M.Sc Thesis

Author: Andrea Cappello

Proving the Business Case of Virtual NOC

System design for the analysis of network alarms, based on time and impact.

Supervisor from Elisa Automate: **Jaiju Joseph**

Academic Supervisor: **Prométhée Spathis**

Aalto University - Sorbonne University
2019/21

Begin.

Abstract

The Telco community has foreseen fully autonomous networks able to perform tasks, handle problems, and be capable of re-configuring themselves without requiring human intervention. To achieve this vision, Elisa Automate is developing an automation system for RAN fault management. In the context of the product they are developing, I propose (i) a system to automate the estimation of the time required by humans to handle alarms to accelerate the proof of concept with new customers, and (ii) an approach that leverages machine learning to automatically extract the counters from the Performance Management data to estimate the impact of alarms on network performance and consequently on the end-users. The value creation chain of fault management automation with its managerial implications concludes the research question.

The system (i) has shown promising results and little more work, following the suggestions provided, will allow having a more mature functionality ready to be tested in real scenarios. On the other hand, system (ii) has had a more challenging development due to several problems, mainly the lack of data for the experiments. Furthermore, the final approach has not yet been identified and will require further research to find a workable conclusion.

Together, systems (i) and (ii) will make it possible to monitor the direct and indirect effects of automating fault management, the manual labor hours saved, and the network performance increase, respectively.

Acknowledgments

We live in an era where progress is given for granted and it seems the natural path for our species.

I am aware, however, that progress is the result of hard-working people that dedicate their lives to study and advance one little wheel at a time in the complicated machine that is our society.

I want to thank the people I interacted with in the past two years. I have grown and learned mostly thanks to the people I met during my studies at the EIT Digital Master School, while supported by my family and my beloved ones. The experiences and the friendships I accumulated are and will be precious.

The Automate team has been the most recent group of people to trust me and I hope my contribution will be valuable to the development of their mission. My mentors have been there in the moments of need and they are aware of the great work they did to help me succeed.

Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Meaning
RAN	Radio Access Network.
B2B	Business to Business.
NOC	Network Operation Center.
CAPEX	Capital Expenditure.
OPEX	Operating Expense.
CSP	Communications Service Providers.
DSP	Digital Service Providers.
QoS	Quality of Service.
SLA	Service Level Agreement.
UI	User Interface.
AE	Automation Engine.
PM	Performance Management.
AI	Artificial Intelligence.
ML	Machine Learning.
KPI	Key Performance Indicator.
FTE	Full Time Equivalent.
PoC	Proof of Concept.
UML	Unified Modeling Language.
UE	User Equipment.
UPS	Uninterruptible Power Supply.
LTE	Long Term Evolution.
VSWR	Voltage Standing Wave Ratio.
SQL	Structured Query Language.
XML	eXtensible Markup Language.
AUC	Area Under the Curve.
ROC	Receiver Operating Characteristic.
OTT	Over-The-Top.
EBITDA	Earnings Before Interest, Taxes, Depreciation and Amortization.
OSS	Operation Support System.

1 Introduction

According to recent reports published by industry leaders, including Gartner [1], Deloitte [2], and Ericsson [3], the Telco industry is undergoing a wave of radical innovation, including network virtualization and network openness. The growth we are witnessing in recent years is made possible by technology advancements and unpredictable events such as the COVID-19 pandemic, which dominated 2020 [4]. These factors are leading to increasingly complex systems. To cope with this turn, an update in the way we handle these systems is needed. In particular, the current manual fault management practice will eventually become insufficient, and automation will be necessary to keep pace with the requirements of new technologies, including 5G and beyond [2]. This work focuses on telco systems, such as the Network Operations Center, that are undergoing digital transformation. Network operations refer to the activities that service providers rely on to monitor, manage, and respond to alerts on their network's availability and performance [5].

1.1 Elisa Automate

Elisa Oyj is one of the biggest telecom operators in Finland [6] and Elisa Automate is a business division of Polystar, a subsidiary of Elisa Oyj. It concentrates on developing and commercializing telco automation innovations that were pioneered by Elisa in its network. It has developed AI automation for mobile networks and implemented it in its networks for ten years and counting [7]. The solutions proposed by Elisa Automate offer a set of field-proven use cases for planning, operations, and optimization. To meet operators' challenges related to increasing network complexity, one of the viable approaches is to automate the network management and fault resolution processes. Automation of RAN processes through a Virtual NOC (Network Operation Center) ensures considerable and measurable benefits. With this automated operations solution, CSPs (Communications Service Providers) can automate 100% of the alarms, create and update tickets and solve them automatically 24/7. They reduce OPEX and improve resolving time. Operators can ensure better quality at a lower cost [7]. Customers are other telecom operators outside Finland. Currently, the team is working on two products: Virtual NOC (Network Operations Center) and Automation Engine (AE).

2 Background I - Business Concepts

This section briefly describes the several terms used throughout this work and also explains the relationships between them.

2.1 CAPEX

The term Capital Expenditure (CAPEX) refers to investments to enhance, upgrade or change an asset [8], such as infrastructure or technological capability, to improve their performance, and subsequently the revenue. E.g. Telecom operators investing in next generation technologies to improve their services and thereby, increase their revenue.

2.2 OPEX

The term Operating Expense (OPEX) refers to the funds that an enterprise needs to use to maintain its operational capability, required for the day-to-day functioning of the business [8]. E.g. the expenses needed to keep the network up and running.

2.2.1 Relation between CAPEX and OPEX

In most cases, OPEX can be optimized for the long run with initial CAPEX investment. Especially when we talk about automation we implicitly talk about CAPEX increase [9]. Automation systems save OPEX in the long run but to be implemented, they initially require CAPEX. Additionally, CAPEX and OPEX are treated differently for tax accounting purposes [9].

EBITDA (Earnings Before Interest, Taxes, Depreciation, and Amortization) has been heavily used as a parameter to shape the way an operator decides to invest to offer services to its customers [9]. Since opex directly eats the EBITDA, opex is kept low while CAPEX is increased. Therefore, operators prefer investments that decrease opex in the long term.

2.3 Technical Interface

The technical interface means the integration built between different systems to allow data flow between them. As an example, an electronic connection

is built between payroll software and the incomes register, allowing data to be transferred between the two applications [10].

2.4 Business Interface

A business interface represents a point of access where a business service is made available to the environment. A business interface exposes the functionality of a business service to other business roles or actors [11]. It is often referred to as a channel (Telephone, Internet, Ticketing System, etc.).

2.5 Churn Rate

In this work's context, churn rate refers to the annual percentage of subscribers that leave their service provider to move to another operator. I.e., the fraction of people that stop their monthly subscription with *Operator X* and start a similar subscription with another operator, may it be *Operator Y*, *Operator Z* or *Operator F*.
This decision may be due to different reasons, including discontent at the quality of service received or better subscription offers.

2.6 Retention Rate

Retention rate is the opposite of churn rate (see 2.5), it measures how many customers decide to renew their contract with the same operator once their previous contract ends. Retention Rate is an indicator of how good the *product market fit* is [12].

2.7 Key Performance Indicator (KPI)

A Key Performance Indicator is a measurable value that demonstrates how effectively a company is achieving key business objectives. Organizations use KPIs at multiple levels to evaluate their success at reaching targets, as they are abstractions of business outcomes and they help to take actions. High-level KPIs may focus on the overall performance of the business, while low-level KPIs may focus on processes in departments such as sales, marketing, HR, support and others [13]. In the project scope, it is seen as a way of measuring the effectiveness of an organization and its progress towards achieving its goals.

2.7.1 The SMART approach

The SMART approach is a well-known criterion to set objectives [14]. SMARTER stands for Specific, Measurable, Attainable, Relevant, and Time-bound. It is worth keeping in mind that if it is difficult to measure a KPI, the objective is not yet sufficiently defined. If it can measured, it is possible to take action to influence it.

3 Background II - Towards Autonomic Networks

The concept of autonomic networks follows the initiative started by IBM in 2001 in autonomic computing. The aim is to create autonomous, i.e. self-managing networks to overcome the growing complexity of the internet and enable future growth [15].

According to the Telco community [2], the future holds fully autonomous networks able to perform tasks to handle problems and capable of re-configuring themselves without requiring human intervention. Such vision is dictated by the increasing density and complexity of modern networks. This section explains the meaning of Zero-touch network automation.

3.1 From CSPs to DSPs

Communication Service Providers (CSPs) are undergoing a transition to becoming Digital Service Providers (DSPs). The difference is about the nature of the communication services provided, with the great majority related to internet-based services. They need to transform their network into a platform for on-demand services that are fast to deploy and manage, flexible in terms of QoS (Quality of Service) and SLAs (Service Level Agreements) with which they need to bear with the three main pillars of 5G: eMBB, mMTC, and URLLC [2].

3.2 Zero-touch network automation

In such an intricate ecosystem, automation, and eventually zero-touch automation, is the only feasible way to keep the OPEX (Operating Expense) constant whilst handling the complexity and saving time in resolving issues [3]. Maintaining the operational capacity of the network is the main share of OPEX and in the long run, it is possible to keep these expenses constant by leveraging automation.

Network automation is summarized by planning, deploying, configuring, orchestrating, and assuring the networks and the services using software. Zero-touch or autonomous networks is the desired outcome of implementing network automation. In particular, the emphasis is on reducing human errors

and delays in network management and fault management.

Network automation initially requires increasing the operator's CAPEX to invest in technology that in the long run reduces the OPEX and alongside network upgrades, helps the operator to retain its customers.

3.3 Network Management

Today's network management paradigm is represented by the FCAPS management framework, which stands for Fault, Configuration, Accounting, Performance, and Security [16]. Fault management is one of the five functional areas of network management [17].

3.3.1 Fault Management

Fault management deals with multiple functions [16]:

- Fault detection: node level and network level mechanisms to automatically detect faults, errors, and failures, and share knowledge about such incidents throughout the operational lifetime of a node and the network.
- Fault diagnosis: the process of deducing the exact source/cause of an error or failure from the set of observed error/failure indications [18].
- Fault removal: automated fault removal or with human intervention.

In [16], Chaparadza says: *"a fault is the cause of an error. It is the physical or algorithmic cause of a malfunction. Errors may cause deviation of a delivered service from the specified service, which is visible to the outside world. An error in a network device or software may cause malfunctioning of dependent network devices or software."*

Modern networks are designed for the proliferation of devices, such as IoT devices, and there is great pressure on mobile operators to shorten service downtime and maintain a high quality of service [19]. Since networks have become critical components of modern organizations, faults and downtime can become very costly [20]. Thus, on the path towards zero-touch network automation, fault management is one of the key areas to automate to reach the goal.

3.4 Network Operation Centre (NOC)

The NOC (Network Operations Centre) is the location from where network monitoring and control is exercised [21]. Each operator can have one to multiple NOCs. The centers are in charge of keeping the network up and running by handling faults happening across the entire network or on a specific layer of the network e.g., one NOC dedicated to the RAN (Radio Access Network) and one NOC dedicated to optical communications.

To handle network faults there is a hierarchy with multiple levels of support:

- 1st Level Support (Tier 1): it is represented by the NOC, where engineers monitor the alarms, gather information, and analyze the symptoms to understand the root cause and how to fix the issues.
- if the NOC is not able to fix the problem, the NOC engineers request the intervention of the second line of support, which is made of specialized technicians that in general can solve more complicated problems. Tier 2 takes over incidents that cannot be resolved by Tier 1.
- 3rd Level Support (Tier 3): it can happen that an issue requires very deep knowledge about the equipment, in that case, the third level of support can help the second line of support to resolve the issue. 3rd Level Support is also the most expensive intervention since requires highly specialized technicians to fix a problem. This third line of support is usually outsourced and made available by the network equipment's vendor, e.g. Huawei.

3.5 Virtual NOC

The solution developed by Elisa Automate falls under the umbrella of Fault Management. Virtual NOC is an automation system in the form of a cloud-native application. It aims to eventually achieve zero-touch network automation by replacing the manual operations for fault management normally carried out by engineers in the NOC (Network Operation Centre).

The UI of the system allows to monitor different things, including the state of the network, the actions of Virtual NOC, and general health of the system. This information appears in real-time on the dashboard and depending on the case, two scenarios can happen:

- Partial automation: the NOC is not fully automated so Virtual NOC will work in support of the NOC Engineers.
- Full Automation: in case the NOC becomes 100% automated, the second line of support would take advantage of the UI for monitoring.

Like every system, we can consider it as a black box with inputs and outputs. By logging its activity, it produces statistics about its operations; those statistics will be mentioned in Section 12.1.

3.5.1 Value creation chain of Virtual NOC

Figure 1 represents a graphical explanation of what happen when Virtual NOC is placed into the picture of fault management.

Virtual NOC has multiple explicit benefits as visible in figure 1, including decreasing the cost of managing the network, offer continuous support 24/7, human-errors free intervention while fixing problems, and lower costs. These benefits can be seen from two value creation view points:

- Increase the quality of the services offered by the operator and reduce the portion of the churn rate related to service quality.
- Reduce the cost of running the network and allow the operator to be more competitive with mobile subscriptions at a lower price.

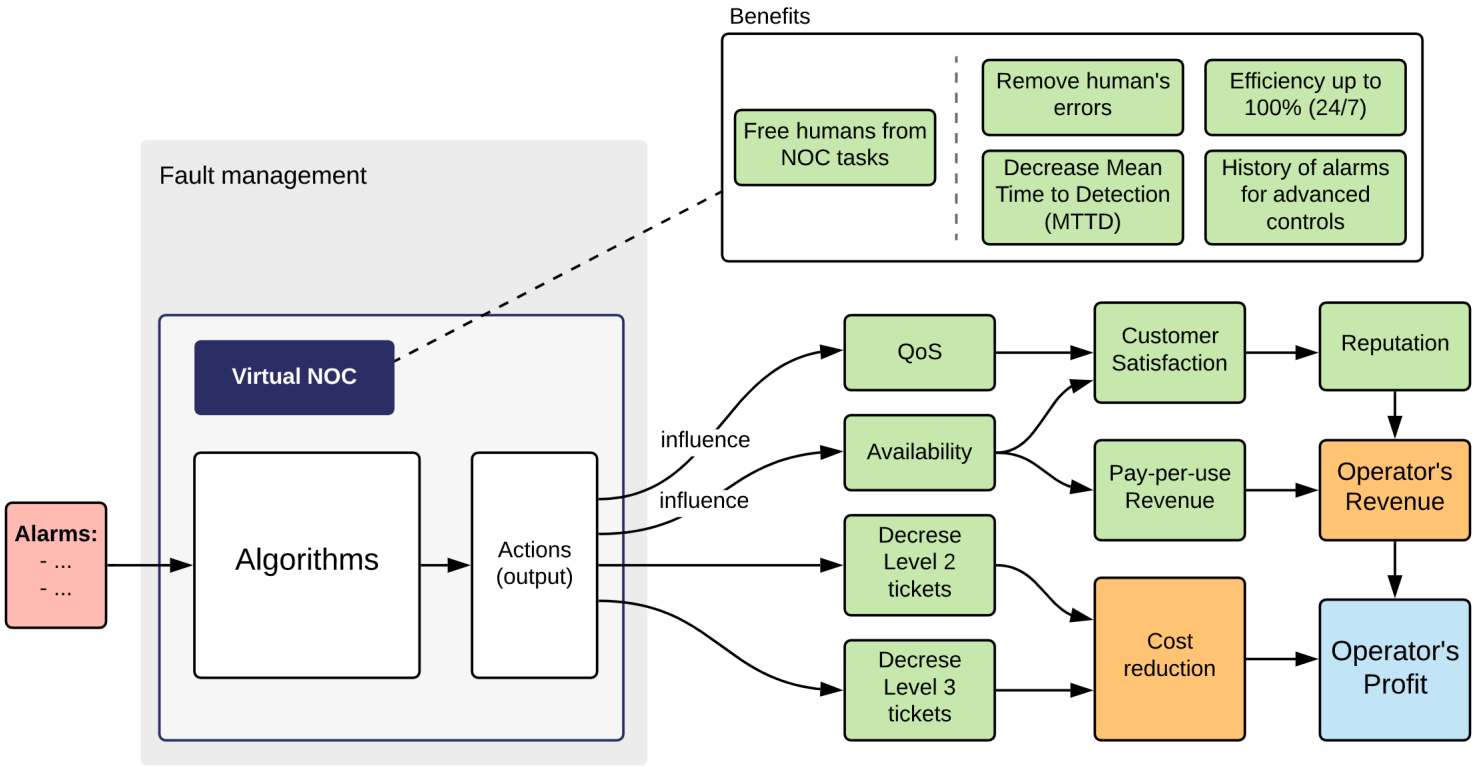

Figure 1: Value creation chain of Virtual NOC

4 Research Questions

This work aims to answer a major research question that can be expressed as follows:

How to prove the business value of Virtual NOC to new customers? How can we evaluate the actions of the automation system? Does OPEX savings represent the only reason to adopt this solution?

These high-level questions imply to deeply understand what Virtual NOC aims to achieve. In this section, I explore potential research sub-questions whose answers could be beneficial to the development of Virtual NOC.

4.1 Churn Rate Prediction

Theoretically, the adoption of such a solution will lead to improvements for the end-users. By automating the fault management process, the mean time to resolve a fault will decrease and the users will be less impacted by a service degradation.

Can we estimate the value of automation through the churn rate?

Thus, the idea to analyze the churn rate of an operator after the deployment of Virtual NOC and compare it with the churn rate without Virtual NOC. This analysis could be used as a selling point to convince new customers, with concrete outcomes.

4.2 Comparison of Human and Automated operations

The value of Virtual NOC can be explained using a calculation based on the FTEs (Full Time Equivalent) necessary to handle a set of Alarms. The resulting time (resulting FTEs) is an OPEX saving for the operator that can free human resources for more meaningful tasks. By leveraging automation solutions the operator can decrease the cost of managing a telecom network and keep pace with the constantly increasing requirements of the end-users of mobile services [22].

Can we Automate the OPEX decrease estimation for each customer?

In this case, answering this question would mean facilitate the task to Elisa Automate's sales team. Less related to the future development of the prod-

uct, it remains relevant since it is an exercise done for each new customer in the initial phase of a PoC (Proof of Concept) to showcase the most immediate advantage of automation, which is OPEX reduction.

4.3 Qualitative improvements with Automation

The complicated nature of these systems and the early stage of enriching network automation with cutting-edge technology lead to hesitation in promising concrete business outcomes [1]. By analyzing the chain of events that leads to user-impacting network malfunction, we may be in the right direction to formally prove the benefits of Automation.

What are the faults that affect the end-user? And how can automation address those and reduce the impact on Network Performance and Service Quality?

These questions, together with OPEX savings in 4.2, cover the spectrum of benefits that automation brings to the operators.

5 Research Question Selection

5.1 Selection process

The initial part of this work was focused on investigating and understanding the problem. Most of the time has been used to brainstorm with multiple experts inside the company, interview relevant people, and research the state of the art of fault management and surrounding topics. This research led to the following discovery: Elisa Automate, with its solutions, has been elevated by Gartner to one of the leading companies in what they define as Hyperautomation [1]. Gartner is a global research and advisory firm providing information, advice, and tools for leaders in IT and other fields [23]. However, the same firm highlights how these solutions are still struggling at quantifying concrete results from automation, which this work makes a step to make clear.

The initial phase also led to the discovery of multiple limitations that were preventing the realization of some ideas. In fact, the idea of churn rate prediction in paragraph 4.1 is left as future work. The details and the current limitations are explained in paragraph 12.1.

5.2 Sub-questions 4.2 and 4.3

In this paragraph, I explain why the remaining sub-questions 4.2 and 4.3 are still very important to answer the initial research question in Section 4.

5.2.1 Importance of 4.2

The comparison between manual and automated operations in 4.2 is particularly interesting for two reasons:

- It can be used initially to accelerate the proof of concept (PoC) of Virtual NOC for new customers.
- And secondly, it can be used as an additional feature in the Dashboard (see 3.5) of Virtual NOC to constantly measure the daily, monthly, or yearly savings of automation.

On top of the advantages of introducing an automated system, listed in the above bullet points, in case the estimation is not satisfying or considered inaccurate by the customer, it would be possible to re-executing it to find a common agreement. By adding information coming from the customers, the precision can be increased.

The aforementioned comparison is currently done manually in several steps as shown in Figure 3, in Section 7, where the current process and the work that I did to add automation to the PoC are further explained.

5.2.2 Importance of 4.3

The qualitative improvements with automation discussed in 4.3 suggest an investigation on a topic with little research available, even inside the company:

- selection of alarms that cause service quality degradation or interruption for the user and measurement of the impact of that degradation through counters and KPIs available from the network.

The goal is to be able to measure the alarm impact and once there will be more data available about Network Actions (e.g. remote reset of network elements) executed by Virtual NOC, correlate those automation system's statistics with the changes in the alarm impact. By handling the fault management with an automation system, the reaction time decreases, the alarms

can be resolved in parallel without queuing, and the efficiency increases, thus, it will be possible to show the automation's results. In Figure 2 you can see a graphical explanation of 4.3.

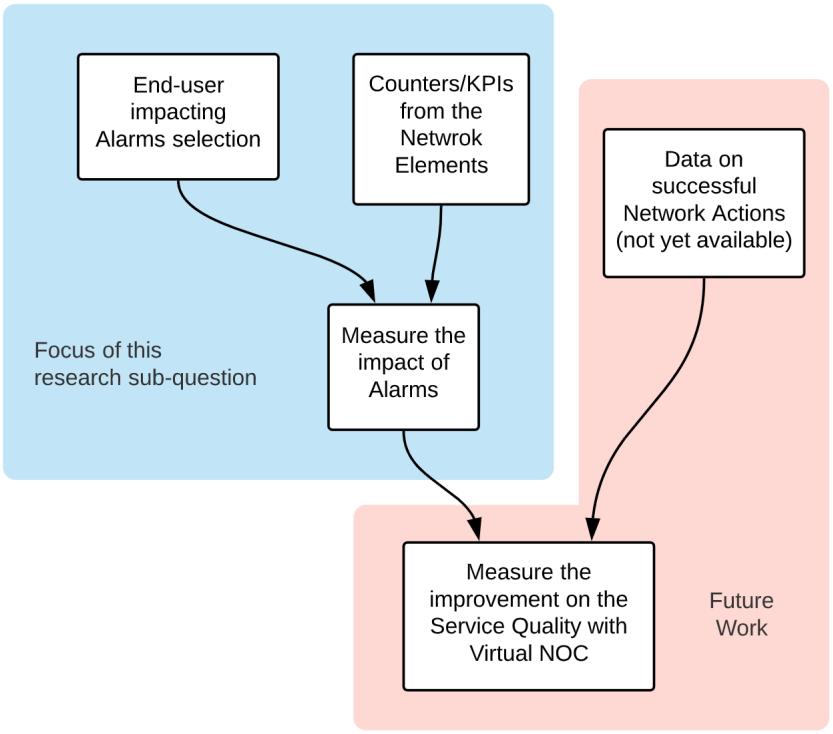

Figure 2: Objective of measuring the impact of alarms.

6 Business Goal

The B2B telecom sector is extremely competitive and dominated by giant groups. Elisa has proved the value in-house, on its network. However, Elisa Automate is looking for a **dynamic approach** to objectively prove Virtual NOC's business value in any customer environment.

This work aims to prove the business case of Virtual NOC. It helps to gain a holistic view of Virtual NOC and understand how actions match business outcomes. By formalizing the promises of this automation solution, this work gives more tools to Elisa Automate to affirm its solution on the market.

6.1 Methodology

One of the non-functional requirements of this work was the necessity of utilizing these results as much as possible, i.e. find an approach that can be used both with new customers and with existing customers. In order to accomplish the business goal, I identified two realistic ways of doing it. These two ways come from the two sub-research questions 4.2 and 4.3.

For the first, about the comparison of human and automated operations, I will explain the current status and the new approach that I propose. The details are available in Section 7, where especially in the second half, I explain how I implemented the proposed approach.

For the second sub-question, about the investigation on qualitative improvements with automation, I will report the findings of the analysis on the correlation between alarms and network counters and KPIs that I carried out during this work. This part is available in Section 9 and following.

7 Time Savings Estimation

Referring to Paragraph 4.2 and 5.2.1 on *Comparison of Human and Automated Operations*, this section and Section 8 further explains how the current process is done and how I implemented a solution to accelerate the heaviest step of the analysis, namely the estimation of the time required to act on the alarms. This section also showcases the results and possible improvements.

7.1 Process Overview

In Figure 3, you can see how the current process appears and how it is entirely manual. This way of carrying out the PoC (Proof of Concept) is very time-consuming and unless another long time is invested in it afterward, it leaves very little room for adjustments. Since the PoC is a common step with new customers to prove the ability and the capabilities of Automate's team to deliver their promises, it has been the focus of this thesis work since the beginning. This process cannot be currently done continuously in production to show the customer how much Virtual NOC is saving.

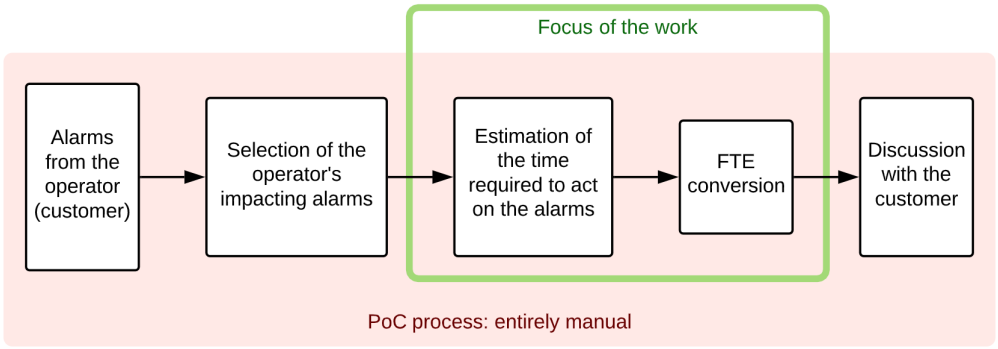

Figure 3: Flowchart of the current process to calculate the OPEX savings.

In order from left to right, the following paragraphs explain the individual steps visible in Figure 3.

7.1.1 Alarm collection

First, the operator exports one week of alarms from its OSS (Operation Support System). Afterward, the alarms are sent to Automate's team through

a secure connection since they contain very sensitive information. They are generated from the operator's network and can contain names and geographical information. In some cases, a security clearance can be required to access those data.

7.1.2 Impacting alarms

On a separate discussion, the operator highlights what are the most impacting alarms they receive and that they trust automation to take over. Then based on the customer's objective for automation, i.e. what are the network problems they want to cover with automation, the alarm types to include in the PoC are selected.

7.1.3 Time estimation

The third step is done based on the impacting alarms selected by the customer. It is the most demanding step since it requires manual analysis of thousands of alarms and it is done by the Automate team. Common tools, such as excel or python, are used for this analysis.

Value	**Category**
1	Fault alarm
2	Clear alarm
3	Event alarm
4	Acknowledged alarm
5	Unacknowledged alarm
9	Change alarm

Table 1: Huawei alarms Category classification.

This process simulates what Virtual NOC would do when the same alarms arrive at the system. It is a manual replica of the automation system to show what the outcome would be, what the decisions taken by the automation would be, and what the actions and tasks delegated to humans would be.

For example, among those alarms, they first need to identify the ones related to faults and remove other categories of alarms, the other categories are visible in Table 1. Then with the alarms left, a temporal analysis to see on which days of the week there are more faults can be carried out. Knowing

the pick hours with more faults can be useful for the next steps. It is also important to understand what the support levels that need to be involved are, see support levels hierarchy in Section 3.4. Then based on this and other information, the team can move on and isolate the alarms the customer cares to handle with automation and estimate the time required for the different activities related to fault management. The resulting alarm occurrences are counted and the time to clear them is calculated.

7.1.4 FTE conversion

The time estimated in the previous step is then converted to full-time equivalent (FTE) and used as a baseline to explain to the customer what would be the advantage of deploying Virtual NOC. The FTE is used as basis for effort reduction and the freed up resources can be used for other more useful work.

7.1.5 Discussion with customer

Finally, the results of the analysis are presented to the customer and a discussion about the method of the analysis follows. At this stage, many questions and critics can come from the customer, who may not agree with the assumptions taken to calculate the FTE result. Iterate the analysis with a new assumption can be time-consuming and challenging.

The above point is one of the key reasons to work on the automation of step 7.1.3 and 7.1.4. The next section explains the system to accelerate the current process.

8 Automated Time Savings Estimation

As mentioned at the beginning of Section 7, in this paragraph, I will go through the proposed system architecture to carry out the PoC with new customers. Here I will focus on the green rectangle visible in Figure 3, which includes the time estimation step and the full-time equivalent (FTE) conversion step.

Starting from the easier part, I will first discuss the FTE conversion step and later go through the third step, which took a more consistent part in this work.

8.1 FTE conversion improvements

As explained in 7.1.4, this is not a heavy step from a time perspective. It consists of applying a transformation to the time estimation to obtain the same value on a different scale, namely the full-time equivalent scale.

Usually, the FTE conversion is week-based, when an employer has a 40-hour workweek, employees who are scheduled to work 40 hours per week are 1.0 FTEs [24]. However, in the context of fault management, where the work required to fix faults cannot be spread over a long period of time, I find it more appropriate to do the conversion on a daily basis. This means that when an employer has an 8-hours workday, employees who are scheduled to work 4 hours per day are 0.5 FTEs. This decision is also in the perspective of using this new approach continuously in production. By doing this, it is possible to show the savings of Virtual NOC on a daily basis. These daily savings can then be aggregated to achieve the desired granularity, which can show the savings made in a few days, in a month, or over a trimester.

8.2 Time Estimation improvements

This paragraph is the main answer to sub-question 4.2. Here and in the following paragraphs, I explain how the current approach (see 7.1.3) has been replicated with a software system to minimize the manual work. A wide-ranging discussion will cover topics such as the logic of the system, the software architecture for the implementation, the results of one test analysis, and the possible improvements.

8.2.1 Idea recap

To start in order, I will first go through the logic behind the idea. The original time estimation is a fairly simple process where the relevant alarms are isolated and a time is assigned to each type of fault. Once you have the time for each alarm, you sum up all the times and that would result in the total time required to fix those alarms. The complexity comes into the picture when you have thousands and thousands of alarms, from which isolating the relevant ones is not a straightforward process, as explained in 7.1.3.

Figure 4 displays the flowchart of how I propose to estimate the time. On

the left, you can see a set of algorithms with fictional names (X, Y, Z, etc...). Each algorithm in Virtual NOC is designed to handle a specific subset of alarms, simply because a similar fault can generate slightly different alarms based on the situation, and the procedure to fix them can be very similar. In a common deal, the customer agrees with Automate to design enough algorithms to cover the agreed percentage of alarms, see 7.1.2.

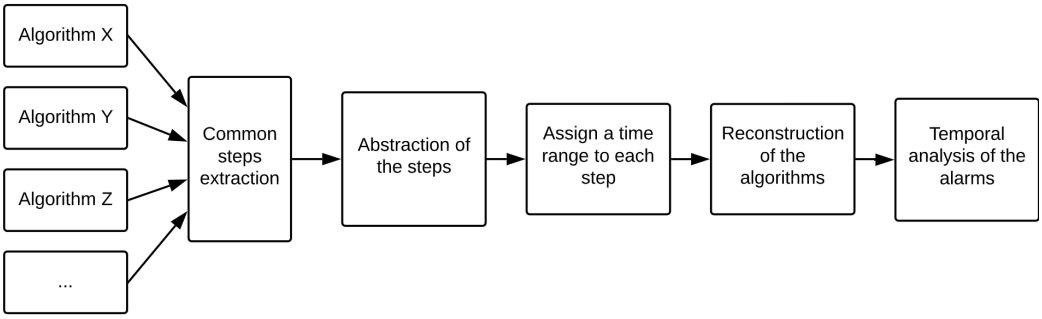

Figure 4: Flowchart of the procedure for the new time estimation.

Here I considered algorithms coming from different customers to have a wider spectrum of alarms and restoring procedures. In particular, I considered enough algorithms to extract almost all the possible steps that can be needed during such procedures.

Multiple steps extracted from the algorithms were very customer-specific, e.g. query planned site visits from a particular ticketing system. Although those steps referred to specific configurations, other customers may have similar systems, thus, I classified the steps into more general ones to be reusable across customers in a later stage.

The next block is about assigning a time to each step, which is where the name estimation derives from. It is difficult to assign a precise time to actions and obtain an estimation that is generic enough to adapt to different customers and configurations. For this reason, following the suggestion of my company supervisor, we decided to assign a time range to each step. The time assignment has been carried out with other experts inside the company who had previously worked as NOC engineers.

Once the common steps were identified and a time range to execute them was assigned, I moved on to reconstruct the algorithms. The newly created

algorithms can be seen as a simulation of the original ones to simulate time execution.

Finally, the last step is actually using the new tool to carry out the time analysis of some alarms from a current customer to test the solution.

8.3 Software Architecture

This paragraph will illustrate the software architecture this tool's implementation is based on. It will go through the main classes and expand the description begun in 8.2.1.

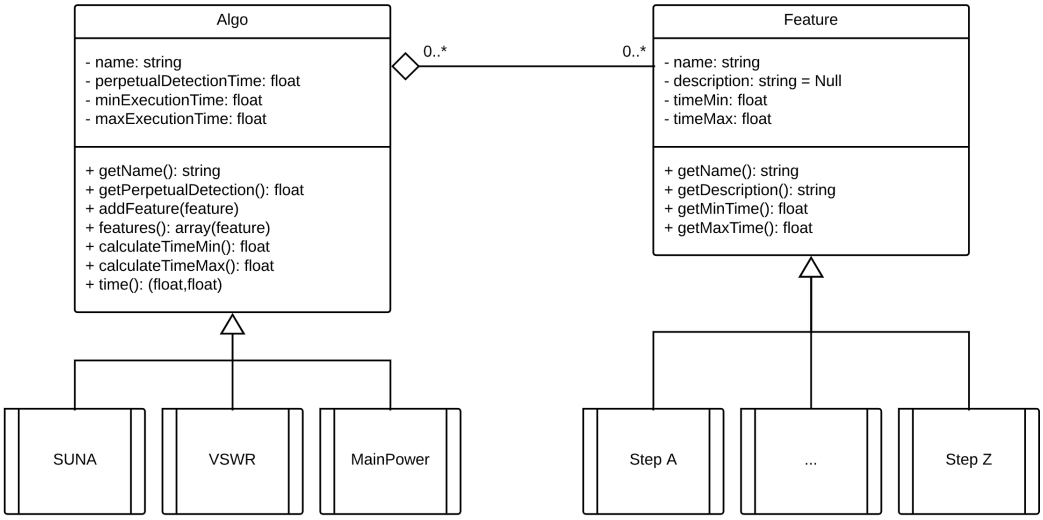

Figure 5: Time estimation UML Architecture.

As you can see from Figure 5, there are two main classes:

- Feature
- Algo

These two classes are used as a basis to inherit the specific attributes. There is an aggregation link between the two with the multiplicity that says that an algorithm doesn't have a limit on the number of features or steps that can contain. The features do not belong exclusively to an algorithm

8.3.1 Feature Class

For convenience during the coding, each step mentioned in 8.2.1 has its own class, it has the same attributes as the mother class Feature. The name is the only thing changing.

In this class, there are self-explaining attributes and methods. To represent the time range that the specific step can require, I used the private attributes *timeMin* and *timeMax*.

8.3.2 Algo Class

Similarly, for algorithms, there is the mother class Algo which contains all the attributes and methods needed, then the algorithm-specific class just takes a different name and specific initialization parameters. Among all the characteristics of this class I will explain the following:

- perpetualDetectionTime: it is a private attribute that tells us how long the algorithm waits before starting to execute the recovering procedure. Every algorithm has a different delay that is issue-specific and can be customer-specific as well. In practice, it waits that time to make sure that the triggering alarms are not caused by some temporal issue.
- minExecutionTime: private attribute and it represents the sum of the timeMin of each feature contained in the algorithm.
- maxExecutionTime: it represents the sum of the timeMax of each feature contained in the algorithm.
- addFeature(feature): it is the public method to add features (steps) to the algorithm.
- features(): a public method that returns the features added to the algorithm as an array.
- time(): a public method that returns the minimum time and the maximum time to execute the algorithm. Inside it calls the methods calculateTimeMin() and calculateTimeMax().

For the first test I have implemented the following algorithms:

- SUNA: Service Unavailable

- VSWR: Voltage Standing Wave Ratio
- Main Power

8.4 Results

To test the implementation described in the previous paragraphs, I used one day of live alarms from a current customer of Automate. In this case, we were not particularly interested in customer-specific alarms or configurations but rather in real alarms coming from a network build with Huawei equipment. This gave me the opportunity to test the logic of my implementation on real data.

This is the output of the simulation without FTE conversion:

```
Out of 1980 alarms with alarmCategory FAULT,
the three algorithms can handle 93 alarms (4.7%)
corresponding to 36 tickets.
Time savings between 8h51' and 37h24'
```

It shows that out of the initial 5000+ alarms received in one day, only 1980 came from relevant categories. The number of alarms that the simulation can handle is dictated by the number of algorithms implemented. Since there are only three algorithms, the number of alarms is limited to 93. Moreover, out of those 93 alarms, only 36 tickets were created and here you can already see the benefit of automation. Without it, the NOC engineers would have to deal with the 93 alarms and realize only afterward that there are only 36 network elements with faults that are sending multiple alarms during the day. Since the life span of an alarm can be up to hours, when multiple alarms come from the same element within few hours they are usually considered as a single trouble ticket.

The simulation estimates that to solve those 36 issues, the time needed can vary between a bit less than nine hours and up to thirty-seven hours. This large time range comes from the fact that the current implementation either calculates the best possible scenario or the worst possible scenario. The best scenario is given by the sum of the minimum time of all the steps required to fix the issues. The worst scenario is given by the sum of the maximum time of all the steps required to fix the issues.

According to the practical experience of some Automate's member, to have a more precise and conservative estimation we can take a value slightly below the average of the two. Formulas explaining the calculation follow.

8.4.1 Calculations

In this paragraph, I continue the explanation of the results with the formulas used in the simulation.
Among the analysed alarms there are:

- a_s: number of triggering alarms for SUNA algorithm
- a_v: number of triggering alarms for VSWR algorithm
- a_m: number of triggering alarms for MainPower algorithm

These parameters characterize the algorithms:

- s_s: steps(features) in SUNA
- s_v: steps(features) in VSWR
- s_m: steps(features) in MainPower

The minimum time to execute the algorithms is thus given by the following formulas, note that *timeMin* refers to one of the attribute of class Feature seen in paragraph 8.3.1:

$$\text{SUNA min time} = \sum_{i=1}^{s_s} \text{timeMin}_i \tag{1}$$

$$\text{VSWR min time} = \sum_{i=1}^{s_v} \text{timeMin}_i \tag{2}$$

$$\text{MainPower min time} = \sum_{i=1}^{s_m} \text{timeMin}_i \tag{3}$$

By combining now the above formulas, the minimum time to resolve all the issues, called lower bound, can be calculated:

$$\text{tot Lower Bound} = a_s \sum_{i=1}^{s_s} \text{timeMin}_i + a_v \sum_{i=1}^{s_v} \text{timeMin}_i + a_m \sum_{i=1}^{s_m} \text{timeMin}_i \tag{4}$$

The maximum time required or upper bound can be calculated with an analogous formulas and using the attribute *timeMax*:

$$\text{tot Upper Bound} = a_s \sum_{i=1}^{s_s} \text{timeMax}_i + a_v \sum_{i=1}^{s_v} \text{timeMax}_i + a_m \sum_{i=1}^{s_m} \text{timeMax}_i \tag{5}$$

And finally, to have a realistic estimation, the value around the mean of the minimum time and the maximum time is calculated:.

$$\text{Realistic Estimation} \approx \frac{\text{tot Lower Bound} + \text{tot Upper Bound}}{2} \tag{6}$$

8.5 Possible improvements

As explained in paragraph 8.4, the simulation provides the estimated time needed to handle the alarms as a time range. With the current settings, the time range can be very long. For instance, implementing 10 algorithms, thus, increasing the number of alarms that can be handled, the estimation can have a very large time difference of days between the minimum time and the maximum time, starting with the same alarms as the above simulation. Thus this needs some improvements.

The problem is due to the tree evolution of the algorithms. In Figure 6, an example of the tree development is reported. Since the steps to follow can vary based on the specific situation, the same alarm can follow different branches in the algorithms and that implies longer or shorter execution times.

One way to improve the accuracy is adding granularity to the initial settings, thus, allowing the user to set more parameters about the alarms to reflect real scenarios. This would reflect in a division of the occurrences of an alarm that follows branch A, branch B, and so on. E.g., if there are 100 occurrences of alarm XYZ, 10% of them will take branch A, 30% branch B, and the rest will go through the remaining branches with similar execution times. The formulas for the calculation in 8.4.1 would require adequate modifications as well. An example of the modifications for SUNA algorithm is shown below.

New parameters for algorithm SUNA:

- s_{s_comm}: SUNA common steps
- XX: branches in SUNA, identified by a letter(s)

- P_A: percentage of alarms going through branch A
- ...
- P_{XX}: percentage of alarms going through branch XX

Each branch, on top of s_{s_comm} steps, has a number of steps unique to itself and they are indicated as follow:

- s_{s_A}: steps unique to branch A
- s_{s_B}: steps unique to branch B
- ...
- s_{s_XX}: steps unique to the last branch

And this is the formula to calculate the minimum time required by all the alarms handled by SUNA:

$$\text{tot SUNA min time} = a_s \sum_{i=1}^{s_{s\_comm}} \text{timeMin}_i + \sum_{j=A}^{XX} (a_s P_j \sum_{z=1}^{s_{s\_j}} \text{timeMin}_z) \quad (7)$$

This improvement would minimize the time range and provide a more realistic estimation that can be easily simulated multiple times with different amounts of alarms going through each branch in the different algorithms. However, this would not eliminate the time range because there is still uncertainty in the precise time required by each step. Lowering the difference between maximum time and minimum time to few hours would be perfectly acceptable for this use case.

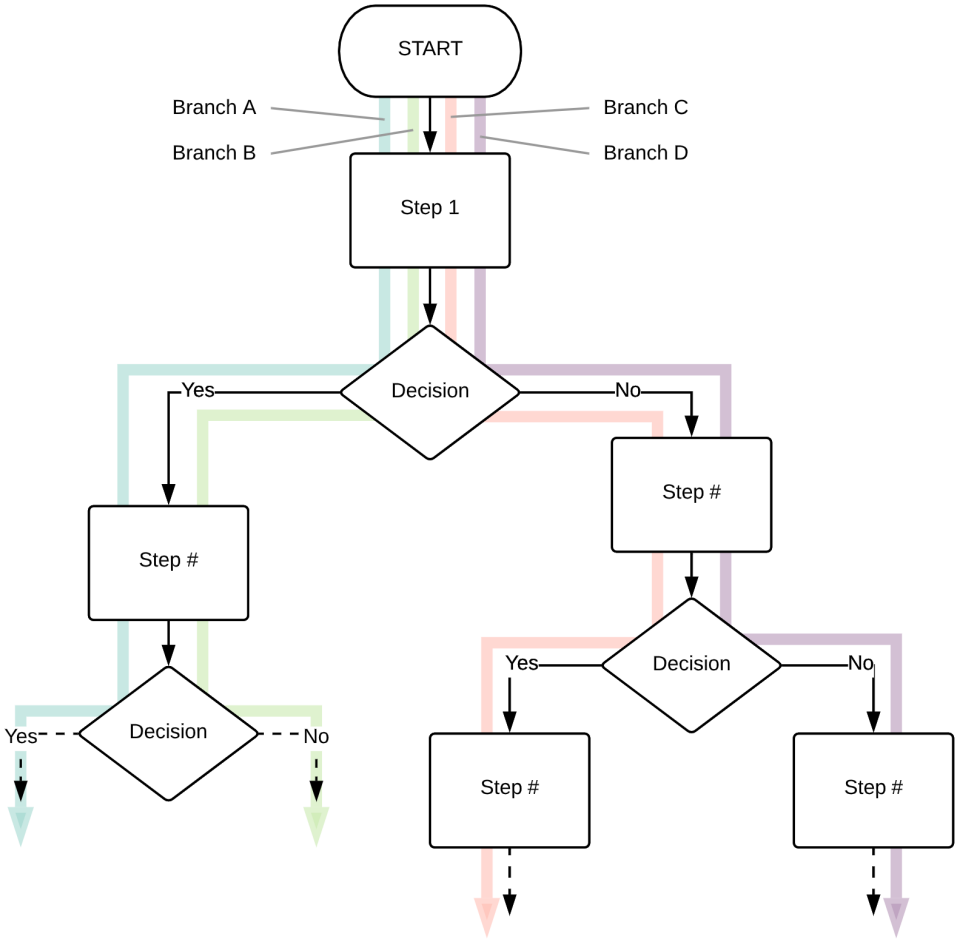

Figure 6: Tree structure of a general algorithm.

9 Alarms Impact

We change the topic now and we refer to the Quality Improvements with Automation sub-research question, see Paragraph 5.2.2. This section and the following ones explain the chosen approach to arrive closer to estimate the impact of faults on the operator's subscribers. This work represents the area marked as *Focus of this research sub-question* in Figure 2.

Alarm impact can be analyzed using the following metrics:

- Number of affected users: the number of people who noticed a problem with their mobile connection;
- Length of the outage: minutes or hours during which the affected users experience disservice, e.g., carrier aggregation example in 9.1.2.

The alarm impact calculation is a challenging problem but once performed, it is a good metric to compare fault management processes with and without automation. Once the results of this part are presented at the end of Section 10, Section 11 expands on the importance of ensuring high network performance for the benefits of the mobile operator.

9.1 Complex Systems

9.1.1 Availability in a complex system

Availability is a fundamental quality of information systems. It represents the quality of being able to be used when requested, without delay. Availability in a complex system, such as a telecommunication network, represents the quality of being obtained or used of every single component that together forms the system. The total availability is given by the sum of the time during which every single component of the system work. The unavailability is given by the sum of the time during which even a minor component alone stops working. The availability calculation then varies depending on the system's topology; for simplicity, consider a system where all the components are dependent on each other (in series) and a single fault causes part of the system to stop communicating. The bigger the system, the higher the probability at a given time that one of the components of the system stops working. In a mobile network, the system is designed to avoid a single point of failure, as opposed to the example above. Usually, a fault of a RAN (Radio

Access Network) component only affects a small geographical area. In those situations, the neighbor cells can take over the traffic not carried by the faulty cell. An isolated fault is unlikely to affect the end-users unless there is a particular congested situation. Nevertheless, the number of faults can quickly grow in a complex system, such as mobile RAN, and if not handled correctly it can result in a service outage.

9.1.2 User-affecting faults

Although telecommunication networks are fault-tolerant in general, there are cases where the faults have a direct impact on the network users.

In 4G networks, losing a cell in an eNodeB can imply the loss of carrier aggregation [25]. In this situation, a UE (user equipment) is connected at the same time to multiple carriers, whose frequencies belong to different bands (different cells) to increase the bandwidth and the throughput. One 4G band can be the 800 Mhz and the second the 1800 Mhz. Supposing that the 1800 Mhz cell stops working for a fault, all the users that were taking advantage of carrier aggregation all of a sudden experience lower throughput. Especially in data-hungry applications, such as video streaming or online gaming, a degraded performance is felt immediately. Section 11 contains a detailed analysis of how user-affecting faults impact the operator revenue.

In Paragraph 9.2 and the following ones, the procedure to identify these user-impacting alarms is described.

9.2 Alarm Labels

Equipment vendors, i.e. Huawei, Nokia, etc., use different names and different grouping systems for the alarms of their network elements. Each alarm contains multiple labels that are used to help to troubleshoot. In this work I mainly considered two of these labels:

- Severity
- Type

Considering that alarms are reported in text format, it is difficult to find insights at a glance, thus, some initial analysis was required to get visual insights. I carried out clustering of the alarms based on severity and on alarms' type. The results will be shown in paragraphs 9.5.

Note that the number of different alarms available in a modern telecommunication network is very high. For Huawei, I found around five/six hundred alarms browsing the documentation of the main network elements [26]. In reality, most of these alarms are minor communications and sometimes ignored by the operators, e.g. they are warnings regarding the expiring date of the licenses. In this work, I considered around one hundred alarms, considered impacting for the operators. This list of alarms came from a collection of alarms communicated to Automate by different customers.

9.3 Alarm Severity Classes

In Table 2, you can see the severity classification for Huawei [27].

Value	**Severity**	**Description**
1	Critical	Indicates that a fault affecting services has occurred and it must be rectified immediately.
2	Major	Indicates that services are being affected and related measures need to be taken urgently.
3	Minor	Indicates that a fault occurs but does not affect services. To avoid a minor alarm from getting severer, related measures must be immediately taken.
4	Warning	Indicates that a potential or impending service-affecting fault is detected before any significant effects have been felt. Take corrective actions to diagnose and rectify the fault.
6	Clear	Indicates that a fault has been cleared.

Table 2: Huawei alarms Severity classification.

Each network installation presents differences, resulting in modification of the severity class originally assigned by the vendor. There might be cases where the severity increases due to the installation. To further explain this matter, let's assume that *Huawei* assigned the label *Critical* to an alarm that turns on when the power supply of a network element is missing. If the operator installs a UPS (Uninterruptible Power Supply) next to the network element to keep it operational during power outages, the severity of that

alarm is not critical anymore. Since the fault is no longer service affecting, it can be assigned a lower severity.

9.4 Alarm Type Classes

Huawei alarms are also assigned to a *type* class. Examples of types are reported in Table 3.

Value	Type
1	Power system
2	Environmental system
3	Signaling system
4	Trunk system
5	Hardware system
6	Software system
8	Communication system
9	QoS

Table 3: Huawei alarms Type classification.

When an alarm is received this information is used to follow a sequence of steps to resolve the issue that are type-dependent. E.g., if the type is *Power*, one of the first steps will be checking whether there is or not a power outage.

9.5 Alarm Visualizations in Clusters

9.5.1 Clusters based on Severity

The first clusters in Figure 7 show how the impacting alarms are divided based on the severity level. As you can see in the same image, there are around 20 alarms in total that are classified into two classes, i.e. green, red and brown balls. This comes from the fact that, depending on the situation and the type of abnormal behavior in the network element, the alarm can have a smaller or wider impact.

9.5.2 Clusters based on Type

Here, the clusters in Figure 8 show the alarms divided by type. The type tells us what the origin of the alarms is. Using a Hardware alarm as an example

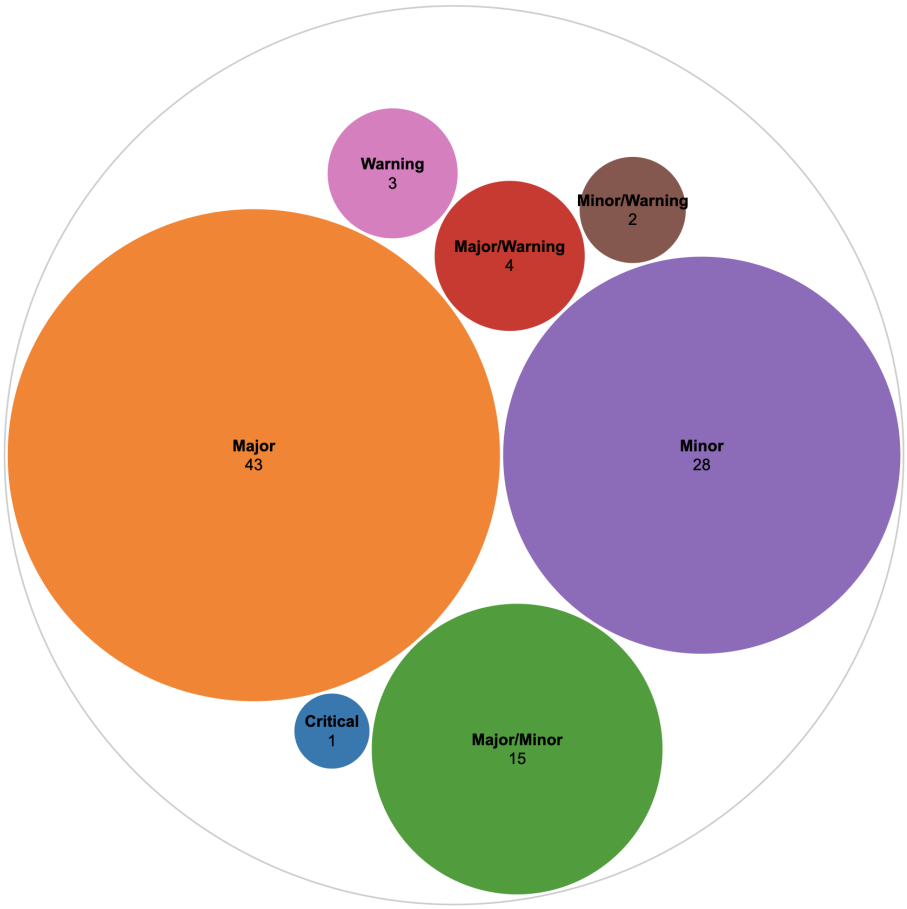

Figure 7: Alarms grouped by severity.

we know that it refers to a physical problem, e.g. a feeder cable not perfectly connected and causing a power mismatch.

9.5.3 Clusters based on Severity and Type

Another interesting view is the classification of the alarms based on both labels mentioned before, Severity and Type. Figure 9 shows how the alarms are divided per type inside the severity class.

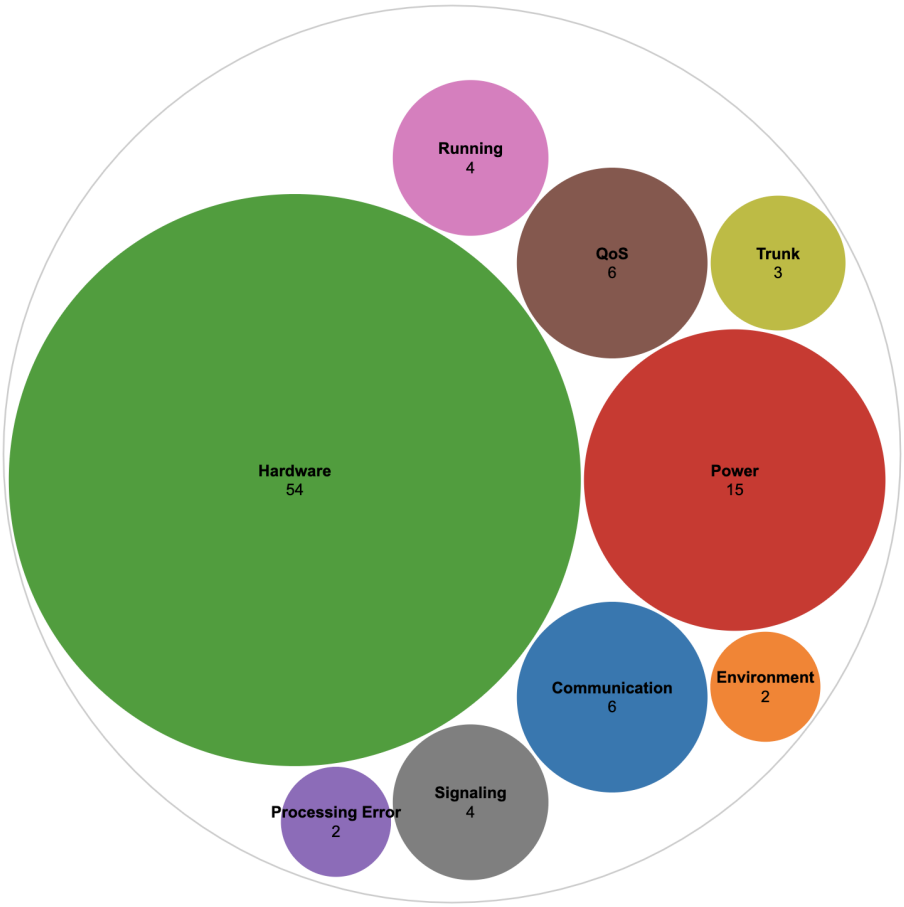

Figure 8: Alarms grouped by type.

9.6 Selection of the Alarms for the Analysis

In this work, alarms originated by Huawei equipment are used and the severity class assigned by the vendor is maintained. Some customer-specific information is also used to enrich the assigned severity class and avoid the problem related to the uniqueness of the installation, mentioned in paragraph 9.3. The type class was not particularly useful for the selection of the alarms since no key information was added to the pre-analysis. It was however an interesting exercise and helped to better understand how alarms are divided, especially thanks to Figure 9.

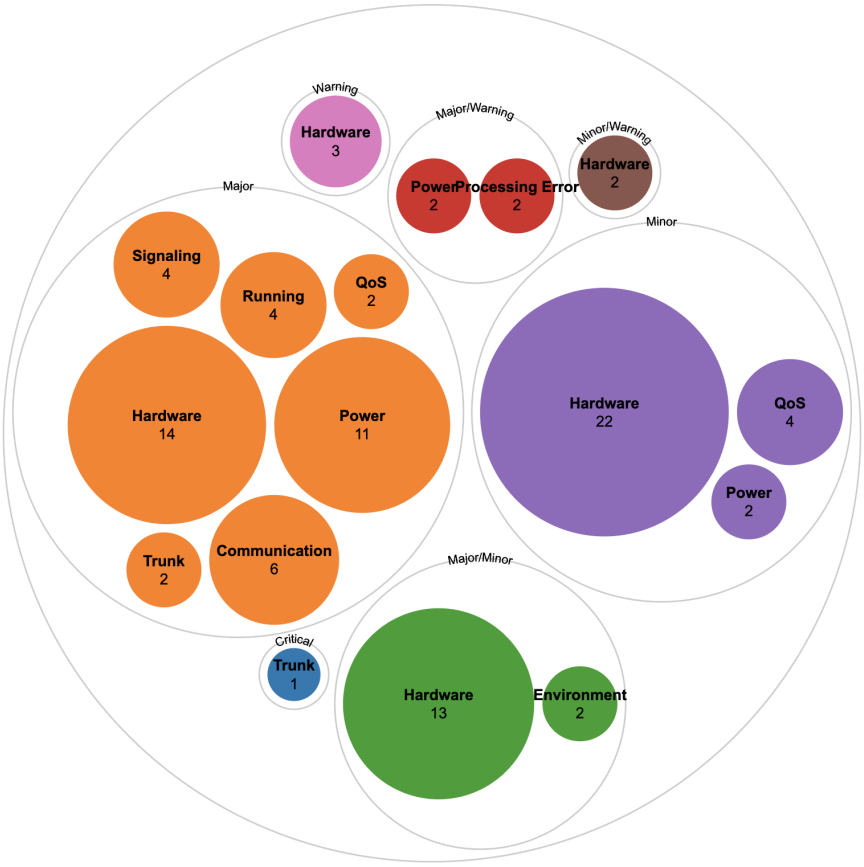

Figure 9: Alarms grouped by severity and type.

To select the final list of alarms to carry out this investigation, the following criteria were considered:

- Severity class: this gives a good starting point to differentiate the alarms based on the vendor assigned indication.
- Technology: LTE related faults were considered.
- Related Algorithm: the studies done by Automate team already gave important information to understand what is the impact of a specific alarm; alarms handled by SUNA (Service Unavailable) algorithm were considered first.

- Impact on the System: according to Huawei documentation, each fault can cause service interruption more or less frequently, sometimes the alarm generated by different faults is the same, thus, only the alarms with a high probability of service interruption were considered.

And in Table 4 you can see the chosen alarms.

Alarm	**Name**	**Severity**	**Alarm Impact**
29213	eNodeB S1 Control Plane Transmission Interruption	Critical	None of the S1 interfaces of the eNodeB can be established, cells cannot be established, and UEs cannot access the network [28].
26235	RF Unit Maintenance Link Failure	Major	Ongoing services carried on the RF unit are interrupted [29].
29240	Cell Unavailable	Major	The cell where the alarm is reported cannot provide services [30].
26204	Board Not In Position	Major	The board fails to work properly, and therefore the ongoing services carried on the board are interrupted [31].
301	NE is Disconnected	Critical	If the NE is disconnected, the services interacted between the U2020 and the NE cannot run properly [32].

Table 4: Selected alarms for the analysis.

10 Analysis Process

This section aims to describe the data sources and the methods adopted for the analysis. It will explain (i) the data used (anonymized to preserve sensitive customer information), with its limitations, (ii) the data analysis method and (iii) the results.

10.1 Data Sources and Screening

Initial disclaimer: the data used in this work is highly confidential, thus, only some extract may have been reported to maintain anonymity.

The alarm data was available from the database running on the customer cloud environment and the data was accessible by running SQL queries to the database. The alarms event are generated in the OSS (Operation Support System) which is the information processing system that assists an operator to manage its networks. The alarms are structured in JSON files and the most important fields are reported in Table 5.

Field Name	**Description**
vendor	name of equipment manufacturer, e.g. Huawei
metadata	field with sub-fields related to the alarm event
alarmCategory	sub-field of metadata, describes the nature of the alarm
alarmID	sub-field of metadata, code identifying the alarm, eg. 29240
alarmMO	sub-field of metadata, alarm managed object, the network element not working properly
raiseTime	sub-field of metadata, time when the alarm was raised in the managed object
alarmClearTime	time when the alarm was cleared
alarmProbablecause	hint about the cause of the alarm

Table 5: Alarms structure.

On the other hand, the PM (performance management) counters are not stored in a dedicated database, they are only available in the OSS. For other

purposes, Automate have an ingestion pipeline to gather the PM data and use them in other solutions. However, due to a technical issue, I was only able to get the raw data from the customer OSS. This raw data consist of a large compressed file per day of data. Inside this file, each customer site (or base station location) is represented in a folder containing 96 XML files reporting the PM counters. Every 15 minutes the values of the counters are updated and saved in one XML file, resulting in 96 files at the end of 24 hours.

Once the relevant alarms were identified in 9.6, the next step has been to find a period during which both alarms and PM counters were available. From the days of alarm network data available, I compared the sites where the aforementioned alarms happened. From this comparison, I selected the one day with the highest amount of alarms, later used to carry out my analysis.

10.2 Data Preparation

The preparation of the both types of data, alarms and PM data, presented many difficulties that will be explained in this paragraph. First of all, both data sources were not ready to be used and a large amount of time has been invested in manual analysis and parsing.

In particular, for the alarms, I widely used excel to have a first visualization of the data. The visualization consisted of looking over the data to look for flaws, carrying out a visual check. To isolate the relevant alarms that passed the visual check, I used pivot tables and condition formatting rules.

For the PM data, since manual parsing was not feasible for the amount of data, I used an internal parser developed by Elisa Automate team. With that parser, I was then able to prepare the data in a suitable format for the analysis. Also in this case, I took an intermediate step with excel to perform a visual check on the data. I checked the consistency of the measurements, the values of the measurements and in general the dimension (96 measures per counter).

The intermediate step through excel was key to understand immediately the limitations of the available data. While for the alarms the system was very reliable, with consistency in the data acquired and all the fields properly filled, for the PM counters I found many issues with the acquired metrics. First of all, across 5 sites analysed, 16.7% of counters on average were inconsistent

with only few measurements through the day (less than 96 gauge points). All these counters were discarded since they were meaningless for the analysis. Moreover, around 66% of the counters with consistent measurements were at zero throughout the day. Since for different sites, different counters were consistently at zero, I excluded the hypothesis that the correct value of those counters was zero. The reasons behind the situation described above are out of the scope of this work.

To fully appreciate the extent of the data used, it is useful to give an approximation of the dimension of the sets:

- For the selected day of **alarms**, more than 17000 events were present across all sites of the mobile network. Due to the limitation of the manual phase necessary to familiarize with the structure of the data, only 5 sites were used. The selection of the sites was based on the highest number of alarms in Table 4 occurring on that day.
- For the **PM data**, each site had 34300 counters on average. After dropping the inconsistent counters and the ones at zero, each site had 9600 counters usable on average. However, there were big differences among the sites considered.

10.3 Counters Selection based on Domain knowledge

The next phase of the analysis has been to isolate some theoretically relevant counters to estimate the impact of the alarms on the users. From Huawei documentation [33], I researched counters related to LTE metrics that could describe the utilization and the performance of a specific site, including the amount of users connected to a node, the number of active users, the number of packets exchanged with the node, cell unavailability duration and others.

For the problem of scarce measurement quality of the counters, mentioned in 10.2, most of the counters researched were not available. In Table 6 you can see some of the available counters among the researched ones, while in Table 7 you can see some of the crucial counters that were not available.

The original idea was to use especially the counters in Table 7 to create a KPI to measure the number of impacted users. For instance, a simplification

Counter ID	Description
1526736788	Maximum Number of UL-Sync UEs in an eNodeB.
1526728320	Duration of Cell Availability.
1526746968	Uplink or Downlink main traffic time (MTT) in a cell.
1526728261	Total downlink PDCP-layer traffic volume in a cell.
1526729005	Downlink PDCP-layer traffic volume sent in the last TTIs before the buffer is empty.
1526729015	Data transmission duration excluding the last TTIs before the downlink buffer is empty.
1526728997	Total duration of downlink data transmission in a cell.

Table 6: Available counters.

of the KPI would be:

$$\text{Total Outage time per Site} = counter(1526727379) * counter(1526737823) \quad (8)$$

Which means that the *Maximum number of user in a cell* times the *Duration of eNodeB Unavailability* gives us a cumulative measure of the disservice caused to the subscribers that were covered by that cell in those 15 minutes. Remember that the granularity of the counters' measurement is 15 minutes.

However, since this straight forward approach had to be excluded due to lack of data, I moved on with an automatic approach to extract relevant counters from the ones available. This system will be presented, together with its challenges in Paragraph 10.4.

10.4 Automatic model for counters extraction

Since the approach we are researching must work in most of the scenarios, if not all, the definition of a KPI with unavailable data in 10.3 did not meet our requirements. For this reason, I explored machine learning based (ML) algorithms to tackle this problem. The goal is to understand what are the counters that are most affected by an alarm and at a second stage, trying to use those counters to estimate the alarm impact. The use of ML algorithms, in this case, is useful to understand what are the most important counters that allow the algorithm to predict an alarm.

Counter ID	Description
1526727378	Average number of users in a cell.
1526727379	Maximum number of users in a cell.
1526728333	Average number of UL synchronized users in a cell.
1526728426	Average number of downlink CA UEs that use the local cell as their primary serving cells.
1526728427	Average number of downlink CA UEs that use the local cell as their secondary serving cells.
1526728516	Maximum number of downlink CA UEs that use the local cell as their primary serving cells.
1526737823	Duration of eNodeB unavailability.

Table 7: Non Available counters.

To rephrase what was stated above, you can imagine it as an attempt to shift from a static KPI, as the one defined in 10.3, to a dynamic KPI that, based on the measurements available, adapts its definition to provide a reliable mean to monitor the impact.
Here we slightly transcend from the definition of KPI (Paragraph 2.7), as a precise and measurable quantity that leads to actions, however, our goal is also slightly different. We need a mean of comparison between automation and manual, not necessarily a number. From this preamble, the birth of the KCI concept, described in 10.11.

10.5 Definition of the problem and Model choice

The problem requires an automatic way to understand the predictive importance of the counters. The prediction problem can fit into the classification problems. To put it simply, we are looking for a model to predict whether an alarm is happening or not, based on the value of multiple counters. This does not ultimately aim to have a system to predict alarms, but aims to find a reliable mechanism to extract the counters through which we can monitor the impact of alarms.

A simple way to extract the feature importance from our classification problem is to use a decision tree algorithm. However, a single tree is usually not enough to produce satisfactory results, it can works very well for the training set but it does not have a great capacity of generalizing once new data

are fed into it. For this reason, I decided to use the Random Forest model, which can achieve higher accuracy in the prediction. Random Forest is a machine learning technique that combines several base models (trees, in this case) to produce an optimal predictive model. This technique belongs to the Ensemble methods [34]. Compared to decision trees, with random forest we lose interpretability, but considering the vastness of the dataset, especially in terms of counters (features) it would be anyway useless to manually follow the decision thresholds of the tree.

10.6 Training Dataset

The work done in 10.1 and 10.2 was a good starting point to prepare the dataset to train the random forest model. In this paragraph I will explain further the next steps needed to prepare the dataset used.

An extract from the raw data is reported as example. You can see that counter 50342575 for *site X* was measured and saved correctly by the system.

	site X
time	50342575
00:00	2.0
00:15	1.5
00:30	0.0
00:45	1.0
01:00	0.5
01:15	1.0
...	...
22:30	8.0
22:45	6.5
23:00	9.0
23:15	10.0
23:30	2.5
23:45	2.0

10.6.1 Manual Labeling

In machine learning, data labeling is the process of identifying raw data (images, text files, videos, etc.) and adding one or more meaningful and

informative labels to provide context so that a machine learning model can learn from it [35]. To have a complete dataset to use in the training and test phases I was still missing the labels. Since the Random Forest model is a supervised method [36], for the training dataset we also need to have the actual outcome. This means that the PM data and the alarms must be condensed in one single dataset, where at any measurement point we have a field saying whether an alarm has been reported or not from the network site. Note that differently from the PM counters, the alarms are not time-series, i.e. alarms occur at any point of time and "1", in the following snippet, means that there has been an alarm in the following 15 minutes.

An example of the labels data follows:

time	29240	29213	26204	301
00:00	0	0	0	0
00:15	0	0	0	0
00:30	0	0	0	0
00:45	0	0	0	0
01:00	0	0	0	0
...	...	...	...	...
22:15	0	1	0	0
22:30	1	0	0	0
22:45	0	0	0	0
23:00	0	0	0	0
23:15	0	0	0	0
23:30	0	0	0	0
23:45	0	0	0	0

This process of manually creating the labels is very time consuming and due to time limitation, I was not able to extend this process to more than five sites. This led to another problem that I will discuss later in Paragraph 10.6.3.

10.6.2 Normalization

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to use a common scale, without distorting differences

in the ranges of values or losing information. Normalization is also required for some algorithms to model the data correctly [37]. The same counter for *site X* in 10.6 is reported after the normalization process.

	site X
time	50342575
00:00	0.0416667
00:15	0.03125
00:30	0
00:45	0.0208333
01:00	0.0104167
01:15	0.0208333
...	...
22:30	0.166667
22:45	0.135417
23:00	0.1875
23:15	0.208333
23:30	0.0520833
23:45	0.0416667

Moreover, the scale of the counters needed to be normalized as well for representation purposes. The orders of magnitude of the available counters are very irregular, ranging from values in the order of decimals to orders of hundreds and thousands. In the case of graphical representation of these values without normalization, a comparison would have been impossible.

10.6.3 Dataset Extension

The manual labeling of the data requires time and due to time constraints it was not possible to label enough data to have a proper dataset, in the order of a few hundreds data points at least. The data available was from a single day and at the end of the labeling process, the dataset was counting 96 data points. Moreover, there was an important problem of unbalanced data. The amount of data points reporting an alarm was around one or two per site, meaning that among 96 data points, around 1-2% was representing alarms. Training the ML model with that little amount of data is not enough as seen from an initial experiment.

To eliminate this problem, the easiest way is labeling more days of data. Unfortunately, due to limitation in the availability of data and time constraints, this possibility was excluded.

To solve this problem, we decided to use the data available from the different sites and create a single dataset. To do that, I extracted the available counters common to all the five sites and merged them to create a dataset with 480 data points. The data was still unbalanced and this fact was taken into account in the implementation of the Random Forest model.

10.7 Experiment

The experiment execution has been straight forward thanks to the preprocessing described in the previous paragraphs that ensured a smooth execution. In a random forest model, multiple parameters can be set to take into account the nature of the data under analysis. To judge the performance of the model, I used the AUC ROC score [38], which tells us what is the capacity of the model of distinguishing between alarm and no alarm. In this setup, to maximize the AUC ROC score, I played with the criterion to measure the quality of the split. Common criteria are:

- Gini index, to measure the impurity.
- Entropy, to measure the information gain.

The criterion measures the accuracy of the classification done by the model and it is used during the creation of the trees.

The highest AUC score was achieved with the following setup:

- n_estimators=80; the number of trees in the forest.
- criterion='entropy'; the function to measure the quality of a split.
- class_weight='balanced'; weights associated with classes, 'balanced' solves the unbalanced data problem.
- max_features='sqrt'; the number of features to consider when looking for the best split.

For further information about the model and the possible parameters, please refer to the documentation of the Scikit-Learn tool used in the code in [39].

10.8 Results and New Counters

With the settings described in 10.7 and the labeled data for alarms 29240, the scores achieved by the models are reported in Table 8.

	Mean ROC AUC
Experiment 1 (Entropy)	0.732
Experiment 2 (Gini Index)	0.713

Table 8: Experiment results.

In model 1, the entropy criterion was used to define the splits while in model 2, the gini index was used for the same purpose. The Gini index is most common in these scenarios, however in this case the model produced better results with the entropy.
In Figures 10 and 11, the respective ten most important counters for Experiment 1 and Experiment 2, as best predictors of alarms, are reported. As you can see there are multiple counters in common, those are highlighted in yellow. Among the common counters, only two were in a different order and they are displayed in an orange box.

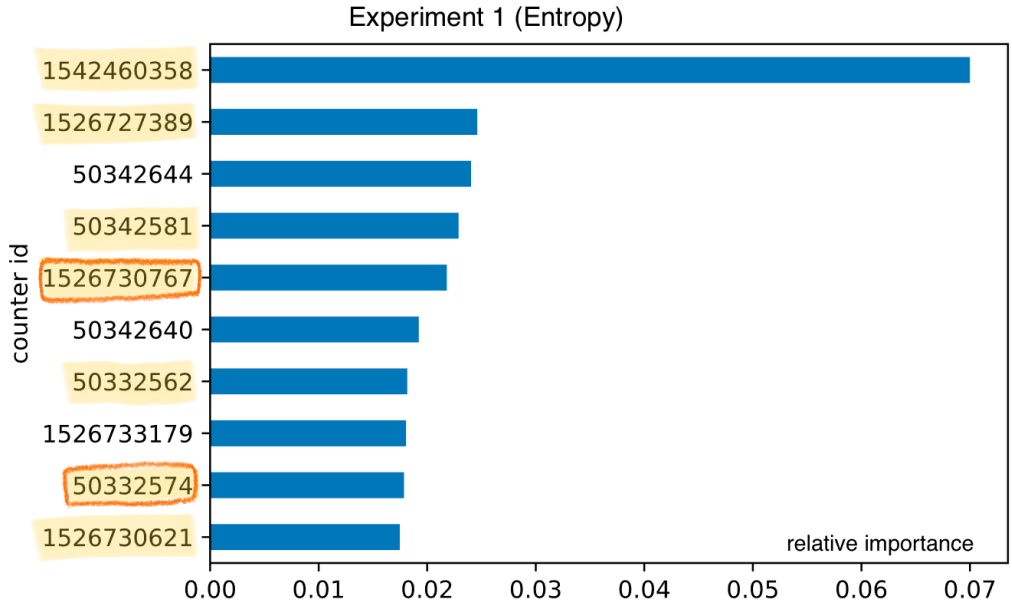

Figure 10: Top 10 counters from experiment 1.

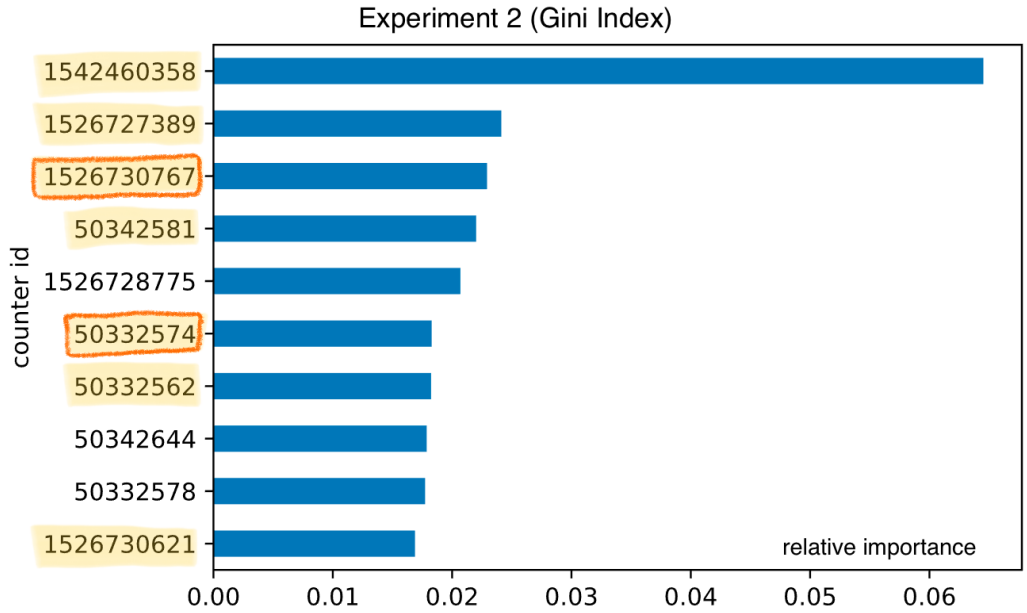

Figure 11: Top 10 counters from experiment 2.

In Table 9 the description of the top ten counters for the two models is reported. The counter are ranked according to the overall highest importance score in the two models. Interesting to mention is that all counters belong to the family *Measurement of Cell Performance*. Moreover, grouping the extracted counters based on the type, e.g. traffic level, setup requests, interference/noise, may be beneficial for future work.

The same analysis for the remaining alarms, namely 29213, 26235, 26204 and 301, is left for future work for various reasons, including time limitations and lack of data. Alarm 29240 has been the most suitable thanks to the frequency of the events across different sites that helped to build the dataset.

10.9 Conclusions

This experiment has shown some potential. Due to multiple constraints, including time and data, the results have been only partially conclusive. The most important part of this work, which includes Section 9 and 10, has probably been the exclusion of some approaches, mostly dictated by inconclusive results and unreliable data. However, I consider the findings collected here

Counter ID	Importance	Description
1542460358	0.0699	Number of IP packets received on the SCTP link.
1526727389	0.0246	Average downlink transmit power in a cell.
50342644	0.0240	Maximum user-plane and control-plane transmit rate at the IP layer over the Iub interface.
50342581	0.0229	The max number of Dedicated Measurement Reporting per second.
1526730767	0.0218	Maximum interference and noise received by PRB 47 in a cell.
1526728775	0.0207	Maximum RRC connection setup duration.
50342640	0.0192	Number of user-plane and control-plane IP packets transmitted at the IP layer over the Iub interface.
50332562	0.0182	Average number of DL CEs consumed in a shared group.
1526733179	0.0180	Number of the control-plane and user-plane packets received at the IP layer over the eNodeB transmission interfaces.
50342644	0.0179	Maximum user-plane and control-plane transmit rate at the IP layer over the Iub interface.
50332574	0.0178	Mean Number of Uplink CEs Consumed by All Cells in an Uplink Resource Group.
50332578	0.0177	The average number of RL Setup Request per second.
1526730621	0.0175	Average interference and noise received by PRB 1 in a cell.

Table 9: Top counters from Experiment 1 and 2.

as a good starting point for the next chapter of this investigation.

10.10 Future Improvements

This work highlights the main issues with the current data and defines the main steps to prepare the dataset to be used with the Random Forest model. This experiment should be carried out again with data from different networks. If possible, the data collection should be done after all the PM counters have been activated for all the network elements in the network management system. Moreover, by using months of observation, rather than a single day, it will be possible to have a more comprehensive view of the alarms and

the counters. This information will be reflected in the model and both the accuracy and the quality of the prediction to extract the counters will increase. I also encourage exploring more advanced ML algorithms or a more fine-tuned version of the random forest based on the data. This approach may also lead to different use cases for the prediction of alarms.

Furthermore, other suggestions are worth considering in the future for data usage. Techniques such as the PCA (Principal Component Analysis) can be used in cooperation with feature engineering with an expert's opinion to reduce the dimensionality of the data and see if such technique can produce some abstract indicators to use directly to measure the impact. However, the downside of this technique is to lose explainability. It will not be possible to extract the original counters anymore but only new indexes that represent the aggregation of multiple counters.
One other suggestion is to consider the time series of the PM counters. With the time series, it is possible to remove the components of the general trends and focus on the part that is more related to the effect of the alarm. Also, the alarm clear information can be included and monitor the changes of the counters' time-series.

This approach can be then extended to other equipment vendors, Ericsson and Nokia for instance. One of the underlying requirements has always been *vendor agnostic* and it is important to test the approach with all the common vendors to understand if the solution can be generalized or requires changes.

10.11 Key Comparison Indicator (KCI)

In Paragraph 10.4, I introduced the idea of a dynamic indicator to monitor the impact. From that idea came the concept of Key Comparison Indicator, to replace a normal KPI in an unreliable situation where the data is not always available. It detaches from the classical definition of performance indicator as a measurable number and embraces the inaccuracy that comes from a partial measure.

Figure 12 reports a visualization of this concept for clarity. It consists of a range of values, as visible on the right, divided into little intervals, where each interval corresponds to a performance state of the system.

With the isolated counters in Table 9 and from a deeper analysis with more data mentioned in 10.10, the next step would be to create a model that based

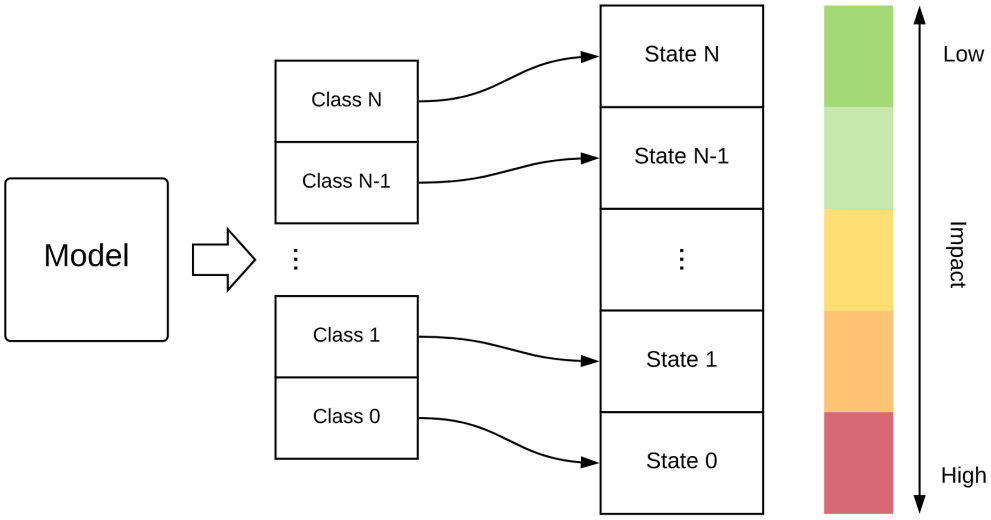

Figure 12: Impact Key Comparison Indicator.

on the values of the counters can predict a class, where a class is the output category of your data, e.g. class 0 there are no alarms, class 1 there is an alarm, class 2 there are major alarms. That class can then be matched to the state of the system, as visible on the left part of Figure 12.

Regarding the nature of the model, there are some limitations in using another ML algorithm. If we assume that the input features (counters) are not constant, we cannot train a model and expect to perform well when the key counters are missing. Thus, more investigation on this is required. No concrete result for this model has been produced during this work.

This indicator would help to create statistics that evaluate how well the network is working and how much the automation is contributing to maintaining good network performance. By saying that during a reference day, the network was e.g. 70% of the time in the green state while without automation the network was e.g. 60% of the time in the green state, we could finally measure the action of automation.

11 Network Performance and Revenue

This conclusive section expands on the initial intuition that led me to the work done in Section 9 and 10. It provides a panoramic on the literature that shows how network performance reflects on revenue. This section paved the way to some intuitions and helped to narrow down the broad topic of *Proving the business case of Virtual NOC*.

11.1 Introduction

The revenue of a mobile operator may come from a considerable number of activities and the number of factors that influence it is even higher. Just think of entertainment services, known as OTT (over-the-top) services; equipment sales such as mobile devices, computers, or TVs; or more recently home automation and industrial IoT [40]. All these revenue streams run in parallel with providing network connectivity. Moreover, the enormous growth of data volume has not led to corresponding revenue growth, making the telecom business less profitable than 10 years ago [40]. Following these considerations, Elisa Automate aims to reduce the cost of provisioning network connectivity, maintaining profitable providing communication services. It targets profitability through its core product, named Virtual NOC, which aims to improve Network Performance, alongside reducing OPEX with automation. The existence of other factors influencing the operators' revenue is acknowledged but it is not relevant to the scope of this work.

Although network performance and revenue are not directly related, the former has an indirect impact on the latter. Through several steps, user-perceived performance will eventually affect the bottom line of the business, and to capitalize on this, the operator must have long-term plans. Delivering a better network experience can have an impact between one and two EBITDA percentage points [40].

Note that in this context the words *user*, *customer*, and *subscriber* refer to the same person which is the person that utilizes the mobile service and often pays for it, thus, they will be used interchangeably.

11.2 Link between Network Performance and Revenue

This paragraph explains the relationships shown in Figure 13, which are supported by an extensive literature that originated more than thirty years ago [41] [42]. In Figure 13, a qualitative description of the existing relationship between *Availability*, *Quality of Service*, *Customer Satisfaction*, *Corporate Reputation*, and *Operator's Revenue* is presented. The relationship between each block in Figure 13 will be explained in the following sections.

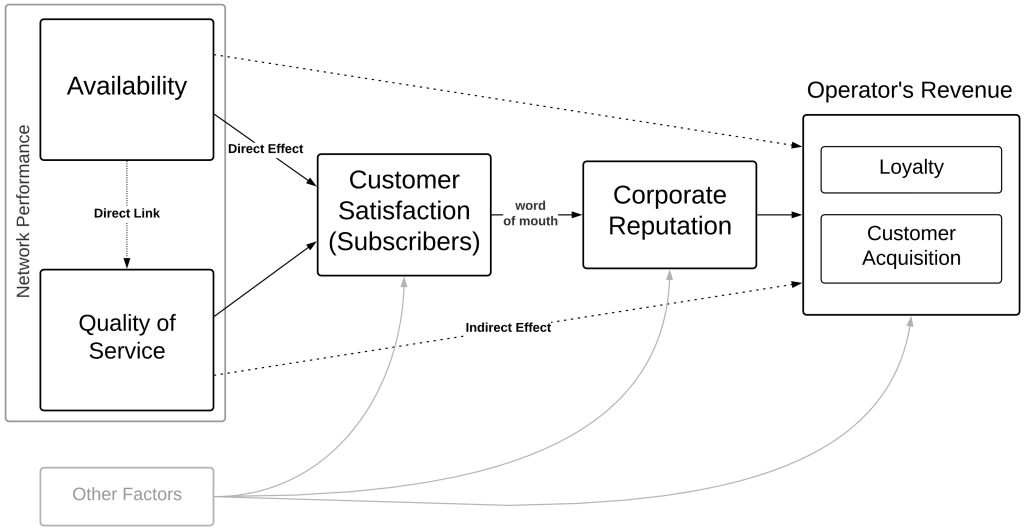

Figure 13: Link between Network Performance and Revenue.

Moreover, the paragraph wants to clarify how Network Performance is measured in terms of availability and quality of service. Network Performance is the aggregation of availability and quality of service (QoS). In general, the user experiences good performance when both availability and QoS are high. To have high QoS, availability is a prerequisite as shown by the direct link. In practical cases, there are infinite combinations of availability and QoS, thus the user-perceived network performance is not in a black and white area but more in the gray area.

11.3 The role of Customer Satisfaction

Despite the intuitive inverse correlation between degraded network performance or downtime and customer satisfaction, in this section, I want to

support the logic with relevant literature as proof. In particular, the following paragraph explains the direct effect that network performance has on customer satisfaction, as visible in Figure 13.

11.3.1 How Customer Satisfaction is formed

As stated in the literature, when the perceived performance exceeds or matches the expectations, the customer usually becomes satisfied [43]. For large companies, the above concept and the continuous research for broad acceptance by their customers has fostered the work of many researchers. In fact, several studies, including [44] and [45], have shown a concrete correlation between network performance and customer satisfaction, i.e. the quality of the service has a positive impact on customer satisfaction and future loyalty. Other factors have a significant impact on satisfaction, including customer service and assistance, but they are out of the scope of this work and only acknowledged.

To maintain customer satisfaction the firm is expected to provide high service quality standards [46]. Especially for high ARPU (Average Revenue Per Unit) customers, who have higher quality expectations. Moreover, among several factors such as customer service, untrained agents, and inferior self-care options, network unreliability is the most influential factor on customer disappointment. Network unreliability is the result of poor quality of service and low availability. From a survey conducted by TechSee in 2019, around 64% of customers who left their operator did it due to poor level of service experienced over a while. Dissatisfaction then leads to 79% of people sharing their negative experience with others, and in the worst case, 22% also share it on social media where the effect is amplified, as reported on the survey [47]. Thus, performing consistently well helps to be trustworthy and inflates business profits and competitiveness [48]. Figure 13 shows that the path to arrive at revenue is through customer satisfaction.

11.4 Corporate Reputation

Since the '90s, reputation has been a metric with high interest for companies [49]. It represents a soft asset that is hard to imitate and can ultimately lead to superior profits [50]. According to the Sector Reputation Analysis [51] conducted by Brand Finance in 2020, telco operators register the lowest reputation score across multiple industries, with only a few exceptions. On average, the score stood at 6/10, with few cases able to arrive at 7.8/10. In

many industries, we have seen how new players can triumph by investing in those activities that otherwise make the reputation plunging, disrupting the industry. Hence the telco sector is still vulnerable to disruption [51]. This situation is especially an advantage for the first operators capable of improving customer satisfaction, Figure 13.

11.4.1 Definition of Reputation

The exact definition varies from study to study. However, in the context of this work, what best resemble and describe this concept is given by Gotsi and Wilson (2001) in [52]:
A corporate reputation is a stakeholder's overall evaluation of a company over time. This evaluation is based on the stakeholder's direct experience with the company, any other form of communication, and symbolism that provides information about the firm's actions and/or a comparison with the actions of other leading rivals.

11.4.2 Importance of Reputation

As stated before, studies on reputation have been carried out since the past century [41]. However, most of them focused on manufacturing companies. This is because service-based companies were nonexistent in that period. With the rise of the Internet and mobile communications, internet service companies started to rise in parallel and reputation became even more important. In services where the physical evidence is rarely possible to evaluate, potential customers attribute very high importance to the reputation of a digital service [53] [54]. Due to the intrinsic intangibility of the services offered by mobile operators, the implications of reputation are higher compared to other industries [55].

11.4.3 Reputation and word of mouth

Customer satisfaction and trust directly impact corporate reputation as shown by recent studies focused on the implication of reputation [56]. On top of other things, reputation has direct consequences on word of mouth. Word of mouth is still considered as one of the most powerful marketing strategies to reach organic growth [57], which simply means that the company can expand its customer base and increase sales exclusively by its own resources. In other words, the company uses its resources primarily to increase customers'

satisfaction and trust. This will eventually lead to a word of mouth increase which is related to reputation and as mentioned before in Section 11.4, it leads to easier customer acquisition and higher profits.

In recent years, word of mouth has been used as a marketing strategy through advertising campaigns on social media. Several studies, including [58], showed how offline word of mouth is amplified by online word of mouth on social media and vice versa. This helps positively at the condition that the operator can provide a quality experience for customers, otherwise, it will fall short along the way [59].

11.4.4 Managerial Relevance and Limitations

In line with the above, an operator should focus more on increasing its reputation rather than decreasing its churn rate, i.e., focusing on one of the causes (reputation) rather than the effect (churn rate). The churn rate itself usually attracts high interest from mobile operators but as explained in Section 2.5, it is mainly influenced by external factors that are not under the operator's control. Thus, focusing the resources on improving its reputation, which is under the operator's control, can lead to more tangible results, including trust, word of mouth, and customer loyalty [60]. Although there has been great interest in this matter for the past 30 years, it is important to mention that the review of the literature shows a lack of consensus about the optimal framework or approach to measure reputation. This is due to the intangible characteristics of this metric.

Proving the optimal approach to measure reputation is out of the scope of this work but as stated by the studies cited above, there is a clear link between reputation and revenue.

To conclude, as mentioned in Section 3.2, zero-touch network automation is the way to win the current evolution of networks, and fault management automation helps precisely in that direction. Virtual NOC improves the operator's reputation by ensuring network performance.

12 Excluded approach

In this section, a summary of inconclusive work done is reported with the main steps taken. This part is intended to keep track of the ideas born along the way and to explain why it has not been possible to follow that direction. However, this information can be kept as a cue for future work once the conditions or the data required will be available.

12.1 Churn rate prediction with automation statistics

Different techniques are adopted in the literature to estimate and predict the churn rate. These activities have an important value for companies to find an accurate model for customer churn prediction, for the identification of churning factors, and to develop strategies to minimize the rate and increase retention. Major techniques use machine learning or hybrid approaches and use a mix of synthetic and real-world data sets to create and validates the models, including data set such as Call Detail Records as used in [61]. It is important to remember that churn rate is a general term and is relevant to multiple fields and sectors. Thus, non-relevant information to the Telecom sector is used in a large number of works [12]. In this context, the focus has been narrowed down to the telecommunication sector. During the pre-analysis phase of this work, the idea of predicting the churn rate using the metrics coming from Virtual NOC emerged for various reasons. Most importantly, the churn rate is a well-recognized metric across multiple industries, and finding a working model would have been a great advantage for the company. As mentioned in section 11.3, service quality and network performance impact users' satisfaction, and unhappy users tend to churn. On the other hand, happy users develop loyalty and the churn rate will eventually be positively affected, but how much will it be?
Unfortunately, linking the work done by Virtual NOC, or in general, the work done by an automation system to churn rate is not immediate and I have not found studies on this theoretical tendency. Moreover, due to different challenges, most of the data necessary to carry out this idea was missing. Thus, this interesting idea is left for future work once the necessary data are available.

Since the churn rate is influenced by a vast number of factors, including competitor price strategy which is out of the operator's hand, it is difficult

to say that this approach would give sufficient results to prove the value of Virtual NOC, as explained in Section 11.4.4. However, from a high level, to be possible to construct a model based on proven machine learning algorithms to predict churn rate as in [62], it would be necessary to have data about the *Network Actions* done by Virtual NOC to ensure the continuity of the network and the evolution of the churn rate of the operator that implements the automation solution. This information would be used in addition to the data already used for similar models and they should come from the operator itself.

13 General Conclusion

This work aims to advance the visibility of the improvements that come with an automation system, in this context Virtual NOC. Measuring the improvements that come with automation is a challenge that most companies face and here I collect the main findings of this work.

With the first automatic system (Section 7 and 8) I showed how the spared manual work can be continuously estimated and utilized in different situations. First of all, it can be used during proof of concept phases with new customers. Secondly, it can be used as a new feature for the production system to allow the customer to have a measurable quantity of how much time has been moved from repetitive work to more meaningful tasks. In this part, there have been minimal challenges and with little extra work, the improvements proposed in Paragraph 8.5 can be implemented to reach an initial maturity of the feature.

The second automatic system (Section 9, 10, and 11) shows a promising way to address the difficult challenge of measuring the change in the impact of end-user affecting alarms. In this part, many challenges were found and more research is needed to see if the proposed approach is viable. Lack of data was one of the main problems at this stage. To continue the work is fundamental to prepare in advance the data. Furthermore, further reflection is also needed to understand how to combine the output of the proposed system to show a useful impact indicator.

Once these two systems are fully developed and fine-tuned, together they will enable monitoring of the direct and indirect consequences of automation - manual labor saved and increased network performance, respectively. The final goal will be to facilitate the understanding of why it is important to adopt automation systems in telecommunications networks and to confirm benefits of implementing such system to the organizations that instituted it.

References

[1] A. Nandan and P. Liu. Csp hyperautomation insight: How to differentiate a go-to-market strategy. 2021. Retrieved from `https://gartner.com`, on 14.05.2021.

[2] P. Tavares and H. Pinto. The age of telecom network automation, 2021. Retrieved from `https://bit.ly/3rNlkdR`, on 15.03.2021.

[3] E. Fersman. Zero-touch is coming. 2019. Retrieved from `https://tinyurl.com/yjtaevcp`, on 15.04.2021.

[4] Wikipedia. Timeline of the covid-19 pandemic. 2021. Retrieved from `https://tinyurl.com/yd8byk29`, on 07.06.2021.

[5] ThousandEyes and Cisco. What is network operations & best practices. 2021. Retrieved from `https://tinyurl.com/y575zkw5`, on 07.06.2021.

[6] Statista. Market share of telecommunication providers in finland, 2019. Retrieved from `https://bit.ly/3qRcTNg`, on 15.03.2021.

[7] Elisa Automate. Automate your mobile network: Automating engineering and network operation ran processes, 2021. Retrieved from `https://elisaautomate.com/`, on 11.03.2021.

[8] Investopedia. The difference between an operating expense vs. a capital expense, 2020. Retrieved from `https://bit.ly/3w0Vd5H`, on 26.03.2021.

[9] M. Hussein. Telecom investment strategies, capex or opex?, 2017. Retrieved from `https://tinyurl.com/ygr2aunx`, on 26.03.2021.

[10] Vero. Technical interface, 2018. Retrieved from `https://bit.ly/3co5Mb0`, on 25.03.2021.

[11] The Open Group. Business layer, 2019. Retrieved from `https://tinyurl.com/yfgvm888`, on 25.03.2021.

[12] Baris Karaman. Churn prediction, 2019. Retrieved from `https://tinyurl.com/yg977w32`, on 12.03.2021.

[13] Klipfolio. Key performance indicator (kpi) definition, 2020. Retrieved from `https://tinyurl.com/yb4d3k6m`, on 03.03.2021.

[14] Wikipedia. The smart criteria. 2013. Retrieved from https://tinyurl.com/ay6b7z8, on 04.08.2021.

[15] Wikipedia. Autonomic networking. Retrieved from https://tinyurl.com/yfmytd3p, on 15.04.2021.

[16] Ranganai Chaparadza. UniFAFF: A unifi ed framework for implementing autonomic fault management and failure detection for self-managing networks. 2008.

[17] S. Aidarous and T. Plevyak. *Fault Management*, pages 268–301. 1994.

[18] Ma łgorzata Steinder and Adarshpal S Sethi. A survey of fault localization techniques in computer networks. *Science of computer programming*, 53(2):165–194, 2004.

[19] Comarch. Predictive maintenance as the first step towards intelligent assurance. *White Paper*, 2019.

[20] Cynthia S Hood and Member Ieee. Proactive Network Fault Detection. 46(3):333–341, 1997.

[21] Wikipedia. Network operation centre. Retrieved from https://tinyurl.com/qc7kkle, on 16.04.2021.

[22] J. Yue. Design and implementation of telecom user business management system. In *2018 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)*, pages 484–487, 2018.

[23] Wikipedia. Gartner, 2021. Retrieved from https://en.wikipedia.org/wiki/Gartner, on 14.05.2021.

[24] SHRM. How do i calculate full-time equivalent (fte) hours? 2021. Retrieved from https://tinyurl.com/y5jbk494, on 15.06.2021.

[25] Wikipedia. Carrier aggregation. 2017. Retrieved from https://tinyurl.com/yheg5p9c, on 19.07.2021.

[26] Huawei. Product support. 2021. Retrieved from https://tinyurl.com/yema2w93, on 10.06.2021.

[27] Huawei. Setting the alarm severity, 2021. Retrieved from https://tinyurl.com/ygqav3z4, on 04.05.2021.

[28] Huawei. Carrier network support. Retrieved from https://tinyurl.com/yzj8ryns, on 18.07.2021.

[29] Huawei. Carrier network support. Retrieved from https://tinyurl.com/yz6m8a8k, on 18.07.2021.

[30] Huawei. Carrier network support. Retrieved from https://tinyurl.com/yjdmbals, on 18.07.2021.

[31] Huawei. Carrier network support. Retrieved from https://tinyurl.com/yjfrof6z, on 18.07.2021.

[32] Huawei. U2020 alarms reference. page 84, 2019.

[33] Huawei. Kpi & counter family. 2021. Retrieved from https://tinyurl.com/ydlr9cgx, on 24.05.2021.

[34] Evan Lutins. Ensemble methods in machine learning. 2017. Retrieved from https://tinyurl.com/yf8w6rnj, on 29.07.2021.

[35] AWS. Data labeling. 2020. Retrieved from https://tinyurl.com/yhftbo9p, on 05.08.2021.

[36] Wikipedia. Random forest classifier. 2021. Retrieved from https://tinyurl.com/jqkuzmu, on 15.07.2021.

[37] Microsoft. Normalize data. 2021. Retrieved from https://tinyurl.com/yj47hrj5, on 23.07.2021.

[38] S. Narkhede. Understanding auc - roc curve. 2018. Retrieved from https://tinyurl.com/y8d8l9uz, on 26.07.2021.

[39] scikit learn. Random forest classifier. 2020. Retrieved from https://tinyurl.com/yyd4kqbu, on 26.07.2021.

[40] O. Wyman. Telco2025: Telcos need new revenue streams. Retrieved from https://tinyurl.com/yfku236w, on 20.04.2021.

[41] Keith Weigelt and Colin Camerer. Reputation and corporate strategy: A review of recent theory and applications. *Strategic management journal*, 9(5):443–454, 1988.

[42] Paul Herbig and John Milewicz. The relationship of reputation and credibility to brand success. *Journal of consumer marketing*, 12(4):5–11, 1995.

[43] Rajiv D Banker, Gordon Potter, and Dhinu Srinivasan. An empirical investigation of an incentive plan that includes nonfinancial performance measures. *The accounting review*, 75(1):65–92, 2000.

[44] Md Nekmahmud Argon and Shafiqur Rahman. *Measuring the Competitiveness Factors in Telecommunication Markets*, pages 339–372. 05 2018.

[45] Rakshit Negi. Determining customer satisfaction through perceived service quality: A study of ethiopian mobile users. *International journal of mobile marketing*, 4(1), 2009.

[46] Gloria K.Q Agyapong. The Effect of Service Quality on Customer Satisfaction in the Utility Industry – A Case of Vodafone (Ghana). *International Journal of Business and Management*, 6(5):203–210, 2011.

[47] TechSee. Reasons for customer churn in the telecom industry. 2019. Retrieved from `https://bit.ly/3wZY8fh`, on 15.04.2021.

[48] Ying-Feng Kuo, Chi-Ming Wu, and Wei-Jaw Deng. The relationships among service quality, perceived value, customer satisfaction, and post-purchase intention in mobile value-added services. *Computers in human behavior*, 25(4):887–896, 2009.

[49] E. Yoon, J. Guffey, and V. Kijewski. The effects of information and company reputation on intentions to buy a business service. *Journal of Business Research*, 27:215–228, 1993.

[50] N. A. Gardberg and C. J. Fombrun. The global reputation quotient project: first steps towards a cross- nationally valid measure of corporate reputation. *Corporate Reputation Review*, 4:303–307, 2002.

[51] Brand Finance. Telecoms 150: the annual report on the most valuable and strongest telecom brands, 2020. Retrieved from `https://bit.ly/31toLLd`, on 29.03.2021.

[52] M. Gotsi and A. A. Wilson. Corporate reputation: seeking a definition. *Corporate Communications*, 6:24–30, 2001.

[53] D. B. Bromley. Psychological aspects of corporate identity, image and reputation. *Corporate Reputation Review*, 3:240–252, 2000.

[54] S. Hardaker and C. Fill. Corporate service brands: the intellectual and emotional engagement of employees. *Corporate Reputation Review*, 7:365–376, 2005.

[55] J. B. Kim and C. J. Choi. Reputation and product tampering in service industries. *Service Industries Journal*, 23:3–11, 2003.

[56] Gianfranco Walsh, Vincent Wayne Mitchell, Paul R. Jackson, and Sharon E. Beatty. Examining the antecedents and consequences of corporate reputation: A customer perspective. *British Journal of Management*, 20(2):187–203, 2009.

[57] Kimberly A. Whitler. Why word of mouth marketing is the most important social media, 2014. Retrieved from `https://bit.ly/3rxXz8s`, on 29.03.2021.

[58] M. Hajli. A study of the impact of social media on consumers. *International Journal of Market Research*, 56:387 – 404, 2014.

[59] M. Glover. Word of mouth marketing in 2021, 2021. Retrieved from `https://tinyurl.com/yynulvo3`, on 14.04.2021.

[60] Marko Sarstedt, Petra Wilczynski, and T. C. Melewar. Measuring reputation in global markets-A comparison of reputation measures' convergent and criterion validities. *Journal of World Business*, 48(3):329–339, jul 2013.

[61] I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam, and S. W. Kim. A churn prediction model using random forest: Analysis of machine learning techniques for churn prediction and factor identification in telecom sector. *IEEE Access*, 7:60134–60149, 2019.

[62] B. P. and N. G.S. A review on churn prediction modeling in telecom environment. In *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, pages 1–5, 2017.

END.