Heidi Friman

Developing a Solution for the Prevention of Cloud Infrastructure Related Outages in Telco Cloud

Helsinki Metropolia University of Applied Sciences

Master's Degree

Information Technology

Master's Thesis

30 November 2018

Helsinki Metropolia University of Applied Sciences

Author Title Number of Pages Date	Heidi Friman Developing a Solution for the Prevention of Cloud Infrastructure Related Outages in Telco Cloud 37 pages + 1 appendices 30 November 2018
Degree	Master's degree
Degree Programme	Information Technology
Instructor	Auvo Häkkinen, Principal Lecturer

This Master's thesis is about data analysis of fault report data to come up with a solution for preventing outages in telecommunications cloud products. The business problem that this thesis attempts to solve is how to detect and prevent network outages before they happen instead of correcting the faults after they have happened. It is about proactive care for the customer, but also about cost savings for the telecommunications vendors. The earlier a fault is detected, the more cost-efficient it is to solve. The scope is limited to faults from cloud infrastructure layer of cloud products. Also during the work, it was noticed that developing the whole solution would have been too big of a task in scope of this thesis and thus this thesis handles mainly the data analysis part of earlier detected faults in cloud infrastructure layer.

The work was done by first getting familiar with outages in the telecommunications cloud, second by investigating data analysis theory and methods. By using the CRISP-DM method a data mining project was done on the fault report data gotten from real telecommunications cloud products. WEKA and R were the tools used in the data mining tasks. Then the results were analysed and suggestions for future work given.

The goal of the project was to create a script that would alert the user when symptoms were detected that would signal an outage. The script would have been based on the data mining results on the fault data. The data analysis however brought into attention that the data that was investigated was not good enough and the data set large enough to provide valuable information. This forced the outcome to be re-iterated, and so the outcome was changed from a script to a list of improvement suggestions for the fault reporting tool.

Although the outcome was changed, the study provided valuable information, since it revealed that the data is not very suitable for data mining. The improvements suggested to the tool will make the data much more usable in the future. That is also the intent of the company. The development of the whole solution for outage prevention remains then a work to be done in the future.

Keywords	Telco cloud, virtualized network function, data analytics, CRISP-DM

Contents

Abbreviations

3G	3rd generation. Refers to telecommunications networks technology.
ARFF	Attribute-Relation File Format. The file format used to input data to WEKA tool.
CRISP-DM	Cross-industry standard process for data mining. A process model for data mining.
DSRM	Design Science Research methodology. Common process to conduct research in Information systems.
EDA	Escape Defect Analysis. Analyses why the defect got through to the end-product.
EMS	Element Management System.
ETSI	European Telecommunications Standards Institute. Produces prevalent telecommunications standards.
IBK	A k-nearest neighbors algorithm used in data analysis
IS	Implementation specification
KPI	Key performance indicator. A measurement that tells how well the product is functioning
LTE	Long Term Evolution. An extension of 3G networks in telecommunications. Sometimes also called 4G.
NFV ISG	Network Functions Virtualization Industry Specification Group. A group in charge of designing the virtualization in telecommunications
NFV	Network Functions Virtualization. Architecture of the network functions in virtualized telecommunications networks.
NFVI	Network Functions Virtualization Infrastructure. Part of the NFV architecture where hardware and software resources are virtualized.
RCA	Root Cause Analysis. An analysis to find the real root cause of a problem.
SMO	Sequential Minimal Optimization. A support vector classifier in WEKA.
VNF	Virtualized Network Function. An entity that handles certain functions in a virtualized telecommunications product
WEKA	Waikato Environment for Knowledge Analysis. An freely available data analysis tool.

1 Introduction

Maintenance in telecommunications networks is a big part of the business, consisting of software and hardware upgrades and monitoring of KPIs (key performance indicators), but also of solving service degradations, hardware and software problems and network outages. As information technology is moving rapidly to cloud based solutions also in the telecommunications field, fault management and troubleshooting need to change. The later a fault in the system is detected, the more expensive it is to correct. Problems found at customer live networks require immediate attention from research and development (R&D) teams, thus postponing other development work or sometimes taking engineers physically to other parts of the world for a visit to customer sites to solve problems. Such work is very expensive and wastes valuable resources. Critical faults that happen in live telecommunications networks also affect end-users and can be vital, if the whole network is not accessible. These faults where all or part of the network does not transfer traffic are called network outages. Telecommunications vendors aim to have as few minutes of outages per year as possible. Thus, as the network elements are being moved to cloud infrastructure it is necessary to look at critical problems that might arise from this change and prepare for them.

Customer satisfaction is always severely affected when outages happen, both from operators' and from vendors' point of view, which in turn has a direct effect on business success and financial results. So, it is understandable that operators want and need proactive fault detection, faster problem resolutions and less outages. This also helps the providers by reducing the amount of maintenance work load and by enabling easier troubleshooting. One of the problems has been that all the correct troubleshooting data is not available to solve problems as the fault is detected too late or some log collection has not been active at the time of the occurrence. It would be very beneficial for all parties if the faults were detected fast or even prevented before they happen. This would help the technical support personnel in troubleshooting, and the customers would be happier with shorter resolution times. From their perspective another important view point is that they want to know the status of their network at all times. The operators monitor their networks for outages, but they would like to have even more in-depth view on the status of the network devices. To solve this issue, Nokia has provided a feature that checks the status of a network device in cloud based products.

This study aims at developing a model to detect symptoms of outages before they happen and notifying the operator or customer care personnel of prevailing threat of outage. This way they can react quickly and prevent the outage. This study will not cover outages caused by application software, but focus is on cloud infrastructure and hardware related problems. This study is part of developing an existing Preventive and Troubleshooting Framework. The existing framework supports only traditional network elements currently, but this new model will expand the framework to include virtualized network functions in telecommunication cloud as well. Thus, the research questions for this study are

- What symptoms cause outages in cloud infra?
- How to detect the symptoms from troubleshooting data and alert end user of an outage threat?
- How to visualize the data near real-time in a user-friendly way?
- How to develop a solution that it easily adaptable to other uses and products also?

Outcome of the study is a model on how to prevent cloud infra based outages in telco cloud as well as an evaluated pilot of the model. Pilot is done with one cloud based VNF (virtualized network function), with one cloud infrastructure provider and in laboratory setting. Suggestions for improvements in the model will be done based on the pilot.

This thesis is organized in such a way that in the second chapter about methods and materials, the research method is introduced as well as the scope of the project. Also, outages are briefly analyzed. In the third chapter the environment of the study, telecommunications cloud, is described. Then a short literary review of similar solutions is given, and data analysis concepts are introduced including text analysis. Current state analysis and objectives of the study are the topic of chapter four. There the current troubleshooting framework is described and the requirements for the project are related. Chapter five progresses through the CRISP-DM (cross-industry standard process for data mining) process to develop the solution until the evaluation of the data analysis. In chapter six the results of the project are further assessed and compared to the objectives set at the beginning. Additionally, the outcome is re-iterated. In the final chapter, chapter seven, there is discussion of the outcome and an insight to the future is proposed.

2 Methods and materials

In this chapter, first the project scope is defined. Second, the research method and design is specified. Third, outages in cloud environments are investigated. Finally, the expected outcome of the study is indicated and evaluation method described.

2.1 Project scope

This project is done in the field on telecommunications, from the equipment manufacturer's point of view. In telecommunications area, the scope is narrowed to 3G and LTE network elements, and in there to virtualized network functions (VNF) (figure 3) and services related to telecommunication cloud solutions. As these telecommunication network elements are very complicated products and have many features that could be studied in the context of virtualization and cloud, it is necessary to point out that this study will only concentrate on troubleshooting improvements.

The research is conducted as a proof of concept study mainly because the solution for a troubleshooting data analysis in VNF is needed, but does not exist yet. As this solution is part of a bigger troubleshooting framework, it will not be in commercial use very soon. Thus, some parts of it are left rather vague to keep the confidentiality of the company. Goal is to create a working solution for a specific case of troubleshooting data analysis that can then be generalized to many cloud products in telecommunication cloud and for many other analysis usages also. The solution is developed by exploring the existing solutions in the Preventive and Troubleshooting Framework, studying similar solutions and evaluating other possible approaches, then selecting one for the pilot and evaluating it afterwards.

2.2 Research method and design

As this study is conducted within the discipline of Information technology, a research method is selected by keeping that in mind. Information systems is an applied discipline that is quite much concerned with producing a practical applicable solution to the research question, whereas in other sciences such as social sciences or linguistics, the focus is more on explanatory and descriptive conclusions. Peffers, Tuunanen et al have proposed a Design Science Research Methodology (DSRM) to cater for the needs of

information systems research. Design science concept itself was introduced to information systems already in early 1990s, but it was not generally used in research papers perhaps because there was no common research methodology. DSRM was then proposed to work as a generally accepted framework for information systems research, based on the guidelines provided by Hevner et al. in Design Research in information systems research (2004).

Design science research methodology is selected as the methodology for this study, as it provides a good foundation for conducting the study in a way that is generally understandable and comparable in the related research field. Design Process is depicted below in figure 1.

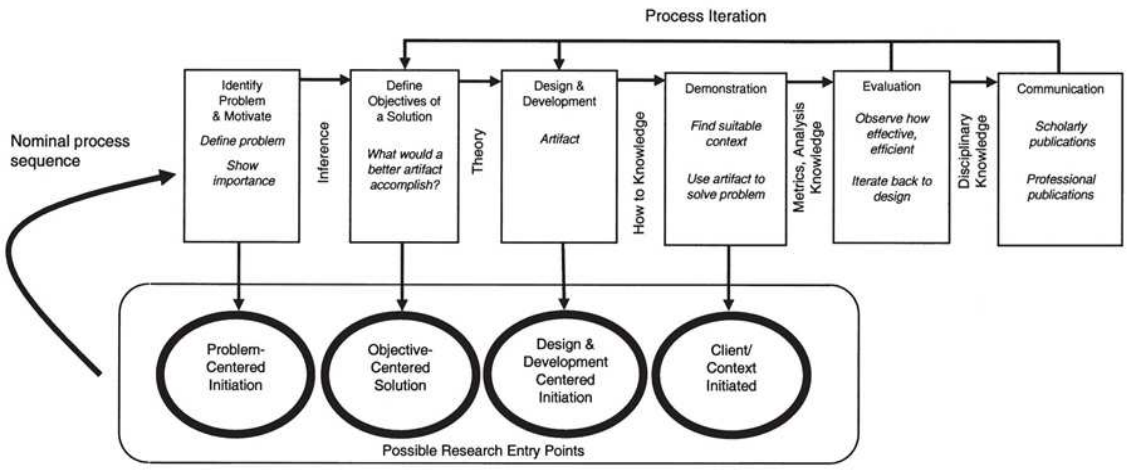

Figure 1: DSRM process model (Peffers et al. 2007)

In this paper the research starts from the problem-centred entry point, as seen in figure 1. First a technological problem is defined and a business problem explained to understand the importance of solving the problem. Then the objectives of the solution are defined as requirements for the outage prevention model as well as for the proof of concept. In this part also, a study of the existing reasons for outages are studied and symptoms derived from them. The design and development phase will include discussion of similar solutions and of other relevant literature in order to select an appropriate implementation for the proof of concept. The environment for the proof of concept will then be built in research and development laboratory and all parts implemented. Finally, a demonstration will be done to stakeholders to show that the model works. Based on the demonstration, an evaluation of the selected methods and tools will be made. The results will also be compared to the requirements defined earlier. Suggestions will be made based

on the results and adaptability of the model discussed. In the end the study, its importance and its quality will be discussed in the context of the research community in information technology.

2.3 Cloud infrastructure outages

In order to prevent an event from happening, symptoms of the event need to be known. When they are known, they can be detected and appropriate action taken. When planning a solution that will prevent outages in telecommunications cloud products, the first problem to solve is to know what the symptoms of the outages are. An overview of online problem reports for cloud infrastructure is done to analyse general outages in cloud environments that might have an effect in cloud infra. Also, problem reports from software development in telco cloud will be analysed. There is already some literature and reviews available on cloud computing outages. Most material is from IT cloud which is a little bit different from telco cloud, but still the infrastructure is the same or similar. It is good to study what has caused service-disruptive outages in public cloud to learn what might be the pitfalls in telco cloud infrastructure also.

Outages are a reality also in public cloud services, but minimizing downtime is a good objective to secure service level agreements. This is even more important in telecommunications cloud where some services are critical, like the ability to make emergency calls. A survey done on outages of the biggest public cloud service providers revealed that all had had outages every year ranging from 1 to 10 per provider. Root cause analysis if the survey results is displayed in figure 2 below.

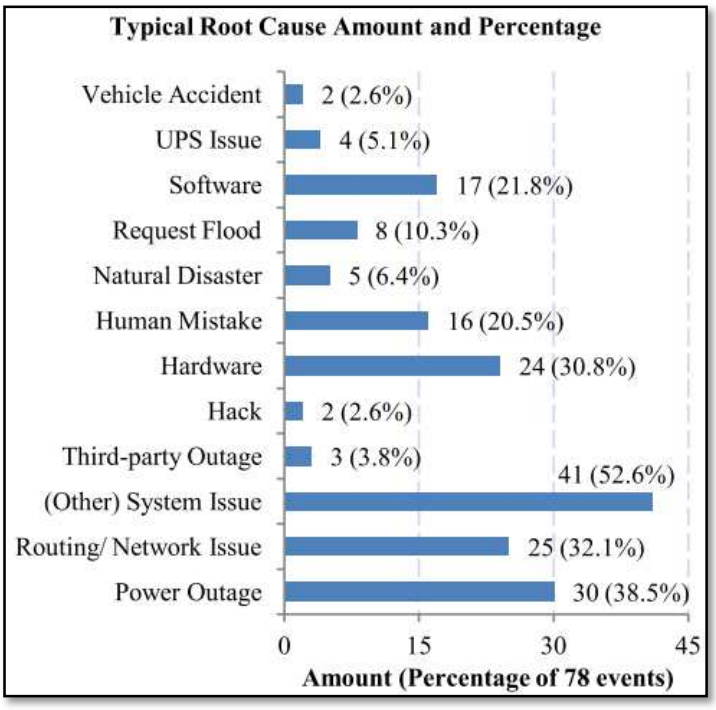

Figure 2: Percentage of root causes to public cloud outages (Li et al. 2013)

Two biggest points of failure are power and network issues according to the picture. These can be usually classified as cloud infrastructure problems. Power outages are difficult to correct and require longer downtimes. Problem is that elasticity management or auto-recovery functions for virtual machines do not work if the whole rack is powerless. Power loss affect every data center at some point of time, so it is important to prepare. Power outages can be caused by many reasons, such as failed power supply related hardware, problems in external power distribution network and failing backup systems. These same problem are reality also on telecommunications data centers and providers should take these into account. (Li et al. 2013).

Thankfully, with the emergence of big data analysis, predictive algorithms can be used to prevent some of the hardware failure before they happen. There are also guidelines on how to build secure data centers to protect them from external threats, like thunder-storms and collisions.

2.4 Outcome and evaluation

In the beginning when the existing framework is studied and other relevant sources also, an understanding will be gotten on whether the existing solution can be used in the VNF environment. Some parts of the solution might be usable as such, but other parts need to be designed again. The existing solution will also be compared with other similar solutions found during the literary review, and a conclusion will be made on whether they are better than the one already in use. Then an approach will be selected to become the proof of concept in this study. This decision will be made by discussing with subject matter experts and by analyzing the requirements. The guidelines to the proof of concept study will be gathered as user stories, and the success of the study will be measured against them.

The proof of concept implementation will be done by integrating the actual hardware and software component together in a research and development laboratory. The solution will be tested and verified with the user stories being used as the guiding documentation. The results of the study will be then evaluated against the user stories and finally discussed with the stakeholders. Decision will be made of whether the study is successful enough or if another approach needs to be tried. Finally, a written report will be made that will include the approach and results as well as evaluation result and recommendations for further use. Evaluation of the solution will the validated with discussions with the key stakeholders. The testing of the solution needs to be planned so, that it also includes the validation of the results.

3 Theory

This thesis is about building a script by analyzing troubleshooting data from cloud infrastructure related problem reports to monitor the status of the VNFs. In this chapter, there is first a brief explanation on what network functions virtualization in telecommunications cloud means. Second, there is a review of similar monitoring solutions to what this thesis hopes to produce, and third a short introduction to predictive data analysis and data mining methods, that will be used to produce a predictive script for outage prevention.

3.1 Cloud infrastructure in telecommunications cloud

Network Functions Virtualization Industry Specification Group (NFV ISG) was established in late 2012, and it is working under European Telecommunications Standards Institute (ETSI). The ISG's purpose is to write specifications for the industry and at the same time making sure that multivendor networks work together and are easily manageable. The architectural framework of network functions virtualization specified by NVF ISG is provided in figure 3 below. In the picture three virtualized network functions (VNFs) are presented in one cloud environment. VNFs are similar to network elements such as radio network controllers, base stations, media gateways in traditional bare metal based telecommunications networks. Here however the hardware layer is detached from the actual application software and the hardware is pooled so that all the related VNFs can utilize its resources.

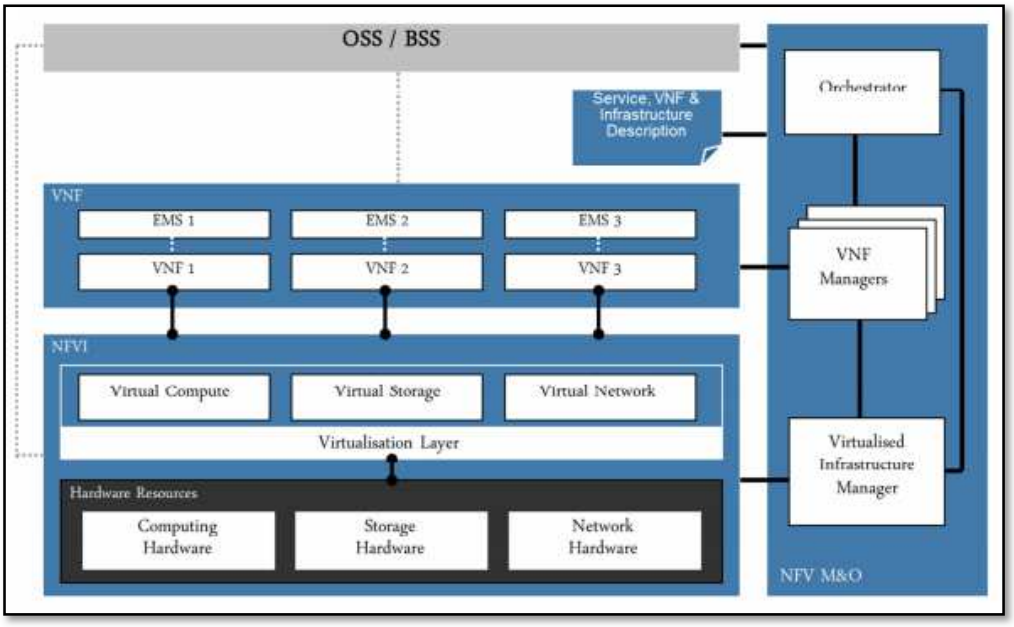

Figure 3: Network Functions Virtualization Architectural Framework (Chiosi et al. 2013)

The architecture is divided to three parts: Network Functions Virtualization Infrastructure (NFVI), Virtualized Network Function (VNF) and NFV Management and Operation. The NFVI part includes hardware and virtualization layers. VNF layer implements software functions that a specific telecommunication network element needs. NFV Management and Operation layer is concerned with the management functions of the other layers and it is the way to connect to the NFV. This thesis concerns only the lowest layer of the

architecture, the Network Functions Virtualization Infrastructure layer, and software faults related to it. These lower layer problems often cause difficult problems in upper layers that are not always easy to drill down to, thus it is important to address these issues early on. In Nokia Airframe solution the hardware is Nokia Airframe Data Center hardware, virtualization layer is either based on Openstack or VMWare, and VNF layer includes the application software.

3.2 Literary review of similar solutions

It is worthwhile to look at the competing offering of monitoring solutions from other telecommunications vendors and from possible open source projects with real-time monitoring capabilities. This is important to understand what the current status in this field is, and to possibly get input for this project. Ericsson (Ericsson, 2015) has a Fault Management solution that is also concentrated on providing faster root cause analysis. Public information on their solution is very limited. They claim that they are able to resolve problems and automatically correct them. The solution can also be integrated to include automatic trouble ticket creation. (Ericsson, 2015). Former Alcatel-Lucent, now part of Nokia, has a well-branded solution called Cloudband which handles all management and operations functions depicted in figure 3. Cloudband includes also cloud infrastructure related troubleshooting features. In this thesis Cloudband solution is ruled out because the author has no access to it.

There are several open source solutions for monitoring networks and systems. Most of them concentrate on tracking the network devices, checking the connection speeds and hardware related monitoring. Some of the solutions offer alarms and notifications on an online graphical user interface. (Venezia, 2014)

3.3 Predictive data analysis

Data mining is a growing practice in the field of information communications technology. Now that more and more data is collected and stored in large datacenters, and there is more computing power as well as connectivity, researchers have invented practices to take this big data into use. Data mining means investigating the data to come up with new knowledge, to learn something useful. Some of the most common uses of data mining are description, estimation, prediction, classification, clustering and association.

These are often used as overlapping methods or they are used in sequence, but the words describe the objective of the data mining task. Here prediction is of most interest for this thesis, however classification and association are also needed to analyze the reasons of the existing problems related to cloud infrastructure outages. (Larose & Larose 2015)

Prediction is the task of mining the data to come up with prediction of future value. This is done by using traditional statistical methods that are also used in data estimation and classification. It is about making rules from the already known data to predict the future values of data.

Classification is a data mining method where data is divided into predefined classes. The instances of data belong to only one class and the model can decide which class the new data belongs to. Some often used classification techniques are decision trees and Bayes' rules. Support vector machines are then a mix of linear regression techniques and classification techniques. (Witten et al. 2017)

Association rule learning is a more complicated concept, where the model can decide any attribute of a new instance, whereas in classification the purpose is to determine only the class. It is about finding dependencies inside the data. (Witten et al. 2017)

3.4 Text data mining

As with all data mining, the purpose of text data mining, or text mining, is to produce new knowledge from existing texts. There has been much debate on what text mining really includes and how it differs from information retrieval, computational linguistics or stylometry (Tonkin & Tourte 2016). Here it is considered as a broad term consisting of text analytics and information retrieval. Computers are much better capable of sorting through millions of lines of text than humans. Problem with text mining and analysis is that computers cannot understand semantic meanings and often written text is human produced. Computers cannot then understand if there is for example same word used in different connotative ways. Text data mining differs from data mining in a sense that it has a large overall data, presented by the whole lexicon of usually one language, and only a small part of it is used in the data set (Aggarwal & Zhai 2012).

The test data set needs to be preprocessed before it can be analyzed. Preprocessing may mean stripping the text of irrelevant content such as advert in a web page, or language detection or segmentation of text to relevant components. In text mining text is handled either as a bag-of-words, meaning every word is independent of the others, or as strings where words exist in relation to one another. Text analysis starts with conversion of text to a form that serves the purposes of the object. In this phase words that have no meaning for the analysis, called stopwords, are removed or tagged and words can be stemmed to their basic form or lemmatized with their corresponding dictionary word (Tonkin & Tourte 2016). Sebastiani (2001) called this text indexing.

Text data mining has many functions that can be used depending on the domain, objective and problems that are investigated. Some of these include information extraction, supervised and unsupervised learning, opinion mining and cross-lingual mining. In this thesis the domain is clear: computer science. This investigation is also not very much concerned with the semantics of the words. In this narrow area of fault reports on cloud infrastructure, the meaning of the words is usually the same, and so semantic analysis is left out. For this text mining task we select the methods of supervised learning such as decision trees, nearest neighbor algorithms and rule-based classification. (Aggarwal & Zhai 2012).

Supervised learning is also called classification, which was described briefly in the previous chapter. It can also be used in text mining. Classification is used for instance in email spam filters, where the classification problem is whether an email is spam or not. Classification always needs a training data set and specific classes that categorize the data. These produce a classification model that can be used to sort the unclassified test data into one of the classes. In the case of email spam filter, a new email going through the model gets labeled to “spam” class or “not-spam” class. In classification, the text is usually treated as bag-of-words and the text need to be converted by stop-word removal and stemming as described before. There are several methods to do this conversion as described in Aggarwal & Zhai: A Survey of Text Classification algorithms (2012).

According to Aggarwal and Zhai (2012) most techniques for data classification have been adapted for use in text analysis. They state that linear classifiers are best suited for text data mining, especially in the case of support-vector machines. This is also suggested by Sebastini in his article Machine Learning in Automated Text Categorization (2001).

Meta-algorithms are also available for text classification that help in making the classification more trustworthy. These meta-algorithms use combining of classifications, changing the algorithms according to a specific need or changing the training data. Using these is recommended to improve the results of the classification model. (Aggarwal & Zhai 2012). None of the classification methods provide 100 percent accuracy on classifying items accurately, although with machine learning the accuracy is much better than manual classification by human experts (Sebastiani 2001).

4 Current state analysis and objectives of the study

In the current state analysis part of this chapter, the framework used for building the solution for outage prevention is described for the reader to better understand the environment where the solution should be integrated into. Then the objectives of the study are defined by explaining the requirements for the final solution as well as requirements for the proof of concept part.

4.1 Preventive and Troubleshooting Framework

Preventive care is part of analytics solutions that Nokia offers to operators. These analytics solutions contain data monetization, end user focused and network focused solutions. Network focused solutions include self-organized networks and preventive services where the networks elements are in focus. Among some other services, preventive care in preventive services uses machine learning to try and prevent problems before they are noticed by end users. This research will produce a small part of the preventive services suite. (Nokia Solutions and Networks, 2016).

The Preventive care services is already running with some customers for their bare metal network elements. It is not yet a sellable item for cloud based VNFs. However, cloud based services have also been developed quite far already, but the solution is missing still this feature that will be developed for this thesis. The solution for outage prevention of a VNF will use the current Preventive and Troubleshooting Framework hardware implementation, data collection methods and data storing methods. However, in the second phase all of these will be transferred to Nokia Airframe cloud infrastructure also. That work will be done as a smaller part of this thesis, as the outcome is expected to be a working version of full cloud implementation of the service.

Since preventive and troubleshooting framework is already deployed to several customers, many of the tools can be reused. Scripts for data collection will be run by the script engine, call data monitoring will be handled with data handler, analysis will be done by analysis engine and visualization by visualization board. Although the tools have been chosen for this project, they will all need new implementation for this solution to work.

4.2 Requirements for outage prevention model

The requirements for the final solution has been gathered from Nokia's global services architects, who have discussed those with the customers. First requirement for the end solution is that all the functionalities should be running on top of Nokia Airframe hardware. As the customers are moving their network elements to data centers, they want all other needed hardware to use cloud technologies also. Second requirement is that the solution should provide a real-time view on the status of the VNFs. Technically however, it is enough to provide current data every five minutes. Some data might not be needed that often, such as configuration data, because it is not likely to change very often. So, the data collection and analysis intervals will be defined better during the implementation and discussed with the key stakeholders.

4.3 Requirements for the pilot case

The wanted outcome of this work, is a pilot case, where a script is written that will alert the people operating the network, that there is a problem in the virtualization layer of the NFV that is likely to cause an outage if not addressed. The requirement is that the predictive indicators of severe problems are derived from a mass of problem reports already reported from the virtualization layer in Nokia. Then an algorithm is developed that will predict these sorts of problems and then a script is done that will run in the NFV.

5 Solution development

This chapter contains most of the actual work involved in this study. A data mining process is needed to make the work easier and more controlled. First the method is selected and explained. Then the selected method is applied to the data set of fault data though

the subchapters. They include the data understand and preprocessing phases, modeling phase and evaluation phase, where the data mining results are evaluated.

5.1 Troubleshooting data analysis process

First phase in the development of the predictive script is to gather the data and do data analysis to find out the indicators the can be used to predict severe problems. A commonly used methodology is chosen to have concrete steps on how to proceed and to get reliable results from the analysis. Here, CRISP-DM method is used to process the data mining phase, as it has long history and is the most widely used methodology for data mining (Kurgan & Musilek 2006). Kurgan and Musilek (2006) also suggest that it is the best method for people starting to do data mining. Also, the industry domain of this problem speaks for choosing this methodology. CRISP-DM stands for cross-industry standard process for data mining. It also helps the readers to understand the actions taken and to better compare the results. Although its usage is not increasing anymore, it is still the most widely used methodology for data mining projects (Mariscal et al. 2010).

The CRISP-DM process model is divided into phases (figure 4) that are further divided to generis tasks. The phases are briefly described here, and the following chapters are divided according to these phases. In those chapters the data mining project is carried out to the troubleshooting data. It is not necessary to go through the phases in order and it is common to go back and forth between the phases during a data mining project. There are six phases in a data mining project according to the process model, as seen in figure 4. All phases need to be gone through to complete a successful data mining project.

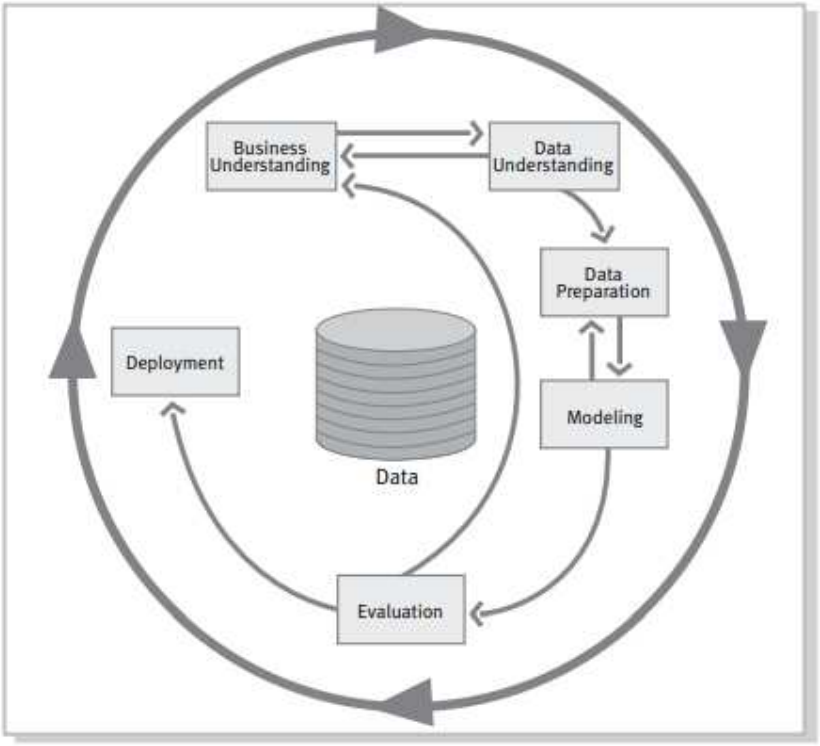

Figure 4: Phases of the CRISP-DM process model (Chapman et al. 2000)

Business understanding phase is about understanding the business problem that is needed to be solved, and then converting that into a data mining goal. It is about making sure that the data is used to produce the wanted outcome form business perspective. In this thesis the first part of the paper discusses these issues.

In data understanding phase the data analyst gathers the data and explores it to become familiar with it and to derive some useful information on how to proceed with the data. Exploration of the data means understanding the general aspects of the data, such as what format the data is in, as well as manipulating the data with some tool so that is can reveal some information on it, such as proportions of the data and relationships of different data attributes.

Data preparation phase is where much of the work is done. Here the data is selected, cleaned, attributed and formatted so that it can be used in data analysis or so that the modeling tools can handle it. In this phase data needs to be considered carefully to get a good quality output of the project. Here is helps if the analyst is familiar with the data and with the business.

In the modeling phase the modeling techniques for a particular data mining goal are selected and tested and assessed to see which technique will be used for modeling. Often the selected modeling technique might require going back to data preparation phase to have the data in correct format.

In evaluation phase the results of the model are compared to the business objectives set for the data mining task to see how well the model performs. The model can also be tested in real-life if possible. After this the next steps for the project will be decided.

In the deployment phase the deployment of the model is planned and the whole data mining project is reviewed. This process is followed in the next chapters, starting from data understanding phase. The first phase, business understanding, has already been discussed widely in the previous chapters of this thesis, thus there is no need to repeat it here.

5.2 Data understanding and preparation phase

Data collection is done first to select the data set that will be used in forming the predictive algorithm. Data was collected on 31st of July 2016 from Nokia's fault reporting software. All the faults that were corrected to any cloud infrastructure software were taken into the data set, and only those problem reports were included that were closed or ready for testing. Even though data collection seems like an easy task, it is however difficult to decide which fields are needed for this purpose and which fields do not provide any valuable info. Data cleaning can account to more than half of all the work needed in data mining (Larose & Larose 2015). Data was downloaded as a csv-file that can be opened in Microsoft Excel. As company confidentiality issues need to be considered, all product names were replaced with generic names such as PRODUCT1, PRODUCT2.

Exploratory data analysis is done to the collected data set to get familiar with the data. This is done in order to get preliminary understanding of underlying subsets of data or possible associations. First it is good to look at the different variables that are included in the data set, and then to explore some of the field values for the variables. Here (table 1) is a list of the variables used for data analysis in case of our data set of cloud infra related problem reports:

Problem ID
Title
State
Modified Components
Product
Problem Type
Action Type
RCA/EDA Root Cause Category
RCA/EDA Root Cause
RCA/EDA State
Fault ID
Fault Analysis Title
Fault Analysis State
Root Cause

Table 1: Problem report variables

Below in table 2 is a sample of the data, including the names of the fields. It is good to look at the data to get a general feel of it and perhaps detect some errors in the data.

Problem ID	Severity	Title	State	Product	Problem Type	Modified Components	Action Type	RCA/EDA Root Cause	RCA/EDA Root Cause Category	RCA/EDA State	Fault ID	Fault Analysis Title	Fault Analysis State	Root Cause
NA05919385	A - Critical	one of VMs crashed with error "KVM: unknown exit, hardware reason 31"	Closed	OpenstackA	Software	OS snapshot 20160527-hotfix1	< empty >	< empty >	< empty >	< empty >	FA193665	one of VMs crashed with error "KVM: unknown exit, hardware reason 31"	Technical Analysis Done	< empty >
PR078273	A - Critical	Compute node state change to Degraded after launching cWLC VMs	Closed	PRODUCT1	Software	TiS 15.09, TiS 15.05p4prelim	< empty >	< empty >	< empty >	< empty >	FA101807	Tis WR 15.05-ER : Compute node state change to Degraded after launching cWLC VMs	Technical Analysis Done	< empty >
PR080220	A - Critical	RCP_15_35: CoreDumps in systemctd-journalctl observed in Airframe machines	Closed	PRODUCT1	Software	TiS 15.09, TiS 15.05p4prelim	< empty >	< empty >	< empty >	< empty >	FA101807	Tis WR 15.05-ER : Compute node state change to Degraded after launching cWLC VMs	Technical Analysis Done	< empty >
PR084798	A - Critical	RedHat: new instances sometimes get the same interface multiple times configured	Closed	OpenstackA	Software	openstack-heat 2014.2.3-5.el7ost	< empty >	< empty >	< empty >	< empty >	FA109544	RedHat: new instances sometimes get the same interface multiple times configured	Closed	Problem caused by 3rd party HW/SW
PR086754	A - Critical	Airframe server not reachable	Closed	OpenstackR	Software	TiS 15.09	< empty >	< empty >	< empty >	< empty >	FA111674	Airframe server not reachable	Technical Analysis Done	< empty >
PR087553	A - Critical	Red Hat:Communication gets blocked between controller/compute hosts once launching VMs	Closed	OpenstackA	Software	kernel images 3.10.0-229.20.1.el7, Kernel 3.10.0-229.15.1.el7.x86_64	< empty >	< empty >	< empty >	< empty >	FA112546	Red Hat:Communication gets blocked between controller/compute hosts once launching VMs	Technical Analysis Done	< empty >

Table 2 - First rows of problem report data

The table shows that there is a unique identifier for all the entries, the Problem ID field. All the values in the fields look normal, although all the values for Action Type, RCA/EDA Root Cause, RCA/EDA Root Cause Category and RCA/EDA State are empty. However, when looking at the whole data set, these fields do have values in some row, thus they

should be kept in the data set as they might provide valuable information. This does however present a problem of what to do with the missing values. According to Larose & Larose (2015), there are several methods of handling missing data. The values can be replaced with constant values determined by the data analyst, with values taken at random from the same field in the data set, with mean values for the fields or with values derived from other values in other fields of the data entries.

5.2.1 Root cause field

In this case root causes cannot be selected at random or by means of the values. The root cause of a problem is always difficult to determine and here false data must not be inserted into the data set. When looking at the values found in the data set for these fields, Action Type entries only have values of “share lessons learned” and “local change”. These values do not add any value to answer the question of what causes outages in cloud infrastructure, and so this field will be removed from the data set. RCA/EDA Root Cause is a free text field and so it would be impossible to come up with somehow derived values for it. The constant value of “Missing” will be added to empty cells in that field. RCA/EDA Root Cause Category is a categorical field, but it cannot be filled as there is no indication of what the root cause is. Thus, here also empty cells will be marked as “Missing”. RCA/EDA State is a categorical field that has values of “Assigned” and “Reviewed”. Here a constant categorical value of “Not Started” will be added to all empty entries, to make it clear that there is no RCA/EDA analysis done for these items. In the last field, Root Cause, there are also missing values, and as this is a categorical field, also those will be marked as “Missing”. The search in Excel for empty entries in the data resulted in finding around ten occurrences in Modified Components entries, where there were two values in the field, one empty and one marking the corrected component. In these cases, the empty values had most probably been left there by accident, and so the empty values were removed and the other values in these fields remained. In this way data has been cleaned a lot already, and unnecessary info is not taken into statistical analysis.

Another important aspect of the data are outliers. They are the data points that may be invalid, errors or even valid data points that can cause problems in many statistical examinations. After a rough data cleaning was done in Microsoft Excel, further analysis will be done in R Studio, a graphical user interface for R which is a freeware data analysis tool. The cleaned data is taken to R Studio as a csv file.

All the data in the data set is qualitative and not quantitative, thus many statistical functions cannot be used with this data. Data frequencies of the categorical data are looked at to get a better understanding of the data set. In *prontodata* data set, is was noticed that the RCA/EDA Root Cause category –field takes only 9 values, while other 459 are missing. See table 3 below.

RCAEDArootcausecategory.freq	
3rd Party Products	2
3rd Party Products, Out of Scope	2
Code Error	1
Design Deficiency	1
Missing	459
Missing Requirement	1
Missing Requirement	1
Misunderstanding of design	1

Table 3:Frequency of values in RCA/EDA Root Cause Category field

In this case 98 per cent of values are missing, and the nine values cannot be taken as representative of the root cause categories. This field is removed from the data set. The same is true for RCA/EDA Root Cause –field. It has only six values and so the field will also be removed as well as RCA/EDA state –field. It serves no purpose when all the other RCA/EDA fields were removed. The learning from this is that if there are fields for root cause analysis, they should be taken into use more frequently. Otherwise they have no value for a larger audience.

5.2.2 Modified Components field

Data labelling could be used to put numeric values to data set that represent the categories in the data set, but it is not advised to do so as the values often bring value suppositions with them. However, reclassification of categorical values can be used here. This means that multiple categorical values are grouped together to form fewer categories. In the pronto data set, the modified components field contains 318 unique values for 468 rows as gotten from below commands in R:

```
ModifiedComponents = prontodata$”Modified.Components”
ModifiedComponents.freq = table(ModifiedComponents)
as.data.frame(ModifiedComponents.freq)
```

As this is an interesting field for our study, it must be made more suited for data analysis. Some of the values in this field are such as *ceph-puppet 2.11, ceph-puppet 2.4, TiS15.12 patch007, fm-plugin 2.4, NBI 1.5, openstack-nova-2014.2.3-65 2014.2.3-65, openstack-nova 2014.2.3-69.* The modified components are almost all unique, mostly because they have different versions of the same software component but also sometimes because of spelling errors and different ways of writing the same, for example with a hyphen or without as can be seen from the above examples. Here quite a lot of work is done to make the data usable. writing forms were corrected to place misclassified values to correct ones, such as underscores to hyphens. The reclassification is done according to the research question. Here modified components will be reclassified so that all values pointing to one piece of software, such as *ceph-puppet* or *openstack-nova*, will be classified to the same group. It is important to know where the problems most frequently are in the software, the version is not as important. Version info was removed from the values, as well as all the different writing forms were changes to same ones. There was still one problem with this data, as the field for one row in the data set had one to six values in the modified components field. Modified components field was then saved independently so that all values could be analysed as separate corrections. This was necessary to see which components had had the most problems and could indicate better probability of problems in the future also.

Quite a few problem reports were corrected to company internal documentation or customer documentation. Here these problems were removed from the data set because the focus is on finding software faults that can be detected in the running software. Thus, it does not add any value to add problems that were fixed in documentation. In cases were problem was corrected to software and documentation, the modified component of documentation was removed from the data. In the beginning the data set had 318 unique values in modified components filed. After re-classification and data cleaning, there are 65 unique values

5.2.3 Severity

Severity is the key variable in this data set. Severity is the deciding factor of how possible an outage is from the significant problem situation. The exploratory data analysis objective here is to investigate associations that could lead to an outage. There exist three severities for the errors found. Severity is roughly classified so that severity A is mostly severe and should only be used when there is an outage in the system, severity B in

case part of the system is affected or is other major errors, and C in cosmetic errors that do not interfere with the working of the system. However, severity needs to be treated with caution as every testing engineer can use a different set of definitions. For this data set, it is assumed that A and B severities mean that there is an outage and C-severity means that there is no outage. Of the whole data set five percent of problems are A-severity, 62 percent B-severity and 33 percent C-severity.

5.2.4 Product field

For the whole data set OpenstackA and OpenstackR had the biggest number of faults by far, 220 and 165 each, compared to other products that had less than 20. A 100 percent stacked bar chart (figure 5) was drawn on all faults with different colors for different severities to see if the proportional amount of A severity faults remained the same for the products.

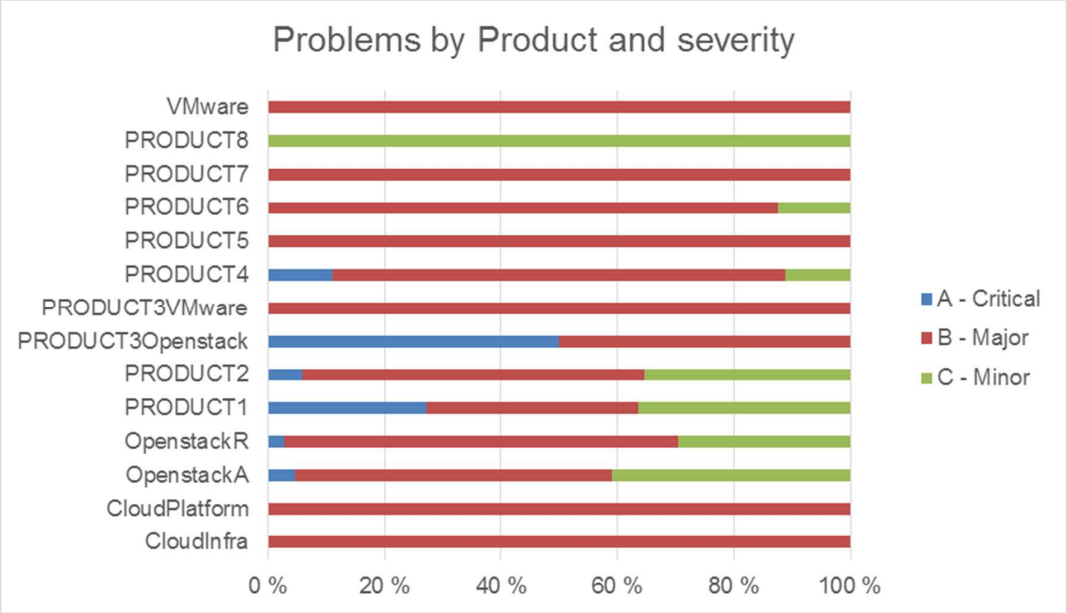

Figure 5: Amount of problems by product and severity

When comparing severity against the product variable it can be seen from figure 5 that only 6 out of 14 products had errors of A-severity: OpenstackA 11, OpenstackR 5, Product1 3, Product2 1, Product3Openstack 2 and Product4 1. It can be seen from the chart that Product3Openstack has the most A-severity faults proportionally but also product1 has more than 20 percent A-severity problem reports.

Modified components field needs also further analysis to see that this field is usable as a predictor for severe faults. For this, we consider this field's relationship with severity field which indicates probable outage. A bar chart and a stacked bar chart of Modified components with severity was done to see whether there is stronger association between certain modified components and higher severity. The components which had over all most corrections, did not have a higher proportion of higher severity problems. Some components are just more likely to have problems of any severity. Only seven components out of 65 had A-severity faults: TiS, cloud-deployer, cept-puppet, OS snapshot, kernel, openstack-heat and symptomreport-event-triggered-puppet. From the bar chart (figure 6) it can be seen that TiS had the most A-severity faults (9) and cloud-deployer second most (6). However, when we compare the portion of A-severity problems found from these modified components to overall portion of A-severity problems in the stacked bar chart (figure 7), it is noticed that TiS has less A-class faults proportionally than the total data set, whereas all other six components have more A-class faults than the overall data.

Thus, we can deduce that there is an association between outage probability and certain components.

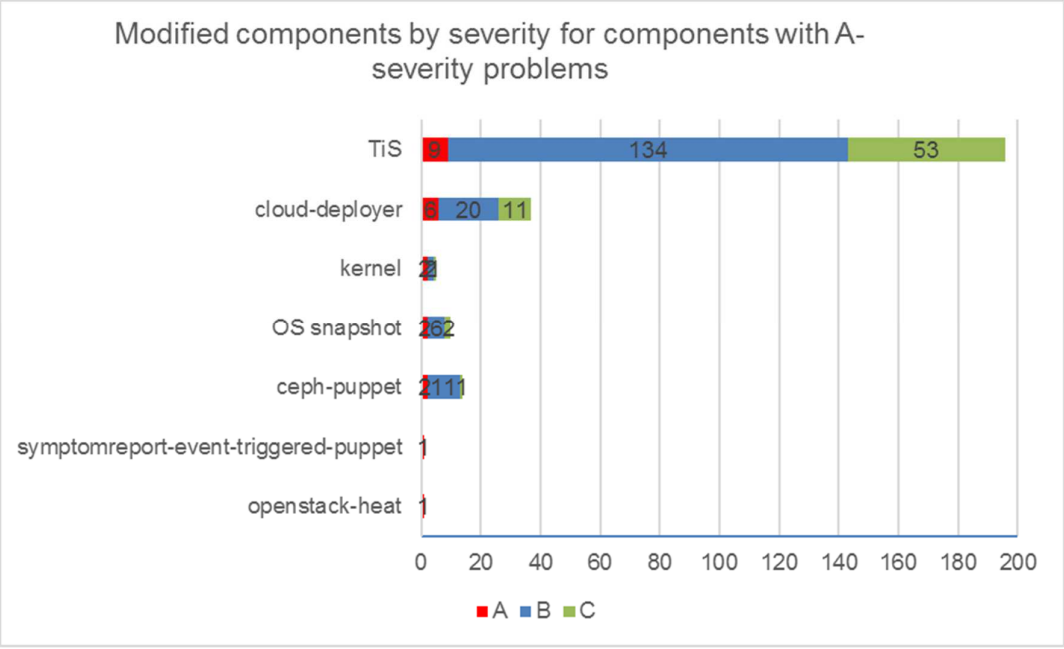

Figure 6: Amount of problems by modified component and severity

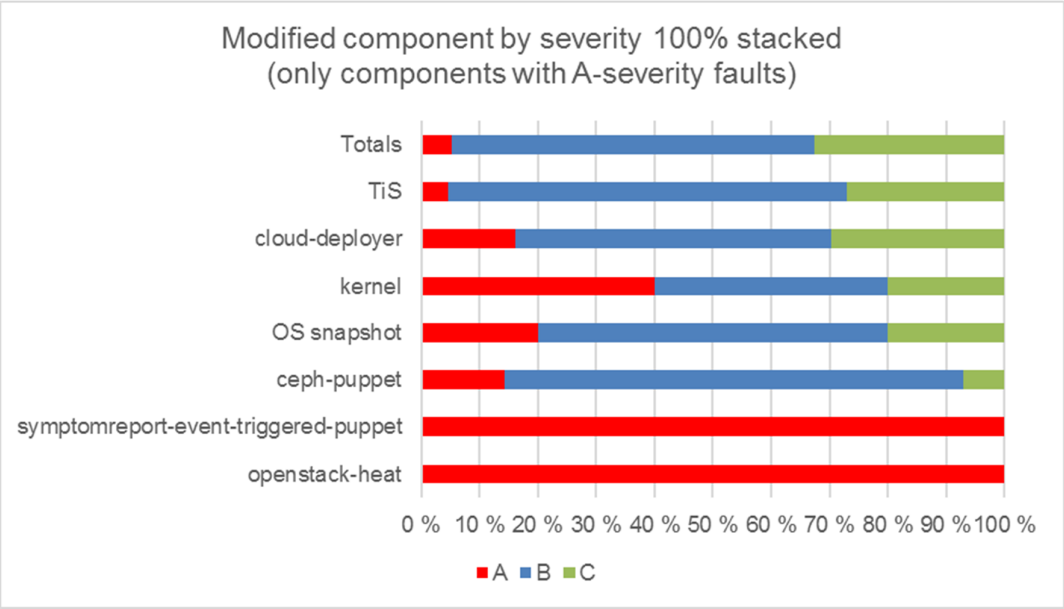

Figure 7: 100 % stacked chart of problems by modified component

In table 4 A-severity faults are cross referenced with product and modified components. TiS problems seem to be only possible in OpenstackR, product1 and product2. OpenstackA seems to have all other components except TiS. This is important information for testing of the model, as TiS problems should be tested in OpenstackR product and all other testing can be done in OpenstackA product.

Modified Components	OpenstackA	OpenstackR	PRODUCT1	PRODUCT2	PRODUCT3 Openstack	PRODUCT4
TiS		5	3	1		
cloud-deployer	4				2	
OS snapshot	2					
openstack-heat	1					
ceph-puppet	2					
symptomreport-event-triggered-puppet	1					
kernel	1					1

Table 4: Amount of A-severity problems by product and modified component

Another interesting thing to look at is the Root Cause field. When comparing the root cause to A-severity faults, most of the root causes are missing. Category *Implementation Specification insufficient/ missing specification* gets three hits for A-severity, *Problem caused by 3rd party HW/SW* gets two. Categories *misunderstanding of requirements*, *Coding error* and *other* all get one hit. When also B-severity faults are considered, *missing* still gets the highest number of faults (189), and after that the three highest categories are *Problem caused by 3rd party HW/SW* (52), *Coding Error* (18) and *IS insufficient /*

missing specs (12). When looking at the data as a 100 percent stacked chart, *Other* category has the biggest proportion of A-severity faults with 50 percent share. Second comes *IS insufficient / missing specs* with 25 percent share and third *Misunderstanding of reqs* with 17 percent. This makes the root causes for the most severe faults clearer. They are caused by badly done or missing specification and misunderstanding of requirements. There is a need to make sure that the specifications are there and that they are understood correctly. The other category however is not very informative. Unfortunately, in our data there is no additional information available for this. This field is not very helpful in modelling the root causes for faults as missing or misunderstood specification cannot be detected from the software code. Thus, this field cannot be used as an input for the algorithm.

5.2.5 Title field

Title field is somewhat different from all the others since the values are all unique and cannot be categorized as such. There is still important information is the title, namely that it states what the actual problem was. As this is unstructured free form text data, text data mining methods are needed. The problem is to investigate whether some text elements can be used as predictors for severe faults causing outages. In this text mining case classification is used instead of clustering, since there are two known groups where we want the bin the fault reports to: outage causing and not. Also, basic text summarization methods are explored to see more complicated algorithms are needed to extract relevant information. One of the text summarization techniques is topic representation. It means that the text is summarized to find words that describe the text. These are called topic signatures. In this way it is easy to understand what the text is about by just looking at a few words.

5.3 Modeling phase

Modeling phase is where the data is tested with different data analysis techniques to find a technique that produces best results for the data mining goal at hand. Here the goal is to learn the symptoms of severe faults. First the product and modified components categories are studied against the severity class to find out if certain products or modified components can be linked to more severe faults. Then the title field is examined. It is a

free form text field and so text manipulations are used to make it more suitable for data mining techniques. Logistic regression and classifiers are tested to find a suitable model.

5.3.1 Product and modified components categories

According to the previous work in the data understanding phase on the data set, the most important categories for the detection of outage related problems are Severity, Product and Modified Components. Severity describes whether the problem in question is in fact an outage or not. In our case, we can classify A and B severities to be counted as outage resulting problems. It is better to detect more problems even if they will not in the end cause a total outage. Therefore, the B severity problems are also counted as outage causing, although they may often cause only partial service breakdowns. Severity C then acts as a non-outage causing problem. This division helps in data analysis. The interesting question is how the modified components and product categories correlate with problem severities and if it can be deduced that they can even act as predictors for an outage related error in software.

When only fault severity and product are considered, there is no strong correlation. None of the default classification algorithms can classify the products according to severity. Mostly all faults are classified as B-severity regardless of the product. Usually the classifiers get around 60 percent of classification right. Also, clustering was measured against the severity classes but there were no correlations found there either. Thus, the conclusion is that product cannot be used to predict the severity of the faults, at least when this training set is used. This is an expected outcome since regardless of product the faults found in cloud infrastructure level should be similar and so the fault severity profiles should be similar.

Modified Components Category is modeled similarly as product category. Modified Components are software modules found in the cloud infrastructure software. First modified components field needs to be converted to nominal instead of string values. In Weka that is done by using StringtoNominal filter. After this conversion, there are 89 entries in the data set with 56 unique values. These values were already cleaned and reclassified in preprocessing phase earlier.

Attribute selection gives only one attribute as significant with comparison to severity. That is modified component nbi. When there is an occurrence of nbi the fault is not severe, whereas when it is missing the fault is severe indicating that nbi is one component that is associated with less severe problems. Otherwise most classifiers cannot tell whether a particular modified component is related to more or less severe faults. There is too much variance in the component field and too little data to have any separation between more severe and less severe faults.

5.3.2 Text classification model for title field

Text classification model is created to explore if the title field brings good predictors for outage related faults. Task here is document-pivoted, because we have constant categories, and new documents are classified to those. Support vector models may be argued to be the most effective way to classify text data. However, as the comparison of different classification models is not easy, it is always best to try different models to see which fits best to the particular data set. For the title field, we will first model support vector model and then try a rule-based classifier. (Sebastiani 2001)

Since WEKA is a more established and user-friendly software for text data mining than R (Feinener et al. 2008), it is used here for this task also. The fact that it is an open-source product has also a lot of significance here for the selection. First WEKA is downloaded and installed from the internet. When it first starts up, user needs to decide which graphical user interface to use. Explorer is the most commonly used one, which includes all the algorithms for data mining. It does not, however, support very large data sets as it stores all the data in memory. In our case here with quite small data set, Explorer will be used to see which data classification algorithm works best for our problem.

WEKA needs the input data to be in attribute relation file format (ARFF). (Witten et al. 2017) Our data is in csv format as it was generated for excel. Arff format is really strict and string type of values are not the easiest for WEKA to understand. Although WEKA does conversion of csv file format to ARFF, at least in this case it did not work very well. It appears that WEKA is quite strict when it comes to special characters and it was quite a lot of work to remove all quotes from the title field text manually (Anon 2017). There is also functionality in WEKA to transform many text files into one arff file. Since most classifiers cannot handle strings, they need to be changed to vectors. WEKA filter stringtowordvector is used to pre-process the text. Multinomial logistic regression from WEKA

model gives 100 percent accuracy for the test set (figure 8), thus it could be used to model the words in the title field according to severity.

```
=== Confusion Matrix ===
   a   b   c   <-- classified as
  22   0   0 |   a = A
   0 276   0 |   b = B
   0   0 145 |   c = C
```

Figure 8: Confusion matrix for multinomial regression

For support-vector model, the only available algorithm by default in WEKA is SMO. SMO stand for sequential minimal optimization algorithm, and it was developed by John Platt. Confusion matrix for SMO looks like this:

```
=== Confusion Matrix ===
   a   b   c   <-- classified as
  22   0   0 |   a = A
   0 276   0 |   b = B
   0   2 143 |   c = C
```

Figure 9: Confusion matrix for SMO

As can be seen from the figure 9, two fault reports were incorrectly classified as B-class. This is acceptable as it is possible that the severity of some faults might be inserted incorrectly in the first place.

However, as the data set is unevenly spread, there might be problems with some classifiers. It is useful to try to make the datasets for all severity classes the same size, so that the proportionally biggest class will not be overrepresented in the classification.
Resample in WEKA is used for this. Resample takes random samples of the data to produce a dataset of desired size. It has biasToUniformClass option that can be used to try to make the classes of similar size. In the data set, there are 22 A-class instances, 276 B-class and 145 C-class. The smallest class A is around 5 percent of the whole data set. Equal distribution can be got in two ways, by overfitting the smallest class or undersampling the majority class. First, we try to double the percentage of the smallest class to see if the classes are then equal size. It produces 14 instances to all classes. However there the whole data set is very much undersampled, but this is a balancing act, as it is not helpful to have too many “fake” A-class instances. Now attribute selection is done to see whether there are some words that can be used to create rules about class selection. InfoGainAttributeEval is used here with Ranker search method. This will rank the words

in the resampled data set and produce a list of which words are most useful to distinguish between the classes. Since the data set is now very small, WEKA finds only two words that are useful: “failed” and “the”. Next visualization is used to check whether easily identifiable pattern can be seen for these words. Figures 10 and 11 show these visualizations.

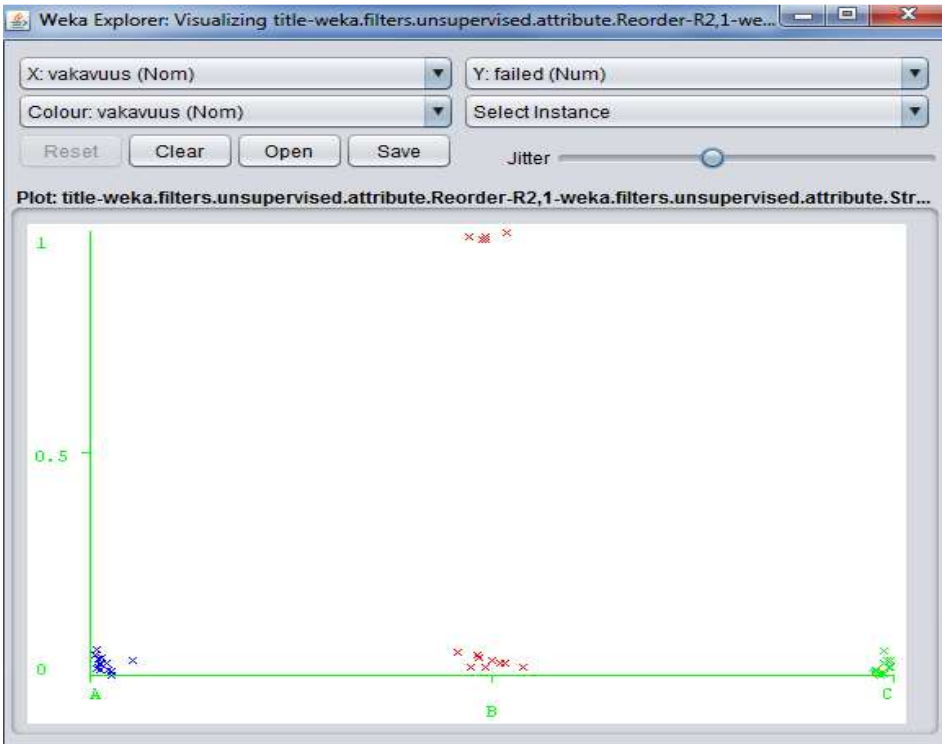

Figure 10: Visualization of word ‘failed’ by severity

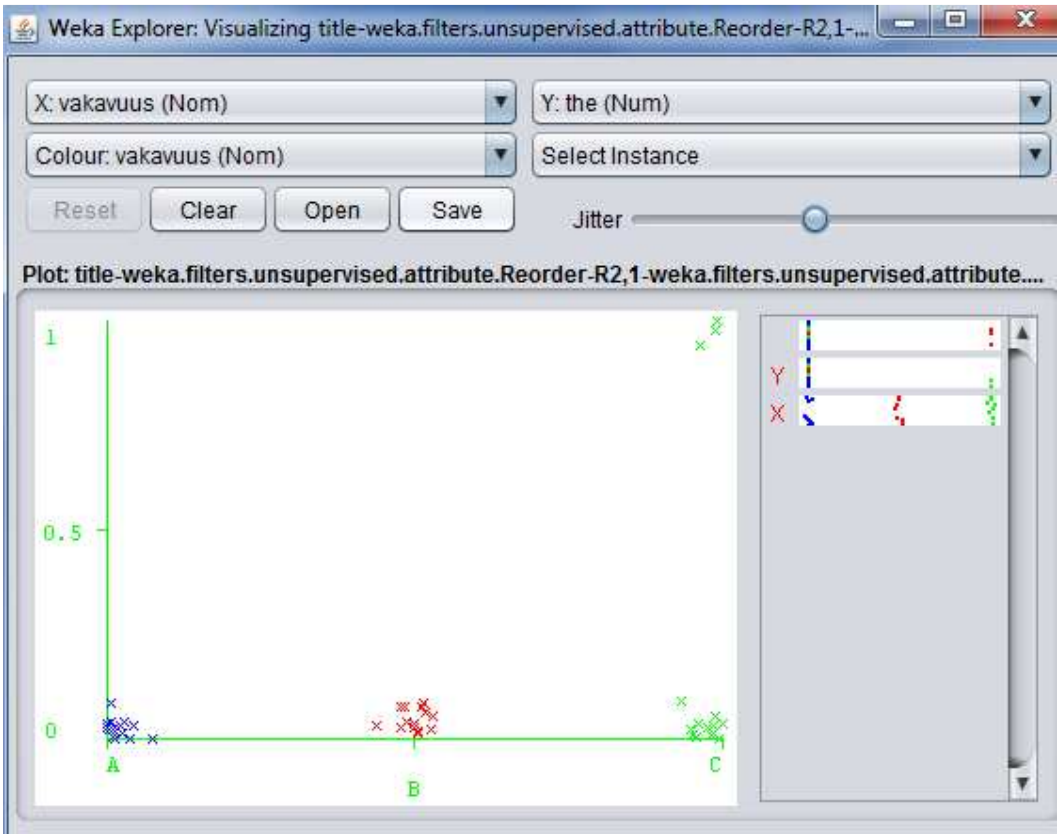

Figure 11: Visualization of word 'the' by severity

From the figures, it can be seen that the word failed only appears is B-class instances, where as “the” appears in C-class instances. “the” however does not seem to be very interesting word in this context. The rules are 100 percent reliable for this very small data set, as can be seen from the figures. All instances where the word “failed” appears are B-severity and all classed where “the” appears are C-class. When we try to use classification of this data set, A-class is always overrepresented. In this sense, the rules are not very plausible.

Another method that WEKA offers for equalizing the classes is the ClassBalancer. It reweights the instances so that all classes have the same weight while retaining the number of instances in each class. When attribute selection is done here, around twenty words are found to be significant in class selections. The number of words in the data set goes down from 1208 to 108. The top-ranking words are: airframe, chu, failed, launching, reachable, exit and test. These all have a ranking value of more than 0,04.

Next problem to model is whether some particular words in the title field can be linked to more or less severe faults. Different classifiers were tested to see which produces best results. In WEKA the selected classifiers were NaiveBayes, NaiveBayesMultiNominal, NaiveBayesMultiNominalText, Ibk, J48, Random forest and SMO. More on how these classifiers work can be read from WEKA tool itself. All classifiers were run with five times cross-validation for the full data set. For changing the title strings to words for data processing with StringtoWordVector, different options were combined with the classifiers. Both word count and binary count were tested. In word count, the number of occurrences of a word is produced by StringtoWordVector, whereas in binary count the word either exists or does not in a title. In this data set both of these produce similar results, because the title field strings are quite short, only a couple of sentences. Thus, words are not repeated usually in a title. Word tokenizer and alphabetic tokenizer were tried. Word tokenizer produces words such as “n”, 2%, 00-01. Alphabetic tokenizer does not take numbers and special characters into consideration. For the purpose of this model, alphabetic tokenizer was seen better as it is easier to evaluate the produces model when the words have connotative meaning.

Usually attribute selection is used in data mining to exclude data that is not relevant to the model. Here one of the basic attribute selection methods was used, InfoGainAttributeEvaluator which evaluates the worth of an attribute in a class. It is used with Ranker that then ranks the relevant attributes that are determined by InfoGainAttributeEvaluator to order of importance. By using this method, the attributes are diminished from 1208 to 18. This means that InfoGainAttributeEvaluator has evaluated that only 18 out of 1208 attributes have value for data mining tasks. In addition to attribute selection also one weight balancer, called ClassBalancer, was tested to see its effects on the model. Weight balancing between classes should be used with caution as most classifiers do not work well with it. It was anyway tested since the data set is very unequally balanced between classes. Results show that this combination with ClassBalancer produces worst results.

Classifiers	StV (word count, alphabetic tokenizer)	StV(word count, word tokenizer)	StV (binary count, alphabetic tokenizer)	StV(word count, alphabetic tokenizer) attribute selection (infogain, ranker)	StV(word count, alphabetic tokenizer), class balancer, attribute selection (infogain, ranker)	StV(binary, alphabetic), attribute selection (infogain, ranker)
NaiveBayes	59,14 %	58,20 %	59,36 %	68,40 %	55,62 %	67,72 %
NaiveBayesMultiNominal	56,88 %	57,56 %	57,56 %	67,27 %	41,00 %	67,30 %
NaiveBayesMultiNominalText	62,30 %	62,30 %	62,30 %	62,30 %	31,50 %	62,30 %
Ibk (k-nearestneighbor)	52,80 %	57,30 %	51,92 %	65,01 %	37,60 %	65,01 %
J48	55,53 %	55,53 %	55,53 %	63,20 %	53,75 %	63,20 %
Random forest	62,30 %	61,65 %	62,30 %	69,98 %	53,40 %	69,98 %
SMO	58,47 %	58,00 %	56,20 %	68,40 %	47,70 %	67,72 %
Average	**58,20 %**	**58,65 %**	**57,88 %**	**66,37 %**	**45,80 %**	**66,18 %**

Table 5: Classifier comparison for title field

From table 5 it can be seen that the most suitable options for almost any classifier are the use of word or binary count, alphabetic tokenizer and attribute selection. These produce the best percentage of correctly classified instances. Results with attribute selection are at least eight per cent better than those produced without it. Also, an interesting percentage is 62.3, which is the result when all faults are classified as B-severity, which is also the biggest class in this data set. As that is not a good result, only results better than that can be considered, and both options that fulfil that criterion use attribute selection. The results of the classifiers average at around 66 per cent with it. When all other options are kept the same, the results are slightly better with word count and binary count. For modeling this data set, attribute selection will be used with word count and alphabetic tokenizer, and with no class balancer.

When selecting the classifier for the data model, it is seen from the table that worst results with the options mentioned above are produced by NaiveBayesMultiNominalText and J48 classifier. Also, Ibk in is the bottom three classifiers. Random Forest algorithm perform best with 69,98 per cent accuracy and NaiveBayes and SMO produce the same result of 68,4 per cent. Since literature supports the use of SMO for text data mining purposes, Random forest and SMO could be selected for modeling this data. Usually the accuracy of a model can be seen from the F measure scores that WEKA gives. For Random forest it is 0.555 and for SMO 0.574.

5.4 Evaluation phase

In the evaluation phase, the models that were produced by the modeling phase are evaluated for their quality and effectiveness. The original research problem is revisited to see how well the model covers all the aspects of the problem. It is also evaluated how well

the data mining objectives set in the beginning are answered by the models produced. After these assessments, it is decided how the model can be used in the real context.

The research question for this data mining task was to find out if a model can be produced to predict outage causing severe faults from cloud infrastructure related fault tracking data. This question was further divided to more detailed data mining questions that were set as:

- How the modified components and product categories correlate with problems severities?
- Can they act as predictors for outage related errors in software?
- Does the title field have good predictors for the outage related faults?
- Can some particular words in the title field be linked to more or less severe faults?

The answer to the first question related to modified components and product categories was given in chapter 5.3.1. Classification or clustering tasks could not find any correlation between product and severity categories, thus product cannot be used to model outage causing faults. Only one modified component was found to cause less severe problems, namely nbi. This information is not enough either to start detecting severe faults, since everything that does not include nbi would then be categorized as outage causing. That is clearly not the situation.

The Title field was then investigated to see whether is reveals more information on severe faults in the system. Title field is a free text form category and so text data mining models were used as described in chapter 5.3.2. Logistic regression model revealed that only words “the” and “failed” correlated with severity category. As “the” is not actually a word, but an article, it cannot be used to tell anything relevant about the data. More occurrences of the word “failed” were seen with more severe problems. That is a relevant finding and could be useful in detecting severe faults, but as an only symptom of a severe fault is cannot be used.

The classifiers SMO and random forest were found to classify many fault reports correctly to different severity classes according to the words found in the problem report title. In evaluating whether these classifiers work well enough, the accuracy of the models need to be looked at. The accuracy of classifiers is measured by precision, recall and F-measure. Precision is the proportion of correctly classified instances divided by all classified instances of one class. Recall is the proportion of correctly classified instances

from all instances that should have been classified in one class. If everything is classified correctly they both score 1.0. Here the precision and recall are around 0,7, which means that seven out of ten instances are classified to correct class and that seven out of ten instances in a class are found. That is quite good percentage although bigger proportion would be nice.(Witten et al. 2017)

The problem, however, becomes that whether this model produces the results wanted for the problem. Even if the model can classify a problem report correctly to a severity class by its title, does this help in building a solution to detect A-class severity problems? Problem report title cannot be used as an input for a script that should detect problems before they happen, since it does not exist at that phase yet. The script would need symptoms that point to a severe problem in the system. The symptoms usually are not listed in the title field and so relying only to the information from the title field is not enough to start building a solution for problem detection.

The data mining problem investigated here was whether fault tracking data can be used to build a model for detecting could infra related faults in the system. The simple answer is no. Most of the data from the fault tracking system was unusable, largely because most of the instances of data were missing in some important categories, such as root cause and root cause category and those categories had to be removed from the data mining task in the first phase. All categories also were free form text fields and that is not most suitable for data mining. A lot of work needed to be done in the data preparation phase for example with the product category. All relevant data was not available in the data set.

The evaluation phase ends with determining what to do with the data mining results. In this case, the results will be used to improve the fault tracking software and its usage so that it is more suitable for data mining and machine learning. That is the direction that the business also is going toward. Then the fault tracking data can be used for any other data mining problems more easily also. Since this is the conclusion, the deployment phase will not be used in the data mining process.

6 Demonstration, evaluation and re-iteration

This chapter examines the demonstration, evaluation and re-iteration phases of the research method. First, the results of the solution development are evaluated against the

objectives set for the study in the beginning. Then the outcome is re-iterated based on the evaluation and a final artifact is delivered.

6.1 Assessment against set requirements

According to the DSRM process model, the next step is to demonstrate the outcome of the design and development and then evaluate it against set requirements. The requirement set in the beginning was that the predictive indicators of severe problems are derived from a mass of problem reports already reported from the virtualization layer in Nokia. Then an algorithm is developed that will predict these sorts of problems and then a script is done that will run in an NFV. The first requirement was not met as described in the previous chapter on data analysis evaluation. Therefore, a script was not possible to be written nor run in an NFV. Higher level requirements obviously were not then met either. They were about running the script with five-minute intervals in the VNF for real time monitoring and reporting, and that the VNF in question was run in Nokia Airframe hardware.

6.2 Re-iteration of the outcome and artifact

Since the design and development phase of the process was not very successful and there was no real artifact produced as an outcome, it needs to be iterated what value the work could have as an outcome. As suggested earlier in the CRISP-DM process evaluation phase, some improvements to the data produced by the problem reporting would be of value in the future for the company. The original business problem of predicting severe faults before they happen still needs to be addressed and problem reporting is a key component in understanding what the problems could be. An improvement report is then provided as a new outcome of the work. It is briefly summarized here and provided as an appendix which will be delivered to the company as a suggestion for tool improvement and process improvement tool.

The improvement report (appendix 1) includes learnings from this study and reasons why the improvements are needed. It also has detailed suggestions for adding new fields to the tool. These are related to the symptoms of the problems, so that it will be easier in the future to learn to detect problems in the network. Another suggestion is to make the fields numeric or categorical, as almost all fields are now free-form text. This helps a lot in data preprocessing phase when the data does not need to be handled manually and

it is easier to handle by the data analysis tools. In this thesis a lot of work was done in the preprocessing phase because the same products and modified components were written differently in many reports although they were supposed to be the same. There were a lot of missing values in the data. The improvement suggestion is to take those fields into use and make the usage mandatory. With these improvements the data produced by the tool will be much more usable for data analysis in the future.

7 Discussion

The objective of the study was to develop a solution for the prevention of cloud infrastructure related outages in telco cloud. This was to be done by analyzing cloud infrastructure related faults to find symptoms of outage causing problems and then creating a script to monitor a virtualized network function and alert the user of possible problems. There existed already a framework called Preventive and Troubleshooting Framework that would have handled the data collection and running of the script.

Developing the whole solution proved to be too big of a task as the work progressed. The data analysis part of the development took a lot of time, mostly because data analysis was something new for the author. It was, however, valuable use of time because of all the learning that was gathered from it. It also proved to be such a big task that it became to focus of this thesis. The process was also hindered by timing issues from the company.

Although the outcome of the process was not what the expectation was in the beginning, it still provides valuable and important information for the company. It also brings the company closer to its goal of using machine learning in every part of it. As the original problem was not solved in this thesis, it is still a valid problem that needs to be solved. After implementing the changes suggested by this thesis, the problem can be revisited. Similar work can be done on other layers of the software, the application layer and the hardware layer.

References

Aggarwal, C.C. & Zhai, C.X., 2012. *Mining text data*, Springer. Available at: https://nsnacademy.skillport.com/skillportfe/main.action#summary/BOOKS/RW$116776:_ss_book:54151 [Accessed April 13, 2017].

Anon, 2017. weka - ARFF (stable version). Available at: https://weka.wikispaces.com/ARFF+%28stable+version%29 [Accessed April 15, 2017].

Chapman, P. et al., 2000. *CRISP-DM 1.0 (Step-by-step data mining guide)*,

Feinener, I., Hornik, K. & Meyer, D., 2008. Text Mining Infrastructure in R. *Journal of Statistical Software*, 25(5). Available at: https://metropolia.finna.fi/PrimoRecord/pci.doaj_xmloai:doaj.org%2Farticle:a6d8fc4380164a99a222d217e1fcdd5d [Accessed April 14, 2017].

Kurgan, L.A. & Musilek, P., 2006. A Survey of {{Knowledge Discovery}} and {{Data Mining}} Process Models. *The Knowledge Engineering Review*, 21(01), pp.1–24. Available at: https://search-proquest-com.ezproxy.metropolia.fi/docview/217510215/?pq-origsite=primo [Accessed October 28, 2018].

Larose, D.T. & Larose, C.D., 2015. *Data mining and predictive analytics* 2nd editio., IEEE Press.

Li, Z. et al., 2013. The Cloud ' s Cloudy Moment : A Systematic Survey of Public Cloud Service Outage. *International Journal of Cloud Computing and Services Science*, 2(5), pp.321–331.

Mariscal, G., Marbán, Ó. & Fernández, C., 2010. A survey of data mining and knowledge discovery process models and methodologies. *Knowledge Engineering Review*, 25(2), pp.137–166. Available at: https://search-proquest-com.ezproxy.metropolia.fi/docview/356893723/?pq-origsite=primo [Accessed October 28, 2018].

Peffers, K. et al., 2007. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 8(3), pp.45–77.

Sebastiani, F., 2001. Machine Learning in Automated Text Categorization. Available at: https://arxiv.org/pdf/cs/0110053.pdf [Accessed April 13, 2017].

Tonkin, E. & Tourte, G.J.L., 2016. *Working with text : tools, techniques and approaches for text mining*, Available at: https://nsnacademy.skillport.com/skillportfe/main.action#summary/BOOKS/RW$239810:_ss_book:115845 [Accessed April 12, 2017].

Witten, I.H. et al., 2017. *Data mining : practical machine learning tools and techniques*, Morgan Kaufmann Publishers. Available at: https://nsnacademy.skillport.com/skillportfe/main.action#summary/BOOKS/RW$250813:_ss_book:120122 [Accessed April 14, 2017].

Appendix 1: Improvement Suggestion Report for Problem Reporting Tool

Pronto Tool Improvement Suggestions for Data Mining

1 Introduction

Problem report data could be very useful in the context of data mining and machine learning. It was found out during a Master's thesis work that the data produced by the tool is not very well suited for data analysis. There the objective was to create a script based on pronto tool information that would alert the user on symptoms that could cause severe problems in could infrastructure software. The objective was not realized as the data was not good enough. Nokia, however wishes to do machine learning, and so here are learnings from the thesis work and suggestions for making the tool better suited for data analysis. The improvement items are divided to two sections: one for the tool itself and the other on how to use it.

2 Improvement suggestions for pronto tool

- Most of the fields in the tool use free-form text. That is not the best solution for data mining because data mining works best with numerical values, and if that is not plausible then with categorical values.
    - change PRODUCT field to categorical with predefined values
    - change MODIFIED COMPONENTS field so that it will search for previously used values and suggest them
    - Change RCA/EDA ROOT CAUSE field to categorical predefined values
- Separate the component and its corrected version in the MODIFIED COMPONENTS field
- Change MODIFIED COMPONENTS field so that each component will have its own row when ported to excel csv file.
- The symptoms of the problem are very interesting for learning. Usually the symptoms of problems come in the form of log writings or alarms.
    - add field for symptomatic log writings
    - add field for symptomatic alarms and how often they occur

3 Improvement suggestions on the usage of pronto tool

- Some important fields from the data mining point of view are not in use at all or most of the values are empty
    - Take RCA/EDA ROOT CAUSE CATEGORY field into use as a mandatory field
    - Take RCA/EDA ROOT CAUSE field into use as mandatory field
- Take the new fields for LOG WRITINGS and ALARMS into use