph.[46] Therefore another assumption is required which is referred to as *faithfulness* assumption. This assumption states the converse of the Markov condition, i.e. if a variable X is independent of Y given a conditioning set Z in the probability distribution P , then X is d-separated from Y given Z in the graph G .

$$(X \perp\!\!\!\perp Y|Z)_P \Rightarrow (X \perp\!\!\!\perp Y|Z)_G \quad (10)$$

Therefore, Markov and faithfulness assumptions together allow us to make a complete connection between the conditional probability distribution and the causal connectivity relations within the graph G .

$$(X \perp\!\!\!\perp Y|Z)_P \Leftrightarrow (X \perp\!\!\!\perp Y|Z)_G \quad (11)$$

4.2.3.3 Graphs as models of interventions A Bayesian network at the first level is quite useful to extract the information about the conditional probability independence relations between a set of variables. As it was explained, $A \rightarrow B$, and $B \rightarrow A$ both reflect the same conditional probability relations and hence are in a same Markov equivalent class, however, they describe completely different causality relations. In a cause-effect relation, the direction of the arrow always points from the cause to the effect, i.e., for a cause-effect relation we always have:

$$cause \longrightarrow effect$$

According to Pearl causal hierarchy [44], we need to build an intervention model to be able to perform a causal inference. To this end, J. Pearl [43] introduced a set of tools which allow us to map the interventional distribution to the probabilistic observational distribution. He showed that for a set of finite disjoint variables, each child-parent family in a directed graph G represents a deterministic function

$$X_i = f_i(pa_i, \varepsilon_i), (i = 1, \dots, n) \quad (12)$$

Where pa_i stands for the parents of variable X_i in G , and ε_i represents the mutually independent, arbitrarily distributed random disturbance.

An external intervention can be encoded as an alteration to a selected subset of functions while keeping the others intact. [43]

The simplest type of intervention, called atomic intervention, is the one in which a single variable, say X_i , is forced to take on some fixed value, x_i . Denoted by $do(X_i = x_i)$, this intervention amounts to removing the equation $X_i = f_i(pa_i, \varepsilon_i)$ from the model, and substituting x_i for X_i in the remaining equation. In other words, we set the value for X_i equal to x_i everywhere in the structural equations where X_i has taken a role. Consequently, the incoming arrows to X_i must be removed in the graph since forcing it to take on a fixed value will make it independent of its cause(s). Such an intervention results in a model which describes the behavior of the system under the intervention $set(X_i = x_i)$ ($do(X_i = x_i)$). [43]

Solving this model for the distribution of X_j , yields the causal effect of X_i on X_j , denoted by $p(x_j|do(X_i = x_i))$.

According to the aforementioned reasoning, J. Pearl presented the definition of causal effect as follow:

“Given two disjoint sets of variables, X and Y , the causal effect of X on Y , denoted $p(y|do(x))$, is a function from X to the space of probability distributions on Y . For each realization x of X , $p(y|do(x))$ gives the probability of $Y = y$ induced on deleting from the model all equations corresponding to variables in X and substituting x for X in the remainder.” [43]

The immediate implication of this definition is that, given a true graph of a causal model in which all parents of manipulated variables are observable, the post-intervention distributions can be inferred from the pre-intervention distributions. Therefore, under such assumptions, the effects of interventions from observations can be estimated.

However, the problem arises in situations that not all of the parents of the manipulated variables are observable. For an unobservable parent, the estimation of $p(x|pa(x))$ is not possible. Therefore Pearl provided simple graphical tests for deciding when $p(x_j|do(x_i))$

is estimable in a given model. The *identifiability* was defined by Pearl to address this investigation as follow:

“The causal effect of X on Y is said to be identifiable if the quantity $p(y|do(x))$ can be computed uniquely from any positive distribution of the observed variables that is compatible with G .” [43]

An accurate estimation of a causal inference requires to block all the non-causal paths in G to let the association flow only through the causal paths. In other words, the covariates in the non-causal paths impose a bias on the value of the effect variable (y) in $p(y|do(x))$, which is required to be avoided in order to estimate the causal effects as accurate as possible from observational data. Such covariates are called *confounder* and adjusting for confounder is known as *adjustment for avoiding confounder bias*. With testing for identifiability, the aim is to investigate whether such an adjustment is possible given a causal structure G . To this aim, Pearl proposed a criterion called *back-door criterion*.

The back-door criterion

The *back-door criterion* is defined as follow:

“A set of variables Z satisfies the back-door criterion relative to an ordered pair of variables (X_i, X_j) in a directed acyclic graph if:

- a) No node in Z is a descendant of X_i , and
- b) Z blocks every path between X_i and X_j which contains an arrow into X_i .” [43]

To illustrate the concept, let's consider the causal structure depicted in figure 14. In this model $Z_1 = X_3, X_4$ and $Z_2 = X_4, X_5$ meet the back-door criterion as none of X_3, X_4 , and X_5 are a descendant of X_i and conditioning on Z_1 and on Z_2 block the paths between X_i and X_j . Nevertheless $Z_3 = X_4$ does not fulfill the conditions of the backdoor criterion, because conditioning on X_4 unblock the path $(X_i, X_3, X_1, X_4, X_2, X_5, X_j)$. Note that X_4 is a collider and conditioning on a collider unblock a path in which it is located.

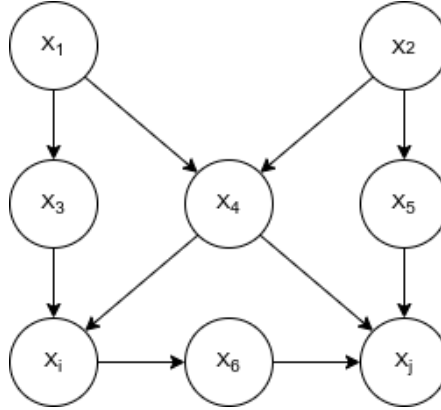


Figure 14: A causal structure to illustrate the back-door criterion.[43]

If a set of variables Z satisfies the back-door criterion relative to (X, Y) , then the causal effect of X on Y is identifiable and is given by the formula

$$p(y|do(x)) = \sum_Z p(y|x, z)p(z) \quad (13)$$

This equation states that given a causal structure, to infer the cause-effect relation between variables X and Y , it is required to block the non-causal path between them by adjusting for the confounder, Z , which fulfill the back-door criterion in order to let the association flow only through the causal path.

4.2.3.4 Causal effect estimation Consider $p+1$ continuous variables $X_1, X_2, \dots, Y$ (also referred to as $X_1, \dots, X_{p+1}$). As aforementioned, under the Markov assumption, the joint distribution of these variables generated from a DAG can be factorized as: [17]

$$f(x_1, x_2, \dots, x_{p+1}) = \prod_{j=1}^{p+1} f(x_j | pa_j) \quad (14)$$

where pa_j denotes the parents of X_j . As alluded to earlier, in order to estimate the causal effect of a variable on the outcome variable, we need to compute the intervention distribution. Assuming that the causal effect of X_i on Y is of interest. Using the ‘do’ operator, it means that estimation of $f(y | do(X_i = x'_i))$ is of interest. It denotes the distribution of Y , having enforced X_i to take on value x'_i . The joint intervention distribution can then be factorized as:

$$f(x_1, x_2, \dots, x_{p+1} | do(X_i = x'_i)) = \begin{cases} \prod_{j=1, j \neq i}^{p+1} f(x_j | pa_j)|_{x_i=x'_i}, & \text{if } x_i = x'_i \\ 0, & \text{otherwise,} \end{cases} \quad (15)$$

The factorization shown in the formula 15 specifies that the joint intervention distribution can be factorized as the product of the conditional distributions only if the variable X_i is set to x'_i . Thus, the intervention distribution $f(y | do(X_i = x'_i))$ can be computed by marginalizing out the $X_1, \dots, X_p$ in 15. This distribution is given by:

$$f(y | do(X_i = x'_i)) = \begin{cases} f(y), & \text{if } Y \in pa_i, \\ \int f(y | x'_i, pa_i) f(pa_i) dpa_i, & \text{if } Y \notin pa_i, \end{cases} \quad (16)$$

To quantify the causal effect of X_i on Y , the expected value of Y when X_i is enforced to take on specific values by the intervention is used. In mathematical form it is given by:[47]

$$E(y | do(X_i = x'_i)) = \begin{cases} E(y), & \text{if } Y \in pa_i, \\ \int E(y | x'_i, pa_i) f(pa_i) dpa_i, & \text{if } Y \notin pa_i, \end{cases} \quad (17)$$

pa_i in the last two equations denote the variables that are the parents of X_i in the graph.

Using the expected value of Y to indicate the causal effect, the average causal effect (ACE) or total causal effect of X_i on Y is defined as:

$$ACE = E[Y | do(X_i = x'_i + 1)] - E[Y | do(X_i = x'_i)] \quad (18)$$

It simply implies that the average causal effect of X_i on Y is the amount of change in the expected value of Y after enforcing the variable X_i to take on two different values.[47]

Using the derivation, this definition is given by:[47]

$$\frac{\partial}{\partial x} E(Y | do(X_i = x))|_{x=x'_i} \quad (19)$$

To simplify the expression some assumptions should be considered. Assume that $X_1, \dots, X_p, Y$ are jointly Gaussian, and we are interested in the causal effect of X_i on Y for $i = 1, \dots, p$.

Having considered the Gaussianity assumption, the $E(Y | x_i, pa_i)$ can be expressed as a linear regression as following:

$$E(Y | x_i, pa_i) = \gamma_0 + \gamma_i x'_i + \gamma_{pa_i}^T pa_i \quad (20)$$

where γ s are the regression coefficients.

Hence,

$$\int E(y|x'_i, pa_i) f(pa_i) dpa_i = \gamma_i x'_i + \int \gamma_{pa_i}^T pa_i f(pa_i) dpa_i \quad (21)$$

is linear in x'_i .

Therefore, under the considered assumptions and having computed the regression coefficients for the mentioned linear regression, the causal effect of X_i on Y is given by γ_i , which is simply the regression coefficient of X_i in the regression of Y on X_i and pa_i . [17]

4.2.3.5 Causal discovery So far it have been assumed that the true DAG associated with observational data is known. Nevertheless, in practice the true DAG corresponding to a system is not known and must be either constructed based on domain expert knowledge or must be learned from data. The process of disclosing the causal relations by analyzing the observational data and construction the causal structure accordingly, is known as causal discovery which have been a field of research for a long time. [9,10] In order to discover a true causal model from which the data has been generated, it is ideally required to identify all of the causal relations of interest hidden in the dataset. Extracting such information from the observational data is not always practical. The introduced algorithms for causal discovery rely on different assumptions without which the problem of revealing the causal information from the observational data will be a difficult task, if not impossible. Based on the considered assumptions, algorithms may be categorized in different groups which the following will briefly present them.

4.2.3.5.1 Constraint-based and Score-based causal discovery methods As aforementioned, to connect the conditional probability relations to the independence relations entailed by a DAG, Markov and faithfulness assumptions must be satisfied. In that case a DAG, at the first level of analysis, encodes the joint probability distribution of a set of disjoint variables as a factorization of conditional probability distribution of variables. Assuming no unobservable variable that has an unobservable direct common cause on the observed variables in the data, and given that the faithfulness and Markov assumptions are satisfied, the conditional independence relations inferred from the data can be used to discover a Markov equivalence class (MEC)

The Constraint-based algorithms

This family of causal discovery relies on statistical independence tests to estimate the relation between variables.

As one the most applicable constraint-based methods, the PC algorithm [18] will be explained.

PC algorithm is one of the algorithms providing a search method based on statistical tests to estimate the conditional independence relations from data in order to identify the MEC. When no unobservable variable exist, PC relies on the fact that under the causal Markov condition and the faithfulness assumption, two variables are directly causally related (with an edge in between) if and only if there does not exist any subset of the remaining variables which can d-separate them [18]. The algorithm starts with a complete undirected graph in which there exists an undirected edge between all pairs of variables. At the next step, it removes the edge between variables that are unconditionally independent, i.e. two variables that are independent given an empty set. Then, for each pair of variable, A and B, having an edge between them, if there exist a third variable, C with an edge connected to either of them, the edge between A and B is eliminated if $A \perp\!\!\!\perp B | C$.

The process of conditional independencies checking continues for all adjacent variables. Finally, the algorithm starts to orient the edges. First v-structures are distinguished and edges are oriented such that they point to the colliders. The remaining edge must be directed in a way that no new v-structure is introduced to the model. Figure 15 illustrates the method for a true causal model shown in Figure 15 A. [46]

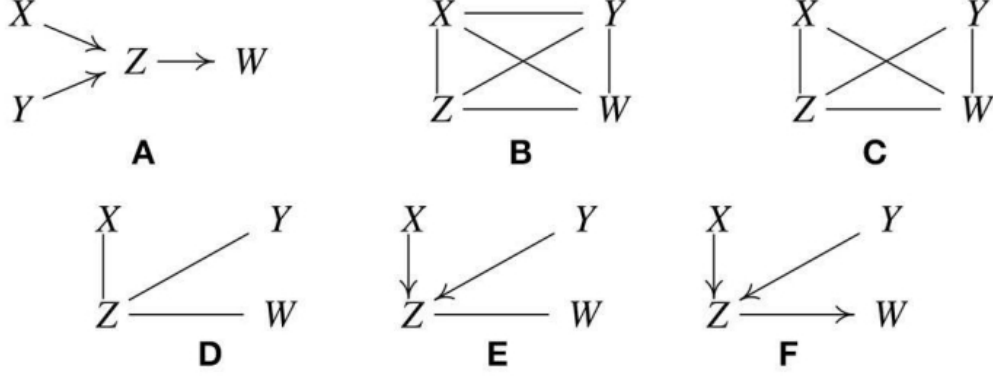


Figure 15: Illustration of how the PC algorithm works. (A) Original true causal graph. (B) PC starts with a fully-connected undirected graph. (C) The X-Y edge is removed because X and Y are independent. (D) The X-W and Y-W edges are removed because of conditional independence. (E) After finding v-structures. (F) After orientation propagation.[46]

Depending on the type of the data, various independence tests are used to infer the statistical relation between variables. Pearson’s linear correlation and Pearson’s Chi-square test are the two test can be used for continuous and discrete variable respectively.[48]

- **Pearson’s linear correlation**

Pearson correlation coefficient is a measure of the strength of the linear association between two variables. It is defined as the ratio of the covariance of two variables, by the product of their standard deviations and for the variables X and Y, it is formulated as follows.[49]

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \quad (22)$$

This measure returns a value between -1 and 1. The values close to -1, and 1 show the strongest linear relation between X and Y, and a value close to 0 means that there is no linear relation between them.

- **Pearson’s Chi-square test**

The chi-square test for independence is used to discover if there exists a relationship between two discrete variables.[50]

The chi-square test performs a statistical test in which the null hypothesis is that the variables are independent. The observations are compared to a model in which the data is distributed according to the expectation that the variables are independent. If the observed samples do not fit the model, the likelihood that the variables are dependent becomes stronger, thus proving the null hypothesis incorrect.[51]

This test is formally defined as:

$$\chi^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i} \quad (23)$$

O stands for the Observed frequency and E stands for the Expected frequency. It is the sum of normalized residuals of samples considering the expected values. Having calculated this value, the corresponding p-value is computed and depending on the significance level, the null hypothesis may or may not be rejected.

The Score-based algorithms

A score-based algorithm family, in contrast to constraint-based methods, begins with an empty graph, and then in a stepwise style, it adds the currently required edges and eliminates the unnecessary ones until it ends in an optimum structure. At each step, the decision of whether adding an edge is made on the basis of improving a fit measure based on some scores such as log-likelihood score.[48] When the score can not be improved any longer by adding the edges, the algorithm starts to enhance the score by removing the edges, if possible, and it continues until no further edges can be dropped.

5 Method

This chapter presents the workflow of the implementation of the methods described in the previous chapter.

5.1 Feature set

The first step of our implementation is to designate relevant features out of available numerous features that not all of them are capable of describing the latency.

As it was explained in the previous chapter, a causal model not only provides us with the information related to the mechanism by which the variables take on their values, but also it enables us to make predictions for the effect variable. Pearl’s causal hierarchy (section 4.2.2) suggests that given a model for intervention level, the inference at association level will be available. In other words, if X is the cause of Y , $X \rightarrow Y$, X must be able to predict the values that Y takes on provided the values of X . Therefor at the first level we need to select all the indicators (features) that are able to predict the target variable. Figure 16 illustrates the workflow of the implemented procedure to select the most relevant features from the various provided datasets.

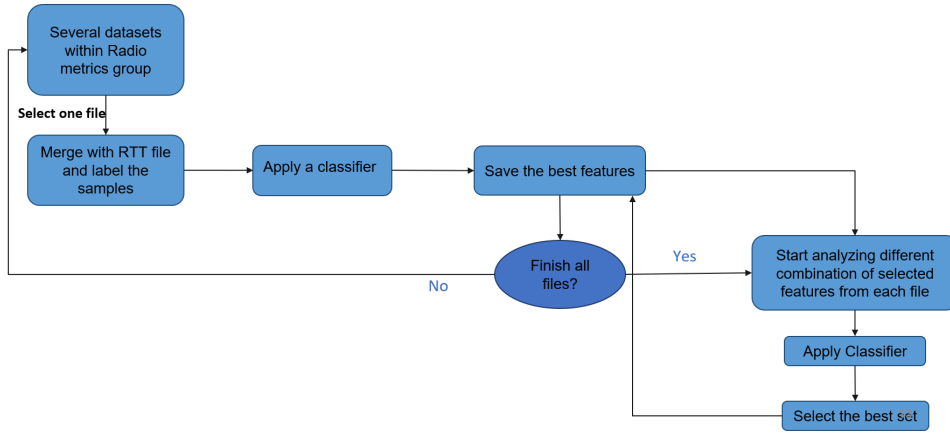


Figure 16: The workflow followed to select the feature set.

As explained earlier in chapter 3, several datasets containing different attributes are available which not all of them are of importance. At the EDA stage, the data was purified by dropping the zero-variance and keeping only one among the highly correlated attributes.

This step involves deeper investigating the variables with a view to selecting the most pertinent ones. It is basically a workflow that is performed recursively to refine the feature set. It starts at the file level by selecting each file separately. It could be possible to combine all of the datasets and then select the relevant features, however, file-level analysis was followed to prevent a considerable increase in the number of features, so applying the model and analyzing the result could be easier. Then the merging phase starts in order to assign an RTT value to each instance in which the selected file is concatenated with the RTT log file. After performing the merging step the labeling task must be executed. At this stage, a binary categorization is carried out based on a threshold provided by the Ericsson supervisor. The instances which have their RTT values below the threshold are labeled as 0 indicating the normal class and those having their RTT values above the threshold are assigned to the abnormal class labeled by 1. Eventually, the data is split to train and test and a binary classifier is trained to perform the classification. Having analyzed and visualized the results, the performances of the models, a decision was made on which indicators must be passed to the next analysis step. Any feature with importance higher than 0 was selected. In addition, the information provided by the Ericsson supervisor as a domain expert was also incorporated to set aside those features that are not considered KPIs.

Once all individual files have been completed, the process continues by combining the features stored in the previous step and analyzing them to arrive at a final relatively relevant feature set. Having been discussed and confirmed by the domain expert, a final selected set of the indicators was passed as the input for anomaly detection and causal inference analysis.

5.2 Anomaly detection by classification

The recorded data from six experiments were available for this study. Same radio traces were recorded during all the experiments but with different network configurations. In experiments 1 to 4, the positions of the UEs(endpoints) were constant whereas UEs were moving around while executing experiments 5 and 6. In addition, the network traffic load was kept constant in experiments 5 and 6 whereas a stepwise increase in the load was applied to other experiments. Figure 17 illustrates the distribution of the target variable after categorizing the RTT based on the defined threshold. The horizontal axis indicates the class labels where 0 and 1 represent the normal and abnormal categories respectively. The vertical axis is the counts of the instances within each class.

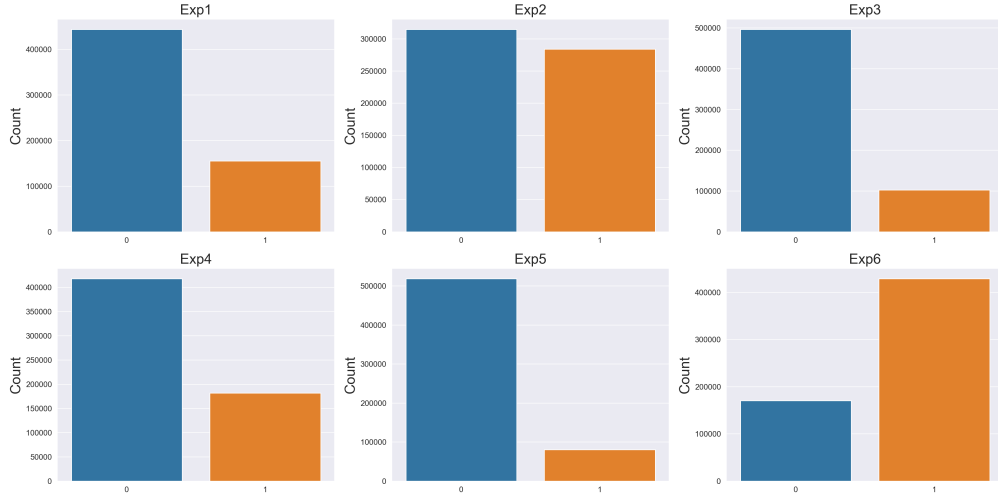


Figure 17: The distribution of the binary target variable. Experiment 2 seems to have the most balanced target variable.

As can be observed, the distribution of the labels is not identical in all experiments. Except in experiment 2, the distributions are all imbalanced so that there is a significant difference between the number of samples per two groups. Therefore, the data from experiment 2 was chosen to train the model and the other data sets were selected as the test sets to evaluate the performance of the model.

After standardizing the data so that the features have unit variance, a random forest classifier was trained on the training data. A dummy classifier was also trained as the baseline to compare its result with the random forest classifier results. A dummy classifier provide by Scikit-learn is a classifier that makes predictions using very simple rules.[38] The *Uniform* strategy was used since the training set is balanced. With this strategy the classifier generates predictions uniformly at random.[38] The hyperparameters were tuned through a grid search and cross-validation method. GridSearchCV module from Scikit-learn library[38] was used for selecting the optimum values for the hyperparameters. For each parameter, a grid of values was passed to the function. The train set is split into five folds and the function estimates the result for the estimator for the values provided in the search space. The combination of the values which could return the best score was finally returned as the optimized values for the hyperparameters. Having selected the best values for the hyperparameters, the classifier was trained on the data generated in experiment 2 and evaluated using the other datasets.

The distribution of the target variable is imbalanced in all of the test sets, therefore, we

need to evaluate our model in terms of precision (or recall) rather than just accuracy since the latter returns an overall score for the performance regardless of the distribution of the data. The former, however, specify the performance for each class so that we can evaluate the model considering the imbalance issue. From the previous chapter, we know that the F1 score is a harmonic mean of precision and recall, thus it might be an appropriate score to be focused on for evaluating the model.

All of the training and evaluating processes were implemented using the Scikit-learn library.[38]

5.3 Causal inference

In order to be able to perform a causal inference based on a Bayesian graphical network approach, a causal structure (a DAG) must be provided encoding the causal relationships between variables. The causal model can either be learned from the observational data or provided by a domain expert. In this study, the causal structure was tried to discover from the observations.

It is assumed that the observational data were generated by an unknown DAG. In practice identifying the true DAG is not possible and in many real problems, even the domain experts may not be able to build a true DAG for the problem. However, considering some initial assumptions, it is possible to determine an equivalence class of DAGs that all capture the same probability distribution. As explained in the previous chapter a CPDAG which is the equivalence class of DAGs is partially directed. It means that the edges whose orientations are different among the different DAGs will be considered as undirected. As a result, depending on how many undirected edges within a CPDAG, a number of possible DAGs can be considered, and the total causal effect of parents on their children can then be estimated.

The main idea of this method is taken from[17] in which the authors focused to solve a causal inference problem in biology.

The approach can be explained as follows. At the first step, it is assumed that there is a true causal model that generated the data, so the causal structure from observational data can be discovered. To that end, under the faithfulness and Markov assumptions, the PC algorithm was implemented to learn the causal structure from the data. The output would be a CPDAG which may contain m different DAGs. For each of those DAGs in the equivalence class, the causal effect of the indicators on the RTT value can be estimated. As a result, for the causal effect of variable X_i on the RTT, m values are available that may or may not be identical. Following this method, lower and upper bounds for the causal effect of X_i on RTT can be reported showing a range of values that could be the effect range of X_i .

For performing the causal inference the continuous variable for RTT was used. In addition to the Faithfulness and Markov assumptions, two more assumptions were also considered allowing to proceed with the explained approach. First, we assumed that there is no unobservable variable in our problem. Second, it was assumed that the variables are jointly Gaussian. Under these assumptions, the PC algorithm was used to discover the causal structure.

In order to estimate the causal effect, the assumption of Gaussianity plays the main role to choose the estimation method. As aforementioned, it is assumed that our observational data comes from a multivariate Gaussian distribution that is faithful to the underlying causal DAG without hidden variables. Under such assumptions, the *Pcalg* package [52] was used for both causal discovery and the total causal effect estimation.

The Gaussianity assumption is the most crucial assumption, as most of the features do not show a perfect normal distribution when visualizing them.(Figure 18) Having discussed the issue with the domain expert, it was decided to proceed on under such assumption as the marginal distributions can be closer to normal when recording the data over a longer period of the system life. Without considering and imposing this assumption, the implementation of the described methods is not possible and the problem

becomes very complicated. Although the current data hardly fulfill the assumption, the results of causal inference can provide an overview of cause-effect relationships among variables.

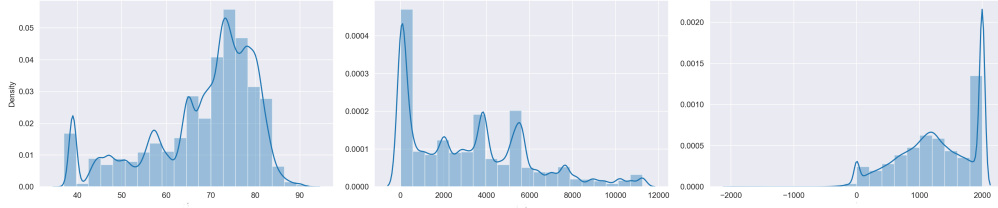


Figure 18: Histograms showing the typical distribution of the features.

The implemented function of the package for estimating causal effect applies *back-doorcriteria* to estimate the total causal effect of a parent on its child. A simple linear regression adjusting is used as the method for estimation. Referring to the package documentation “for example suppose that we are interested in the causal effect of X on Y . if Y is not a parent of X , we can take the regression coefficient of X in the regression $lm(Y \sim X + pa(X, G))$, where $pa(X, G)$ denotes the parents of X in the DAG G ; if Y is a parent of X in G , we can set the estimated causal effect to zero.”[52] This is because of the fact that the causality is an asymmetric relation, as it was explained in previous chapter. In other word, if we have $X \rightarrow Y$, from the causality point of view only the child (here Y) can be affected by the parent(s) (here X).

6 Results

The results of the thesis will be reported in this chapter.

6.1 Anomaly detection by classification

Figure 19 shows the classification performance reports for the train and test sets.

Performance on the train data					Performance on the test data 1				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.99	0.95	0.97	315022	0	0.88	0.80	0.84	444014
1	0.95	0.99	0.97	284338	1	0.55	0.69	0.61	155444
accuracy			0.97	599360	accuracy			0.77	599458
macro avg	0.97	0.97	0.97	599360	macro avg	0.71	0.74	0.72	599458
weighted avg	0.97	0.97	0.97	599360	weighted avg	0.79	0.77	0.78	599458

Performance on the test data 3					Performance on the test data 4				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.91	0.82	0.86	497094	0	0.90	0.85	0.87	418113
1	0.42	0.61	0.50	102810	1	0.69	0.78	0.73	181783
accuracy			0.79	599904	accuracy			0.83	599896
macro avg	0.66	0.72	0.68	599904	macro avg	0.79	0.81	0.80	599896
weighted avg	0.83	0.79	0.80	599904	weighted avg	0.83	0.83	0.83	599896

Performance on the test data 5					Performance on the test data 6				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.87	0.93	0.90	519277	0	0.44	0.50	0.47	170524
1	0.20	0.10	0.13	80689	1	0.79	0.75	0.77	429375
accuracy			0.82	599966	accuracy			0.68	599899
macro avg	0.53	0.52	0.52	599966	macro avg	0.61	0.62	0.62	599899
weighted avg	0.78	0.82	0.80	599966	weighted avg	0.69	0.68	0.68	599899

Figure 19: The classification performance reports for the training and test sets.

The barplots in Figure 20 illustrates the results visually.

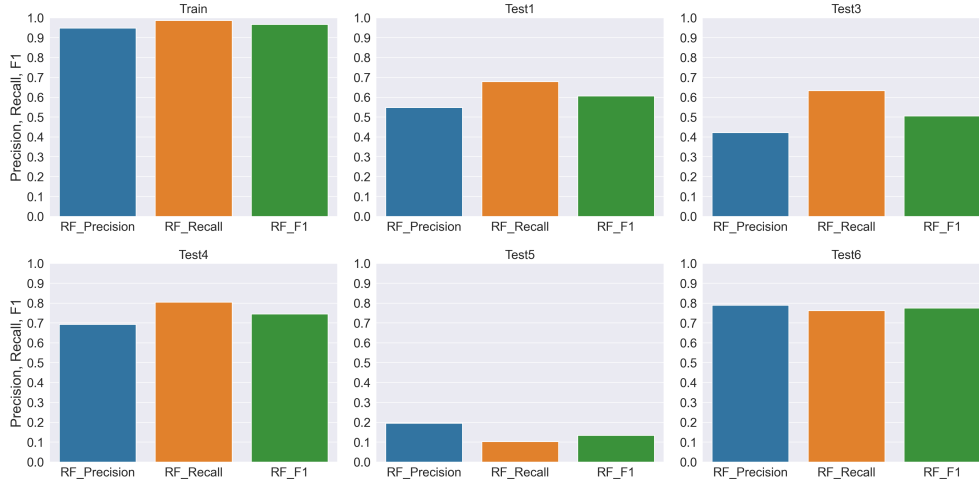


Figure 20: The classification performance for the training and test sets.

Considering the objective of the study which is detecting the anomalies, the model seems to perform fairly acceptable on experiments 6 and 4, as the F1 scores for the anomaly class are 0.77 and 0.73 respectively. This assessment is only based on considering the capability of the model to classify the anomalies into the true category, however, if the classification of the normal instances into the true class has also the same level of importance, the judgment might be different. While the scores show a good result for such consideration in experiment 4, the performance in experiment 6 may not be

considered good. From the anomaly detection perspective the worst scores are related to the test set 5. One reason for such a result could be the extent that the distribution of the class label is imbalanced in this experiment. From figure 17 it can be seen that experiments 5 and 3 are the most imbalanced data sets among the test sets and the performance for these two sets are the worst.

Figure 21 visually compares the model performances and the baseline classifier. In this figure the red and black lines show the trained model precision and recall respectively. The dummy classifier performance scores can be followed along the blue color. As it can be observed the trained model outperformed the dummy classifier for all of the test sets except the test set 5.

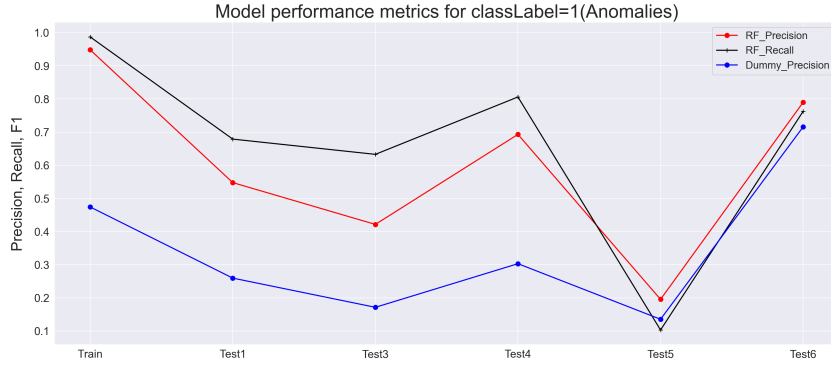


Figure 21: Comparing the trained model performance and the dummy classifier performance.

Figure 22 shows the overall performance of the model aggregating all the test sets into a unified test set. As the barplot in this figure depicts, considering all test sets together the F1 score for our model performance was just below 70%.

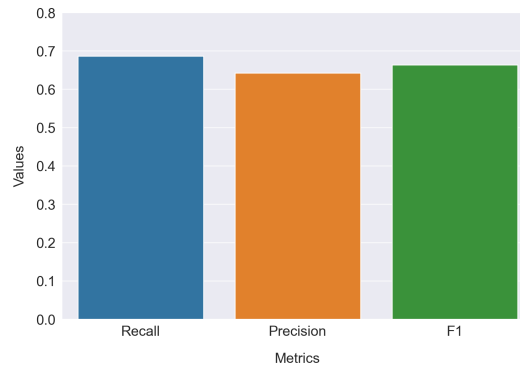


Figure 22: The overall performance scores aggregating all test sets.

And finally Figure 23 illustrates the importance of features in predicting the class labels of the RTT. All the features more or less have been able to contribute in classification. *bs* seems to be the most important variable in terms of predicting the class labels, while the *ul* shows the less contribution in the classification.

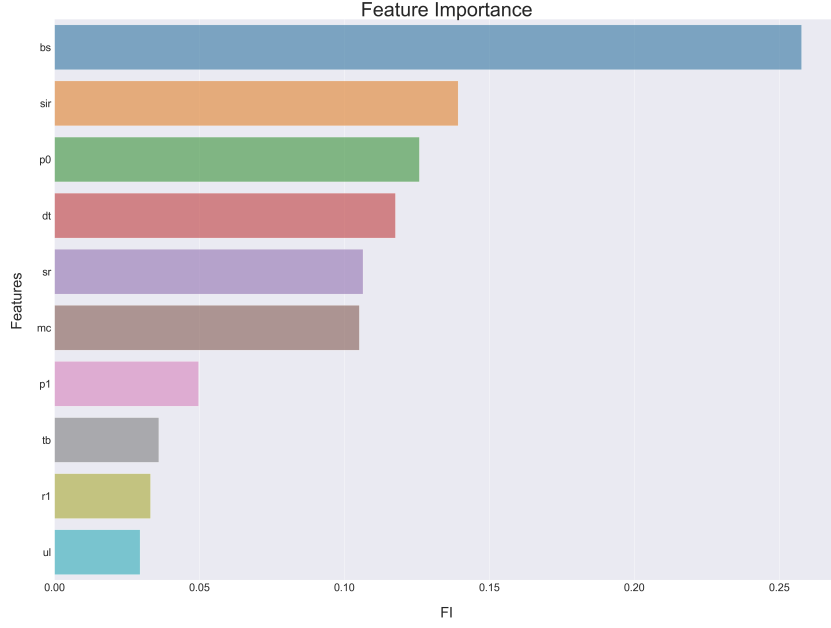


Figure 23: Feature importance resulting from the random forest classification.

6.2 Causal inference

The figure 24 illustrates the resulting model from the PC algorithm. For visualizing the graph [53] was used.

As it can be seen *sr*, *bs*, *mc*, *p1*, and *r1* are the parents of RTT in this structure. Moreover, it is observable that there are three ambiguous edges whose orientations were not determined by the algorithm. They are shown by bidirectional arrows. Therefore there could be many different DAGs belonging to this equivalence class, however not all of them might be a valid graph as some of the combinations might produce a cycle.

Under the mentioned assumptions and implementing the *Pcalg* package function the total causal effects of the parents of RTT were estimated which are presented in table 25.

As the table presents, the causal effect values are identical for *mc*, and *p1*, however, they are different for the other indicators. These values are actually the coefficients in the aforementioned linear regression. For instance, the value 0.240251 for *bs* says that if the other factors are held constant a unit increase in the *bs* value will increase the RTT value by about 24 percent.

Figure 26 illustrates the total causal effect bounds for each indicator. As the figures depict the largest effect stems from the *sr* feature and the minimum effect caused by the *r1* as its causal effect is close to zero. From the range point of view, *bs* seems to takes its value within a wider range compared to other ones. According to this analysis, *bs*, *sr*, *mc*, *p1*, and *r1* are the parents of RTT in the graph. It means that if these features are subjected to some intervention, they will affect the RTT values.

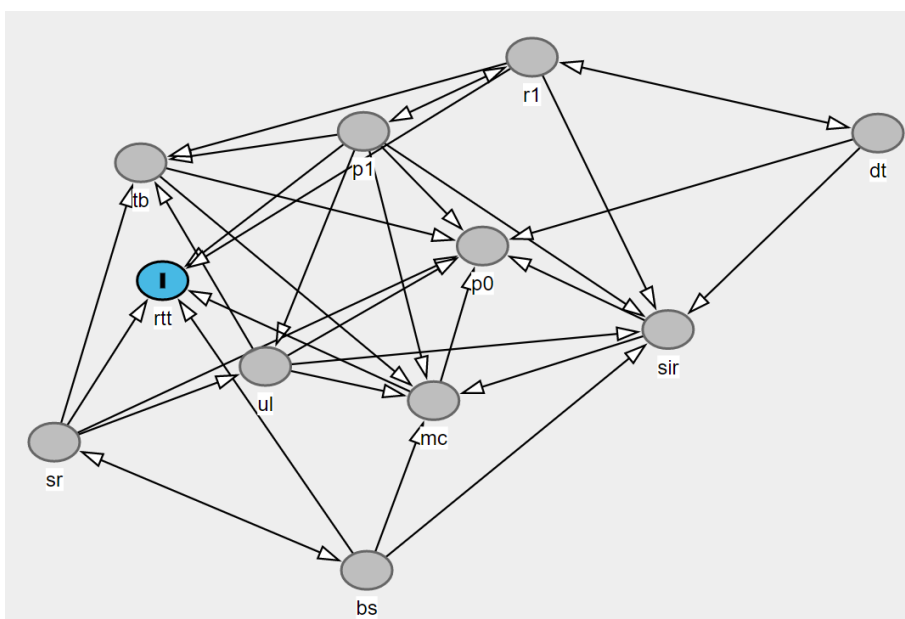


Figure 24: The causal structure discovered using the PC algorithm. Drawn in [53]

	bs	sr	mc	p1	r1
DAG1	0.240251	0.433565	0.281697	-0.351366	-0.025759
DAG2	0.240251	0.433565	0.281697	-0.351366	-0.048125
DAG3	0.240251	0.433565	0.281697	-0.351366	-0.025761
DAG4	0.335114	0.372359	0.281697	-0.351366	-0.025759
DAG5	0.335114	0.372359	0.281697	-0.351366	-0.048125
DAG6	0.335114	0.372359	0.281697	-0.351366	-0.025761

Figure 25: Estimated total causal effects.

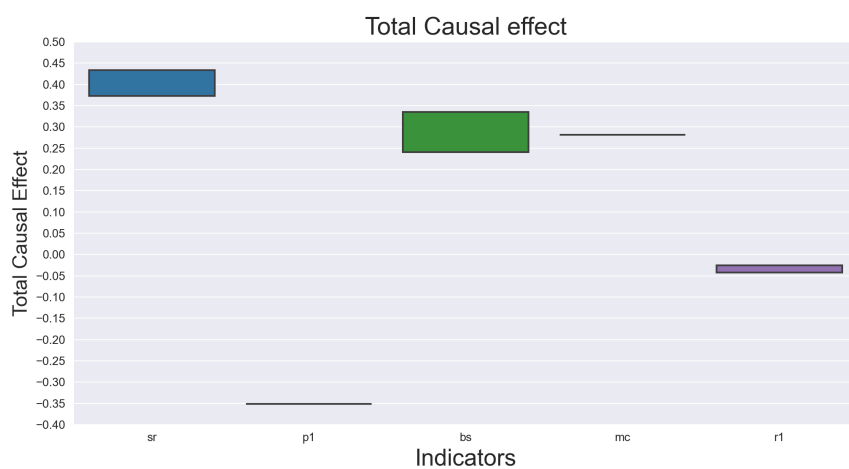


Figure 26: Total causal effects bounds.

7 Discussion

The goal of this study is to detect the anomalies in terms of high latency problems in 5G mobile network communication and analyzing the key performance indicators to figure out the main causal factors affecting the variations in the transmission time of sending a packet through the network. Two approaches were performed. First, we trained a classifier, specifically a Random forest classifier, to predict the anomalies. Then by implementing the causal Bayesian network we tried to infer the main causal factors. The data based on which the training and evaluating executed were recorded over a period of about 12 minutes. The granularity of recording was in microseconds and nanoseconds. The first challenge was how to merge the different datasets to be able to make use of all of the data as well as to be able to concatenate the RTT log file to the radio traces files. Because of the short period of the recording and to be able to match the samples based on time, we changed the resolution to millisecond level. As a result of this resampling, some duplicates and some NaN values were generated, and so we had to deal with that problem by implementing the mean/mode functions to aggregate the duplicates and also impute the missing values which were explained in chapter 3. This procedure might affect the result as it could produce might be considered artificial anomalies. However, this is an indispensable part of solving the problem in real words. For such analysis, the ideal dataset should be as long as possible not only to study a longer life of system behavior but also to avoid introducing some new samples to the data through either the aggregation or imputation. “the larger observation period, the better statistical properties and therefore better predictive and diagnostic capabilities will be achieved.”[8]

For the prediction task, we chose to proceed on with a tree-based ensemble method; a Random forest classifier. In [8] Random forest was used as a method to diagnose the main causes of the events in a network. The idea in that study was to make use of the capability of the Random forest to assign a measure of importance to each feature in order to analyze the root causes of different events that happened in the network. Our purpose of implementing the Random forest, however, was only to predict the anomalies and also to select the most pertinent variables out of numerous available attributes during the preprocessing and preparation. The motivation to implement a tree-based method was that we almost could relax any vital assumptions for our analysis. The random forest algorithm does not assume any particular assumption for the data distribution.[8] Each base estimator(decision tree) recursively splits the data into different subsets in such a way that the maximum purity is achieved for each region and finally it returns an average of the whole trees’ results as the final outcome. In addition, with the help of feature importance, we could make a comparison between prediction and causal inference results which will be explained.

Despite the positive point of relaxing any particular assumption in the prediction stage, we need to impose some particular assumptions for the causal inference. In implementing the Bayesian network approach to infer the causality, faithfulness and Markov assumptions are of crucial importance to be taken into account. These assumptions will allow us to assume that there is an unknown underlying causal graph that generated the data and the data is faithful to that graph. In other words, any statistical independence relationship inferred from the data can be captured by the independence relationship via the graph and vice versa. These assumptions are the most important key to be able to implement the approach in general. In addition to them, we imposed other assumptions which were vital for applying the algorithms were chosen to proceed on with. We assumed that there is no unobservable variable within our data. Moreover, we assumed that the data has a multivariate Gaussian distribution. In fact, the choice of the algorithms made us impose these assumptions and the choice of the algorithm was to simplify the problem. Some more advanced algorithms have been introduced and can be found through in [46],[54],[55],[56], and [57]. In these papers, the authors tried to relax some of the mentioned assumptions in causal discovery. The complexity and time limitation prevented us to dive into these methods, thus for the sake of simplicity we imposed the aforementioned assumptions.

Although all the features played a role in classification, however only 5 out of them

turned out to be the parents (causes) in our discovered causal structure. This could be a point to distinguish between prediction and causality. The feature importance reveals the ability of the features to contribute in the prediction and so they were ranked accordingly. However, this ability does not necessarily disclose the causality of the problem. The causality analysis results validation strongly depends on how close the predefined assumptions are to the reality. Under the aforementioned assumptions, we performed the causal inference and the results showed that 5 indicators may be considered interesting from the interventional point of view. It means if one manipulates the values that they take on, we can see a change happening in the value of RTT. In this comparison, another point also must be taken into account. In classification the binary variable of categorized RTT was used as the target variable, whereas the continuous RTT was considered as the outcome in causal inference analysis. Such a decision was taken in order for being able to presume the required assumptions for applying the causal discovery and causal effect estimation methods discussed earlier.

8 Conclusion

The aim of this thesis was to detect the high latency problem in a 5G mobile network. Specifically, it was interested to investigate whether the key performance indicators are able to predict the high latency instances as the anomaly in the network. In addition, the causal effects of the indicators were also an objective of the study. In the causality analysis, it was interested to know if any of the indicators is a cause affecting the RTT values. For the anomaly detection part, a binary classification was performed in which a random forest classifier was built and trained. Bayesian networks were implemented to study the causality part of the thesis.

Are the key performance indicators (KPIs) of a 5G network able to predict the high latency values in a telecommunication network? Random forest classifier was evaluated on different test sets. For each of the test sets a different performance resulted. Considering the anomaly detection An overall precision of about 70% was achieved, which means 70 percent of the assigned samples into the abnormal class have been classified correctly.

Which of those KPIs can be considered as the main causes that affect the latency?

The discovered causal structure indicates that 5 indicators, *bs*, *sr*, *mc*, *p1*, and *r1*, can be considered as the parents of RTT in the causal graph. While the estimated total causal effect for *mc* and *p1* seems to be a constant value, it varies within a range for *bs*, *sr*, and *r1*.

References

- [1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009, doi: 10.1145/1541880.1541882.
- [2] J. Rooney and L. N. Hauvel, “Root Cause Analysis For Beginners,” *Quality Progress*, vol. 37, pp. 45–53, Jul. 2004.
- [3] “ETSI TS 122 261 V15.6.0 (2018-10) - 5G; Service requirements for next generation new services and markets (3GPP TS 22.261 version 15.6.0 Release 15).” <https://standards.iteh.ai/catalog/standards/etsi/37892c23-61ef-4bb8-bfb2-09c473af0bb6/etsi-ts-122-261-v15-6-0-2018-10> (accessed Mar. 14, 2021).
- [4] M. Solé, V. Muntés-Mulero, A. I. Rana, and G. Estrada, “Survey on Models and Techniques for Root-Cause Analysis,” Jul. 03, 2017. <http://arxiv.org/abs/1701.08546> (accessed Mar. 14, 2021).
- [5] A. Gomez-Andrades, R. Barco, I. Serrano, P. Delgado, P. Caro-Oliver, and P. Munoz, “Automatic root cause analysis based on traces for LTE self-organizing networks,” *IEEE Wireless Communications*, vol. 23, no. 3, pp. 20–28, Jun. 2016, doi: 10.1109/MWC.2016.7498071.
- [6] J. Ali-Tolppa, S. Kocsis, B. Schultz, L. Bodrog, and M. Kajo, “SELF-HEALING AND RESILIENCE IN FUTURE 5G COGNITIVE AUTONOMOUS NETWORKS,” in *2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K)*, Nov. 2018, pp. 1–8, doi: 10.23919/ITU-WT.2018.8598115.
- [7] G. Zhu, J. Zan, Y. Yang, and X. Qi, “A Supervised Learning Based QoS Assurance Architecture for 5G Networks,” *IEEE Access*, vol. 7, pp. 43598–43606, 2019, doi: 10.1109/ACCESS.2019.2907142.
- [8] J. M. N. Gonzalez, J. A. Jimenez, J. C. D. Lopez, and H. A. P. G, “Root Cause Analysis of Network Failures Using Machine Learning and Summarization Techniques,” *IEEE Communications Magazine*, vol. 55, no. 9, pp. 126–131, Sep. 2017, doi: 10.1109/MCOM.2017.1700066.
- [9] “Bayesian network,” *Wikipedia*. Mar. 09, 2021, Accessed: Mar. 14, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Bayesian_network&oldid=1011105360.
- [10] T. J. T. Koski and J. Noble, “A Review of Bayesian Networks and Structure Learning,” *Mathematica Applicanda*, vol. 40, no. 1, 2012, doi: 10.14708/ma.v40i1.278.
- [11] A. Chigurupati and N. Lassar, “Root cause analysis using artificial intelligence,” in *2017 Annual Reliability and Maintainability Symposium (RAMS)*, Jan. 2017, pp. 1–5, doi: 10.1109/RAM.2017.7889651.
- [12] R. Khanafer, L. Moltzen, H. Dubreil, Z. Altman, and R. Barco, “A Bayesian Approach for Automated Troubleshooting for UMTS Networks,” in *2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, Sep. 2006, pp. 1–5, doi: 10.1109/PIMRC.2006.254370.
- [13] R. Barco, L. Nielsen, R. Guerrero, G. Hylander, and S. Patel, “Automated troubleshooting of a mobile communication network using Bayesian networks,” in *4th International Workshop on Mobile and Wireless Communications Network*, Sep. 2002, pp. 606–610, doi: 10.1109/MWCN.2002.1045837.
- [14] R. M. Khanafer *et al.*, “Automated Diagnosis for UMTS Networks Using Bayesian Network Approach,” *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2451–2461, Jul. 2008, doi: 10.1109/TVT.2007.912610.
- [15] H. Mfula and J. K. Nurminen, “Adaptive Root Cause Analysis for Self-Healing in 5G Networks,” in *2017 International Conference on High Performance Computing Simulation (HPCS)*, Jul. 2017, pp. 136–143, doi: 10.1109/HPCS.2017.31.

- [16] L. Bennacer, Y. Y. Amirat, A. A. Chibani, A. Mellouk, and L. Ciavaglia, “Self-Diagnosis technique for Virtual Private Networks combining Bayesian Networks and Case-Based Reasoning,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 354–366, 2015, Accessed: Mar. 14, 2021. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01538572>.
- [17] M. H. Maathuis, M. Kalisch, and P. Bühlmann, “Estimating high-dimensional intervention effects from observational data,” *The Annals of Statistics*, vol. 37, no. 6A, pp. 3133–3164, Dec. 2009, doi: 10.1214/09-AOS685.
- [18] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman, *Causation, Prediction, and Search*. MIT Press, 2000.
- [19] “Telecommunications network,” *Wikipedia*. Mar. 16, 2021, Accessed: Mar. 19, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Telecommunication_s_network&oldid=1012496503.
- [20] “Public switched telephone network,” *Wikipedia*. Mar. 16, 2021, Accessed: Mar. 19, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Public_switched_telephone_network&oldid=1012534419.
- [21] “Node (networking),” *Wikipedia*. Jan. 05, 2021, Accessed: Mar. 19, 2021. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Node_\(networking\)&oldid=998410419](https://en.wikipedia.org/w/index.php?title=Node_(networking)&oldid=998410419).
- [22] J. Sachs *et al.*, “Adaptive 5G Low-Latency Communication for Tactile Internet Services,” *Proceedings of the IEEE*, vol. 107, no. 2, pp. 325–349, Feb. 2019, doi: 10.1109/JPROC.2018.2864587.
- [23] “What is QoS (Quality of Service)?” <https://searchunifiedcommunications.techtarget.com/definition/QoS-Quality-of-Service> (accessed Mar. 19, 2021).
- [24] “Packet loss,” *Wikipedia*. Jan. 29, 2021, Accessed: Mar. 19, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Packet_loss&oldid=1003607742.
- [25] “Jitter,” *Wikipedia*. Mar. 14, 2021, Accessed: Mar. 19, 2021. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Jitter&oldid=1012105349>.
- [26] “What is Jitter? | Nextiva Support.” <https://www.nextiva.com/support/articles/what-is-jitter.html> (accessed Mar. 19, 2021).
- [27] “Bandwidth (computing),” *Wikipedia*. Jan. 27, 2021, Accessed: Mar. 19, 2021. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Bandwidth_\(computing\)&oldid=1003172265](https://en.wikipedia.org/w/index.php?title=Bandwidth_(computing)&oldid=1003172265).
- [28] “Bandwidth Definition.” <https://techterms.com/definition/bandwidth> (accessed Mar. 19, 2021).
- [29] O. Al-Saadeh, G. Wikstrom, J. Sachs, I. Thibault, and D. Lister, “End-to-End Latency and Reliability Performance of 5G in London,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–6, doi: 10.1109/GLOCOM.2018.8647379.
- [30] M. Bennis, M. Debbah, and H. V. Poor, “Ultrareliable and Low-Latency Wireless Communication: Tail, Risk, and Scale,” *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018, doi: 10.1109/JPROC.2018.2867029.
- [31] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, “A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3098–3130, doi: 10.1109/COMST.2018.2841349.
- [32] “Round-trip delay,” *Wikipedia*. Jan. 24, 2021, Accessed: Mar. 28, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Round-trip_delay&oldid=1002445310.
- [33] “RSRP,” *Wikipedia*. Jul. 21, 2020, Accessed: Mar. 28, 2021. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=RSRP&oldid=968798646>.

- [34] “Pandas - Python Data Analysis Library.” <https://pandas.pydata.org/> (accessed Apr. 02, 2021).
- [35] “Matplotlib: Python plotting — Matplotlib 3.4.1 documentation.” <https://matplotlib.org/> (accessed Apr. 02, 2021).
- [36] “Seaborn: Statistical data visualization — seaborn 0.11.1 documentation.” <https://seaborn.pydata.org/> (accessed Apr. 02, 2021).
- [37] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 103. New York, NY: Springer New York, 2013.
- [38] “Scikit-learn: Machine learning in Python — scikit-learn 0.24.1 documentation.” <https://scikit-learn.org/stable/> (accessed Apr. 02, 2021).
- [39] G. Louppe, “Understanding Random Forests: From Theory to Practice,” Jun. 03, 2015. <http://arxiv.org/abs/1407.7502> (accessed May 21, 2021).
- [40] F. Dablander, “An Introduction to Causal Inference,” Feb. 2020, doi: 10.31234/osf.io/b3fkw.
- [41] P. Spirtes, “Introduction to Causal Inference,” p. 20.
- [42] S. F. Savitt, “The Direction of Time,” *The British Journal for the Philosophy of Science*, vol. 47, no. 3, pp. 347–370, 1996, [Online]. Available: <http://www.jstor.org/stable/687983>.
- [43] J. PEARL, “Causal diagrams for empirical research,” *Biometrika*, vol. 82, no. 4, pp. 669–688, Dec. 1995, doi: 10.1093/biomet/82.4.669.
- [44] J. Pearl, “The seven tools of causal inference, with reflections on machine learning,” *Commun. ACM*, vol. 62, no. 3, pp. 54–60, Feb. 2019, doi: 10.1145/3241036.
- [45] “Bayesian network,” *Wikipedia*. Mar. 19, 2021, Accessed: Mar. 28, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Bayesian_network&oldid=1012991471.
- [46] C. Glymour, K. Zhang, and P. Spirtes, “Review of Causal Discovery Methods Based on Graphical Models,” *Front. Genet.*, vol. 10, 2019, doi: 10.3389/fgene.2019.00524.
- [47] B. Chen and J. Pearl, “Graphical Tools for Linear Structural Equation Modeling,” Defense Technical Information Center, Fort Belvoir, VA, Jun. 2014. doi: 10.21236/ADA609131.
- [48] “Bnlearn - Bayesian network structure learning.” <https://www.bnlearn.com/> (accessed Apr. 02, 2021).
- [49] “Pearson correlation coefficient,” *Wikipedia*. Mar. 15, 2021, Accessed: Apr. 02, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Pearson_correlation_coefficient&oldid=1012194631.
- [50] “SPSS Statistics Tutorials and Statistical Guides | Laerd Statistics.” <https://statistics.laerd.com/> (accessed Apr. 02, 2021).
- [51] <https://www.ling.upenn.edu/~clight/> (accessed Apr. 02, 2021).
- [52] M. Kalisch, H. Colombo, and B. Maathuis, “More Causal Inference with Graphical Models in R Package pcalg,” p. 37.
- [53] “DAGitty v3.0.” <http://www.dagitty.net/dags.html> (accessed Apr. 14, 2021).
- [54] F. Eberhardt, “Introduction to the foundations of causal discovery,” *Int J Data Sci Anal*, vol. 3, no. 2, pp. 81–91, Mar. 2017, doi: 10.1007/s41060-016-0038-6.
- [55] E. S. Adabor, G. K. Acquah-Mensah, and F. T. Oduro, “SAGA: A hybrid search algorithm for Bayesian Network structure learning of transcriptional regulatory networks,” *Journal of Biomedical Informatics*, vol. 53, pp. 27–35, Feb. 2015, doi: 10.1016/j.jbi.2014.08.010.

- [56] P. Hoyer, D. Janzing, J. Mooij, J. Peters, and B. Schölkopf, *Nonlinear causal discovery with additive noise models*. 2008, p. 696.
- [57] S. Shimizu, P. O. Hoyer, A. Hyvarinen, and A. Kerminen, “A Linear Non-Gaussian Acyclic Model for Causal Discovery,” p. 28.