Aalto University
School of Science
Degree Programme in Computer, Communication and Information Sciences

Jussi Ojala

On Analysis of the Predictive Maintenance of Railway Points

Processes and Possibilities

Master's Thesis
Espoo, January 17, 2018

Supervisor: Professor Aki Vehtari, Aalto University
Advisor: Jyri Eskelinen M.Sc. Technology

Aalto University
School of Science
Degree Programme in Computer, Communication and Information Sciences

ABSTRACT OF MASTER'S THESIS

Author:	Jussi Ojala		
Title:	On Analysis of the Predictive Maintenance of Railway Points Processes and Possibilities		
Date:	January 17, 2018	**Pages:**	xi + 73
Major:	Machine Learning and Data Mining	**Code:**	SCI3044
Supervisor:	Professor Aki Vehtari		
Advisor:	Jyri Eskelinen M.Sc. Technology		

A railway network is vital condition for a blooming industry and fluent public transportation in most countries. To maintain safety and fluency in the traffic it needs to be constantly repaired. A pivotal part of the network – railway points and their maintenance actions, is heavily regulated, leading to periodical visits to the points. However, these visit do not prevent all failures in the railway points and additionally are very costly. Scientists are constantly seeking possibilities to achieve condition-based regulation. However, all the reasonable approaches studied require both a lot of investments in additional equipment and the co-operation of several companies (those responsible for different aspects of the network).

Recently cloud-based digitalisation in the railway industry has made multi company co-operation more practical and brought possibilities for data analysis. This thesis describes three aspects that are necessary to gain more from digitalisation in this field. First, principles and challenges of data analysis project. Secondly thesis investigates feasibility of railway point failures prediction between periodical maintenance visits with existing data. Thirdly, company co-operation requirements regarding data quality, and the formats of signalling logs and maintenance reports.

The emphasis is on feasibility analysis and, with the help of typical machine learning algorithms, this thesis shows that there is potential to improve maintenance planning with existing data. However, the prediction accuracies achieved in the thesis indicates that without investing in additional equipment or more precise log measures, the accuracies are not in correct level to start processes towards condition-based regulation.

Keywords:	data mining, machine learning, predictive maintenance, railway point maintenance, data analysis, data analysis working procedure, log file analysis
Language:	English

Aalto-yliopisto
Perustieteiden korkeakoulu
Tietotekniikan koulutusohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Jussi Ojala		
Työn nimi:	Rautateiden vaihteiden ennakoivasta huollosta Työtavat ja Mahdollisuudet		
Päiväys:	17 tammikuuta 2017	**Sivumäärä:**	xi + 73
Pääaine:	Koneoppiminen ja tiedonlouhinta	**Koodi:**	SCI3044
Valvoja:	Professori Aki Vehtari		
Ohjaaja:	Diplomi-insinööri Jyri Eskelinen		

Useimmissa maissa rautatieverkosto on yksi kukoistavan teollisuuden ja sujuvan julkisen liikenteen elinehdoista. Liikenteen sujuvuuden ja turvallisuuden kannalta sen jatkuva huoltaminen on välttämätöntä. Vaihde on rataverkon tärkeimpiä osia ja sen huolto on raskaasti säänneltyä, mikä on johtanut jaksottaisiin huoltotoimenpiteisiin jokaisella vaihteella. Jaksoittaiset huollot ovat kalliita eivätkä voi estää kaikkia toimivikoja vaihteissa. Rautatiealan tutkijat etsivätkin tauotta mahdollisuuksia siirtyä jaksollisista huolloista tarvepohjaiseen huoltoon. Identifioidut ratkaisut vaativat kuitenkin uusiin laitteisiin ja useiden eri yhtiöiden välistä yhteistyötä.

Lisääntynyt digitalisaatio rautatieteollisuudessa on yleistänyt pilvipalveluiden käytön ja osaltaan helpottanut datan jakamista useiden yhtiöiden välillä. Tämä on luonut uusia mahdollisuuksia data analyysin hyödyntämiselle. Tässä diplomityössä käsitellään kolmea osa-aluetta, jotka parantavat digitalisaation hyödyntämistä uudessa ympäristössä: (1) Data analyysi projektin työvaiheet ja tyypilliset haasteet, (2) mahdollisuudet ennustaa olemassa olevan datan avulla vaihteiden vikaantumista jaksollisten huoltojen välillä, (3) vaatimuksia datan muodolle ja laadulle.

Diplomityön painopiste on kartoittaa mahdollisuuksia hyödyntää olemassa olevaa dataa vaihteen huoltokäyntien priorisoinnissa. Valittujen mallien ennusteet huollon tarpeesta osoittavat, että dataa voi käyttää huollon aikataulutuksen avustamiseen, mutta ilman investointia uusiin laitteisiin ei ole mahdollisuutta siirtyä tarvepohjaiseen huoltoon.

Asiasanat:	tiedon louhinta, koneoppiminen, ennustava huolto, rautatievaihteiden huolto, data analyysi, loki analyysi, data analyysin työvaiheet
Kieli:	Englanti

Acknowledgements

This thesis was founded and carried out in the premises of Mipro Oy. I would like to thank the company and especially Mr. Markus Santanen initiate data science field inside the company for this thesis.

I would like to thank Professor Aki Vehtari for his supervision and helpful discussions during the project. I would also like to thank my advisor Ms c. Jyri Eskelinen and my colleague Jukka Sirviö for providing guidance on database problems in cloud environments as well on any practical issues needed during the thesis.

Moreover, the thesis would have not materialised without all helpful colleagues in Mipro. I would especially like to thank signalling experts Janne Parkkola, Mika Läntinen, Jari Stockhus and Tommi Kokkonen, as well as, rest of the software development team: Jani Kaminen, Saul Stockhus, Ilmo Lehtonen and Aleksi Ikonen. Last, but not least, I would like to thank my daughter, Oona, who constantly provided writing assistance while working from home.

Espoo, January 17, 2018

Jussi Ojala

Abbreviations and Acronyms

AUC	Area Under Curve
CTC	Central Traffic Control
CRISP-DM	Cross-Industry Standard Process for Data Mining
CV	Cross-Validation
ERTMS	European Rail Traffic Management System
ETCS	European Train Control System
GLM	Generalized Linear Model
HDFS	Hadoop Distributed File System
HIMA	HiMatrix
IET	Institution of Engineering and Technology
LOO	Leave One Out
MLPD	Mean Log Predictive Density
RFE	Recursive Feature Elimination
ROC	Receiver Operating Characteristic i.e. Sensitivity
SIG	Special Interest Group
SMO	Sequential Minimal Optimization
STD	Standard Deviation
SVM	Support Vector Machine
TCS	Traffic Controll System
TMS	Traffic Management System
Q–Q	Quantile–Quantile

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The railway industry is pivotal to most countries in the world. Thus, it is heavily regulated. Typically, it has been started as a state-run operation and later parts of the ecosystem have been privatised. This process has been ongoing in Finland since the 1990s. The best- known part of the system is the passenger rail service operated by the VR Group. However, a fundamental part is the railway network where the trains are operated and in Finland this is mastered by the Finnish Transport Agency and the biggest player is Finrail, which controls train movements and manages the safety of the railway network. Other important entities on the operational side are real estate and maintenance service providers. Besides these companies there are different equipment vendors, contractors and sub-contractors that together provide the working railway network structure. Along with fragmentation on the business side, there is a tendency to integrate the sub-systems of different companies in order to utilise cross-system information to gain more from digitalisation. One of the aspects that could gain a lot from combined system information is maintenance planning. In Finland, rail-related problems explain roughly one third of the delays [Liikennevirasto, 2012] caused by network operation. Failures in the railway point immediately affect the fluency of train traffic. Among the most common failures are problems that require short immediate maintenance, like greasing, switch tuning and snow-related issues [Liikennevirasto, 2012], [Garcia Marquez and Schmid, 2007], [Oyebande and Renfrew, 2002]. This thesis takes the railway's interlocking equipment provider's point of view and looks at the possibilities that information sharing with maintenance providers could bring to the table in relation to these issues.

The most relevant parts of railway-specific terminology are explained on a conceptual level and the data collection procedure is shown in its correct context. While explaining the railway-related components, the terms "switch",

"turnout" and "railway point" are used interchangeably. During the basic concept's description, the data collection and information exchange format requirements between maintenance systems and railway controlling systems are also highlighted. This is essential to allow further co-operation in the future. The possibilities of using combined information to assist maintenance are studied in one example area – Ilmala rail yard (see figure 1.1). From that rail yard, some example switches are selected for the study. Their signalling system-related log files are combined with maintenance data that comes from another company. The data is combined to get an initial understanding of the possibilities to utilise the current data to give more time for caretakers to make their work plans. This study is based on historical data and is the first study of the company on this area. Since data analysis is also a new field in the company, the working process and corresponding challenges are a valuable outcome. Thus, these are included as essential part of the thesis.

.

1.1 Problem statement

Instantaneous failures in railway points cause delays in the traffic and require immediate repair. Any assistance in getting an early warning about these failures would help in the maintenance planning. There is different measurement equipment that could be used to produce data in order to predict these failures beforehand, however before implementing them to the railway system it is important to understand what can be done in the current system to improve the situation. The aim of this thesis is to study the possibilities to predict common railway point failure types from data currently available as a product of maintenance bookkeeping and railway controlling system logging.

The data utilisation and company co-operation in this perspective is new, thus information on the working process and data formats are of additional interest.

1.1.1 The approach to the problem

Initial analysis limited the study of predictability to three common failure types in eight switches, including 19 instantaneous maintenance cases. The information in the log files were compressed to two turn-related times: how long the switch monitoring was without electricity, called *monitor off* time, and how long the point machine engine was on call, called *motor on* time.

[2]http://www.openrailwaymap.org/

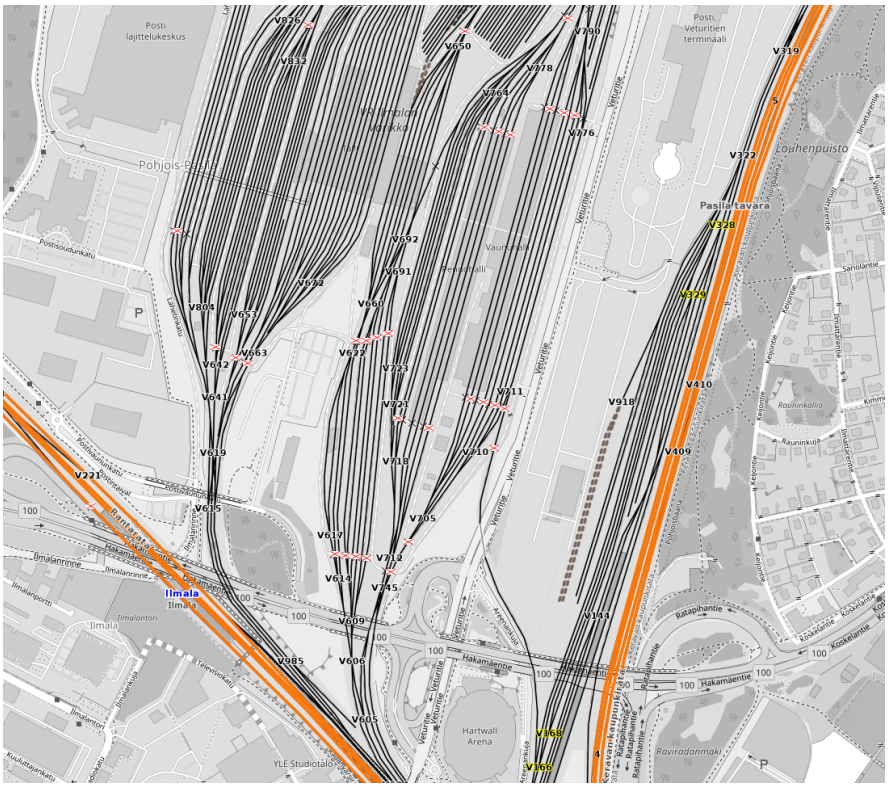

Figure 1.1: An overview of Ilmala rail yard. The different railway switches are marked V619, where 619 is the switch-specific number of that area. A detailed location of the different switches used in this thesis can be found by zooming in on the Ilmala area on the Open Railway Map web pages[2].

Besides these values, the frequency of critical log messages was included in the later phase of the study. The performance of these measures was studied before and after the maintenance in order to see possible differences. To enable distribution-based comparison with common statistical methods, the *independence* of the turns in close vicinity to maintenance was assumed. Several consecutive turns were combined into one sample and characterised with different measures. Statistical supervised learning algorithms were used with *independent* samples to study the achievable prediction accuracy and corresponding error types. The predicted switch condition had two possible states: *normal* (i.e. collected after maintenance) or *maintenance required* (collected before maintenance). The assumption is that one statistical algorithm could predict the selected three failures types.

1.2 The structure of the thesis

This thesis and the selected reference working procedure follow the same phases: understand the background, collect the data, select methods for the study, implement them, produce the results, analyse and discuss the meaning, and go back and forth between different phases.

In chapter 2 the background of the data production process in the related railway environment is introduced. It describes the switches as an essential part of the railway control system and identifies the typical failure types they experience. Signalling log files serve as predictor data and failure types categorise different problem cases, including the one studied in this thesis. Log data is combined around the dates of specified maintenance requests in order to study predictability and the supervised prediction algorithm. This type of problem has been studied in a railway context, thus the key topics of the current research are listed in one sub-section. This clarifies the difference between current research and our study, which is derived from utilising log data instead of much more accurate sensor data. The rest of the chapter specifies the data formats more accurately, including the modifications needed for predictability and prediction accuracy studies.

In chapter 3 the methodologies utilised in this thesis are described. It starts by introducing the reference working procedure that is used to guide the work. Structured work phases are also used later in clarifying the challenges and possible improvements in working habits. This is followed by a short introduction to the theory and history of visual predictability study methods. Later, five prediction models are selected. Their performance is measured using an out-of-sample prediction accuracy measure and this is achieved with cross-validation (CV). The history and principles of this well-

known method are explained before doing the same for each model and their variants (i.e. *logistic regression*, *naive Bayes*, *support vector machine (SVM)* and *random forest*). Besides explaining the model building principle, the chapter includes model-specific feature selection principles and the approach to tuning additional model parameters. The end of the chapter introduces another important measure of the prediction models: their error types and how they can be presented. In chapter 4 the implementation of the work is described. It starts by explaining how the reference working procedure is followed. The circumstances of the work are explained, together with challenges they create. The chapter continues by describing the implementation of data reading and selection from the original format, and also the data used in analysis. This is conducted using the R programming language in the RStudio environment. This decision is justified, together with the implementation choices of important R packages whose working principles and parametrisation are included at the end of the chapter. Chapter 5 includes the major results of the study. First it explains the challenges and learning from the working procedure and continues with an overview of the selected data. This initial view shows horizontal and vertical trends in the data. These were the basic reason for including the frequency of the selected alarm messages as an additional sample characteristic. Moreover, vertical trends indicate that the maintenance-related information is incomplete and for rest of the chapter the study was limited to the data collected in the near vicinity of the selected maintenance cases. The data before and after the selected maintenance events are collected into distributions and compared. This shows a difference, indicating possibilities to predict maintenance requests. The overlap in the event-based distributions highlights the obvious need to use a combination of events (i.e. samples) in the predictions. The properties of the sample characteristics are briefly looked at and this reveals that the selected characteristics are highly dependent on each other.

The end of chapter 5 shows the prediction capabilities of the selected models. Feature selection is utilised in each model and the best set of features is selected. The upper bound of the model accuracy is achieved by calculating the training accuracies and prediction accuracies, which are calculated as a function of distance to the maintenance event. Last, the error profiles of the predictions are analysed.

Chapter 6 discusses possibilities to use the results to assist maintenance planning and possibilities continue the study. It further analyses the importance of different findings and draws conclusions about their meaning.

Chapter 2

Data from the Railway System

This thesis is driven by data that is a side product of both the rail control system and ongoing digitalisation in the Finish railway industry. To understand the starting point of the research we introduce the data's origin in the wider railway industry context. From the Europe-wide harmonisation of the controlling system we look at the operation of a small example area in Finland. At the same time, we clarify the role of traffic management and control work split and show how logging is done in the area control system (the origin of our data).

To understand the context of the technical study and the problem itself, this chapter includes the basic information about railway points and their maintenance. The maintenance reports and failure classes form the basis for problem definition. Predictive maintenance of different failure classes has been widely studied and we describe the focus and findings of that research. Good results from more accurate data used in literature justify our approach to the study. We identify possibilities that a simple combination of existing maintenance information and control logs could bring to maintenance preplanning. At the end of the chapter we compress the collected log data into a form that is more suitable for studying the predictability of the selected failure cases.

2.1 Turnout as part of a rail control system

The umbrella standard for European rail traffic is the European rail traffic management system (ERTMS), which includes rules for train communication, signalling to passenger information. The European harmonisation of the signalling and controlling systems of different European nations is part of the European train control system (ETCS). The standards **of the ETCS**

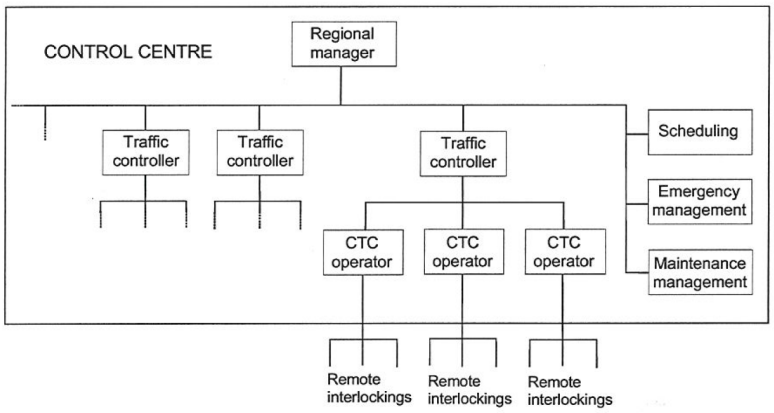

Figure 2.1: The simplified structure of the control centre [Pachl, Second print 2004].

and rules of **ERTMS** aim to harmonise or make backward compatible the interoperability between European country-specific control systems. A similar hierarchy between implemented traffic management systems (TMSs) and traffic control systems (TCSs) follows the operation of any area inside one country. One area-specific TMS includes operation inside the area as well as interoperability between neighbouring areas. The control- and safety-related issues are located in separate signalling and interlocking subsystems that ensure safe and fluent train traffic [Pachl, Second print 2004]. When operating in a specific area in a country, there are several different ways to organise the responsibilities between entities and one of them is visualised in figure 2.1. The regional manager might manage a big area inside one country and solve high-level issues ranging from resource management to different conflict situations. Typically, there is one TMS in one regional area and regional managers have different views about the system open in the control centre. The second level of traffic controllers work below the regional manager and they take care of a smaller physical area and the actual operation there. All of them have their own view of the common TMS. Each view can operate several interlocking systems. In the typical Finish scenario, central traffic control (CTC) operators are missing; one traffic controller operates a bunch of interlocking systems and maintenance is managed by a different company.

A traffic controller gives a train route-related change command via the TMS to the controlling system, which either executes the command with the needed actions or disallows the change. Actions ensure that there is no

collision between trains, that is to say, the route is free and that the correct signalling is arranged for the selected route. When the safety-related actions have been done, the relevant railway points are operated via commands fed to the point machines. In this section, we look at these components from the data collection perspective. We explain where and why the data is created. We first go through the creation of log files and then introduce the maintenance incidents essential to understanding the available maintenance data.

2.1.1 Rail control system logging

In the rail control system, our target area (Ilmala) has been divided into three different control districts. Each district has its own controller and its own TMS. The corresponding interlocking systems then log the actions relating to the railway points. The utilised ALM log files are stored in cloud data storage and transported via OPC protocol. The main data sources used in this thesis are the log files produced by interlocking systems. The simplified version of the logical operation of the system is described in 2.2. Each railway point has its own logical entity that pulls and pushes information to the point machine, which operates the turnout. In 2.2 these are called turn control (TC) logics or Reles, depending on their way of operating. Several such logical entities are collected together in a HIMA (HiMatrix) logical entity that is usually physically located close to the railway and switches connected to it. Multiple HIMA entities are connected to the same OPC server, which is usually physically located inside a building somewhere in the control area.

It is important to notice that each operation gets an operation time stamp in the OPC server, where OPC protocol is used between data storage and Ilmala's servers. This time stamp's granularity in the software is 0.1 sec TC logic, which has around a 20 ms utilisation cycle time, whereas HIMA logic has around 100-150 ms cycle time. Thus, this logging process can introduce approximately 0.2 sec deviation in one specific railway point logging time.

2.1.2 A railway point and its maintenance

There are several different types of railway points, but in essence a railway point allows a train to travel either onto one track or another. The point machine 2.3 operates the turnout normally as a part of the railway signalling/interlocking system. In turn, each problem can cause cumulative delays to the train traffic. Thus, the maintenance of the railway points is tightly regulated and controlled. Periodical maintenance controls are the

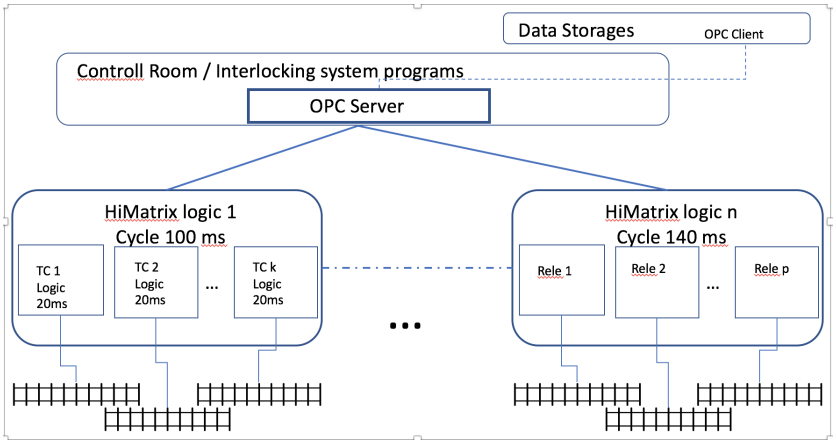

Figure 2.2: The simplified structure of the logging process; there are different types of controlling devices per railway point and the times to provide the information on a specific railway point to the OPC server of the interlocking system vary depending on the location and the time the information enters different logic cycles

main preventative actions that the railway network holder can currently use. The list of the latest guidance can be found on their web pages [Liikennevirasto, 2017]. A description of the general maintenance for a railway point can be found in [Liikennevirasto, 2016a] and in more detail in [Liikennevirasto, 2016b]. In practice this means that on normally operated tracks there are periodical maintenance breaks twice in the year (at the beginning and end of the summer).

Clearly the periodical maintenance cannot prevent all problems in a point machine 2.3 and thus dispatchers need to call maintenance actions on the fly. These maintenance requests form the other source of data. These instantaneous turnout-related failures form a significant amount of track-initiated delays in railway traffic. This was the reason that, on the third of May 2016, one Finnish track maintenance company (VR Track Oy) and rail estate manager (RR Management Oy) organised a seminar on this topic called "Vaihdeseminaari". The presentations in that seminar are not publicly available, but a summary can be found in [Saha, 2017]. This seminar acted as a kick-start to research predictive railway point maintenance. Based on the seminar's outcome we categorise the maintenance requests and actions into the four main categories listed below:

- A: Problems caused by (1) a wrongly adjusted switch – it needs fine tuning; (2) dryness in the switch – it needs oiling; (3) dirtiness in the

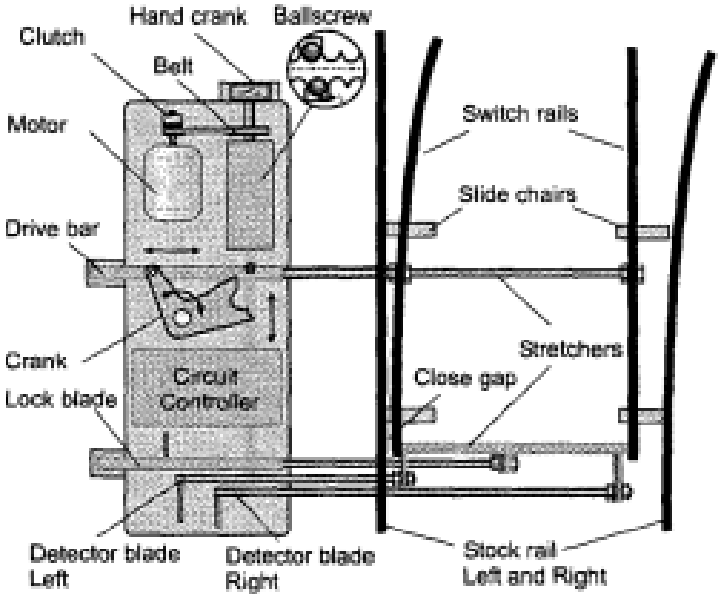

Figure 2.3: A simplified railway point machine [Zhou et al., 2002].

point system – it needs cleaning.

- B: Problems caused by snow or ice (e.g. heating system failure, too much snow on track, ice in system components)
- C: Electrical component failure: (1) point engine failure, (2) problems in cables and wiring, (3) locking failure, (4) contact failure and (4) other electrical component failure.
- D: Other issues: (1) interlocking equipment problems (with the fuse, switchcards, rele ...), (2) passing the switch without permission in the railing point direction w/o functional failure, (3) vandalism, (4) remote control system problems, (5) problems caused by obstacles outside the signalling system (like stones), (6) monitoring problems, and (7) unidentified problems

This granularity of maintenance categories is clearly not enough for good analysis and they should be divided into subcategories. In one of the seminar presentations 25 categories were proposed. However, for the purpose of this thesis it is enough to concentrate on category A, which includes the failure types we want to predict. From the process perspective, we also note that engine failure is part of category C and that was the original problem in the starting phase of the project.

2.2 Research on the predictive maintenance of a turnout

Predictive maintenance in railway systems is widely studied and covers several aspects. We focus only on railway point predictive maintenance research. In the academic world, the focus is on utilising the different sensor or camera information to determine the switch condition. A very good overview of the fault detection principles and articles can be found in [Garcia Marquez et al., 2010]. Even when the research focus has been on sensor and related information exchange architecture, it was shown in an article from 2003 [Garcia Marquez et al., 2003] that the force measures of the switch are sufficient to identify most of the typical failures. They also note as in several other articles that turn direction effect to the prediction accuracy of typical failure and can be over 97 percent in test environment. Among the most common failures in some of researched areas are dryness of the slide chair and the lack of "adjustments" [Garcia Marquez and Schmid, 2007], [Oyebande and Renfrew, 2002].

One common trend in all old industry fields is the change that digitalisation has brought. Most countries have a long legacy of their railway equipment and country-specific regulations. Thus, the approaches to utilising the benefits of digitalisation are varying. However, we took the UK as an example country since the Institution of Engineering and Technology (IET) organised their latest Railway Condition Monitoring Conference in September 2016 in the UK [1]. It targeted UK-related issues, but gave an overview of the topics studied. There were 25 publications from that conference and three interesting topics on the agenda: condition-based maintenance, communication and data standards for RCM application, and predictive analysis tools and techniques. This conference shows that a lot of the literature on data-driven methods still concentrates on how these health indicators are collected and how to use pattern recognition techniques to infer the "remaining life time". The basic assumption is that health indicators are derived by different sensors [Hodge et al., 2015], which typically require additional new devices in the system. For railway switches this requires that at least the engine current consumption is measured and reported [Wright et al., 2016]. Another typical assumption in this research has been that an architecture exists that allows different systems to effectively communicate between each other [Garcia Marquez et al., 2010]. Cloud-based systems are becoming more common in the railway industry, which makes this assumption more valid or technically less

[1] http://conferences.theiet.org/rcm/about/index.cfm

challenging that it has been. Based on the findings that energy measurements would be sufficient for most failure detection, we took an even simpler approach and looked at engine motor times. We did not separate the turn direction to further simplify the situation. There is no direct measurement of engine running times on a railway point machine 2.3, but the interlocking log files have some information on it. Interlocking log files also include other information about railway point machine functions and we selected a bunch of critical measures and the time the monitoring system is out of current as additional information. To the best of our knowledge, there is no reference paper that summarises the possibilities of only utilising the existing interlocking logs and combining that information with the information coming from the maintenance request system. This study gives a first look at the possibilities of this simple approach, which in Finnish circumstances can be done without any new physical devices by exchanging information between the two separate IT systems of different companies.

2.3 Data representation

The data is collected from two main sources: the OPC log files produced by the interlocking system and maintenance requests, and reports given by the real estate management company. Details of the data origins as part of the railway system were explained in the previous chapter 2.1. In this chapter, we explain what information we selected from the data and why. Moreover, we discuss the assumptions made to combine the data from different servers and to shape it into a usable form for the prediction models. Besides the information utilised in our model, additional freely available information could have helped with prediction accuracy (e.g. weather conditions). However, in this thesis additional information possibilities are not considered further.

2.3.1 Log files

All action in a point machine are stored in interlocking log files. The log file includes OPC time and date stamps of different logging events. Besides the different phases of one turn i.e. actions required to change the rail direction, there exists information related to additional equipment and problems in the turn actions. Based on previous research 2.1 and company internal understanding we selected engine *motor on* time as our starting point. A longer motor time means more power is needed to execute the turn. The power itself would be good indicator for possible failure detection. As a result of our initial studies we decided to include two other aspects in our data. The log

These also indicated that it would be proper to measure

Element	Reason to trigger	Other info
SUPERVFAIL_DI	Out of Monitoring. Equipment specific time limits has passed without voltage in point detector.	Often the limit is 10 s
TRAILED_DI	Trailed i.e. the switch has been in monitoring and without using the point engine opened i.e went out of monitor.	-
TURNTIMEOUT_DI	No verification of end position After implementation specific time limit, if not all turn related processes have reported termination	Time limit around 7-8.5 s or 10 s. Possible restarts.

Table 2.1: Log information messages that are related to triggers that have effect on selected measurements, called critical messages in this thesis. Besides these the system logs several other malfunction messages.

files indicating problems in turn actions were defined to be critical alarms. From six identified critical alarms we bunched three together, called *superfail*, *turntimeout* and *trailed*. These messages have an impact on *motor on* times and could serve as additional information on switch condition. Triggering conditions of these messages are included in table 2.1. Besides the frequency of these messages, time that monitoring info of the turn is not available is added as another measure of turn time.

At a high-level log file can be seen as a series of turns and one turn normally gives the log, as shown below. From the log it is possible to extract *motor on* (time between ON and EI) and *monitor off* (time between EI and ON) times of a turn.

```
| 22:06:23,3  | [ILR3] | V619A_SP1STAGE_DI_________6121  | Vaihtokulkutie 1. porras  | EI | 2.9.2013 |
| 22:06:23,3  | [ILR3] | V619A_SP2STAGE_DI_________6121  | Vaihtokulkutie 2. porras | EI | 2.9.2013Â |
| 22:06:23,6  | [ILR3] | V619A_LOCKEDHMI_DI________6121 | Lukittu                                | EI | 2.9.2013 |
| 22:07:27,9  | [ILR3 ]| V619A_SP1STAGE_DI_________6121  | Vaihtokulkutie 1. porras  | ON| 2.9.2013 |
| 22:07:28,0  | [ILR3 ]| V619A_MOTORON_DI__________6121 | Kntmoottorin kyntitieto (SA) | ON| 2.9.2013|
| 22:07:28,3  | [ILR3 ]| V619A_MINUSHMI_DI_________6121 | Asento miinus                          | EI | 2.9.2013 |
| 22:07:28,3  | [ILR3 ]| V619A_SUPERVISED_DI_______6121 | Valvonta                                 | EIÂ | 2.9.2013Â |
| 22:07:28,3  | [ILR3Â ]| V619A_UMLAUF_ID___________6121 | Umlauf-tulo (kntyy/vika/aukiajo) | ON | 2.9.2013Â |
| 22:07:31,2  | [ILR3 ]| V619A_UMLAUF_ID___________6121 | Umlauf-tulo (kntyy/vika/aukiajo) | EI | 2.9.2013 |
| 22:07:31,3 | [ILR3 ] | V619A_MOTORON_DI__________6121 | Kntmoottorin kyntitieto (SA) | EI | 2.9.2013 |
| 22:07:31,3 | [ILR3] | V619A_PLUSHMI_DI__________6121 | Asento plus                      | ON | 2.9.2013 |
| 22:07:31,3 | [ILR3 ]| V619A_SUPERVISED_DI_______6121 | Valvonta                          | ON | 2.9.2013 |
| 22:07:31,4 | [ILR3 ]| V619A_LOCK_ID_____________6121 | Lukitus                             | LUKITUS | 2.9.2013 |
| 22:07:31,7 | [ILR3 ]| V619A_LOCKEDHMI_DI________6121 | Lukittu                           | LUKITTU | 2.9.2013 |
| 22:07:33,6| [ILR3 ]| V619A_SP2STAGE_DI_________6121  | Vaihtokulkutie 2. porras   | ON | 2.9.2013 |
| 22:09:34,2| [ILR3]| V619A_TRACKVACANT_DI______6121 | Vaihdeosuus             | VARATTU| 2.9.2013 |
| 22:09:59,8 | [ILR3] | V619A_TRACKVACANT_DI______6121 | Vaihdeosuus            | VAPAA    | 2.9.2013 |
```

The time between turns tells when a switch has been used before. This is not particularly relevant information considering the complexity it would bring when combining data from different utility switches. Thus, log data of one switch is considered as an ordered series of turns called *events*. Moreover, based on the initial predictability analysis, we grouped these events together to form ordered samples. These are representing a sliding window approach to the real situation and suits better for the selected prediction models. Details of these samples are considered in the following subsection.

2.3.1.1 Event and sample representation of log files

For predictability analysis, the starting point is to look both *motor on* and *monitor off* times separately. For this phase, the triggers serve mainly as an explanation and they are not considered as a measure. However, after that we want to combine the information of these measures together with critical messages. We define one turn data as a collection of log information that relates to change of rail direction at railway point. It starts when the monitoring is off and ends when monitor is on again. We note that due to different triggers (represented by critical messages) or other abnormalities during one *monitor off* time, there can be at least one *motor on* time and anything from zero critical messages to several critical messages. From a modelling perspective, a more important measure is a sample (i.e. one or several consecutive events) and it's characters. Besides sample size the distance in events to/from a "maintenance break" are considered important. In this report, we select seven characteristics for each sample to represent all log-related information.

- x1: The sample mean of the *monitor off* delays. Note that all events that are over 12 secs delay are replaced with 12[2] sec so that individual turn can not fully dictate the result
- x2: The number of different *monitor off* values in the sample; this represents a more robust deviance measure
- x3: The maximum *monitor off* delay in the sample –limited to 12 secs
- x4: The sample mean of the *motor on* times; note that all events that have a delay of over 12 secs are replaced with 12 secs
- x5: The number of different *motor on* values in the sample
- x6: The maximum *motor on* value in the sample (limited to 12 secs)
- x7: The frequency of critical alarms – the number of critical alarm logs divided by two times the sample size

We consider that these characterise a sample well but these can be highly correlated characteristics and a variable selection is used for each model in order to collect a representative set of these characteristics. However even when the switch types selected from Ilmala are similar, there is a difference

[2]Note that if there are some *monitor off* or *motor on* times that are over 12 secs, there are some problems: thus, we can safely keep 12 secs as the maximum level

in these characteristics based on logging procedure (as described in 2.1). For this reason, each event-based measure is calibrated between switches. The calibrations are done by adding a different constant value to *monitor off* and *motor on* measures, where the target is to get a similar mean for each studied switch after the maintenance event.

2.3.2 Maintenance-related information

As described in chapter 2.1, the maintenance of a railway switch happens periodically and as needed. Our aim is to improve need-based maintenance by predicting failures beforehand. The possibility to make predictions from log information is studied. From each maintenance request, there exists information about who made the order, when the order was made, what was the reason for the order, how urgent the order is and reference number allocated to the order. The corresponding report from the caretaker includes free text on what has been done, a coarse classification of the problem position (e.g. a switch related problem) and the reason for the problem, and when the maintenance was finished. This information was compressed to include the request delivery time, the request class 2.1 and finishing time. The compression to the different classes was done by the thesis writer. The information was gathered from two sources and they were only partly overlapping, thus it is likely that it is not complete. Moreover, as explained in 2.1 there is periodical maintenance twice in the year (at the beginning and end of the summer). The exact days of the periodical maintenance were missing.

Based on the assumption that we did not have full maintenance history available, we made a decision to concentrate on data close to maintenance incidents. We further assumed that data after the maintenance would serve as an example of normal behaviour of turns and data before a maintenance request would be studied as to whether there is a clear indication of the coming maintenance request. We studied the possible failure types and identified those that could be predicted based on the phenomena (i.e. the reason behind the failure is evolving over time). Based on the other studies we selected a couple of the most common of such failure reasons and grouped them together to form group A. The maintenance information of the non-switch-related cases were ignored.

In summary, there is no clear rule about when the requests have been sent and it might depend on the person as to when request is done, even though in many cases the need is clear. Similarly, there is no clear rule as to what to write in the report, thus the details in caretakers' reports vary per person. Thus, it was decided to include only those cases which had quite clear categorisation into group A.

Chapter 3

Predictability and Prediction Models

This chapter describes principles and history of the selected methods utilised in thesis. The chapter is divided to three parts first one describing reference working procedure, second the visual techniques utilised to study predictability and third prediction models and their performance measures.

In initial phase time serie plots are utilised as old trend identifying technique. For rest of the predictability and prediction measures studies we will make our most important simplified assumption that hold throughout the analysis i.e. **independence of the studied turns**. This is not true in general, but the positive conclusions remain valid and can be only improved with more accurate data modelling assumption.

Method used to visualise predictability of the maintenance is quantile-quantile (Q–Q) plots for event distributions collected before and after the maintenance. Whereas sample characters are studied with histograms and their correlation properties via bivariate analysis of selected characters. The phenomena seem to be predictable and possible prediction accuracy levels is studied with selected set of prediction models.

The initial set of prediction models are typically selected based on experience and there is no clear general procedure for that [Burnham and Anderson, Second edition 2002]. Our set of different prediction models is based on discussion with supervisor and try to achieve diversity among models and emphasize simplicity of implementation. Selected four different models are described in the end of the chapter i.e. Naive Bayesian, Logistic Regression, Support Vector Machine and Random Forest. Besides models the chapter describes process to measure model performance i.e. cross validation and error analysis.

3.1 Reference working procedure

In data-driven research, problem definitions are often very vague (e.g. due to several aspects that should be included). There are already many definitions of "data science". For example, Cao [Cao, 2012] states that "data science is a new interdisciplinary field that synthesizes and builds on statistics, informatics, computing, communication, management, and sociology to study data and its environments (including domains and other contextual aspects, such as organizational and social aspects) in order to transform data insights and decisions by following a data-to-knowledge-to-wisdom thinking and methodology." Thus, the role of data scientists might also vary greatly inside a company [Kim et al., 2016]. Moreover, often a lot of different information pieces have been collected as a side product of business tasks and at some later point there has been the will to utilise the collected information. In these cases the pruning and problem definition can be a huge task and a data scientist's job reflects more that of a project manager [Lawrence, 2017]. Thus, it is important make companies aware of the different commitments and phases needed in order to get successful results from a data science project. We use the well-known cross-industry standard process for data mining (CRISP-DM) model [Chapman et al., 1999] as our reference model and in the results we highlight the challenges we faced. In this section, the background and main principles of CRISP-DM are highlighted.

3.1.1 CRISP-DM principles

The CRISP-DM process has been developed as an industry co-operation project under EU funding. Five big companies formed the core of the project and collected several of their data mining experts to identify the common factors of their working procedures. The first documented version of CRISP-DM was revealed in 1999 [Chapman et al., 1999], three years after the establishment of the corresponding special interest group (SIG). A short description of the process can be found in [Shearer, 2000]. As in any field, in data mining the details of working processes vary between companies or persons and procedures evolve over time. Despite this, a short poll held among 200 data scientists in the year 2014 showed that the CRISP-DM model forms the basis for the working procedure of many of these data scientists [poll, 2014]. CRISP-DM gives guidelines with which to form a data mining project. It breaks the work into six phases in the following order: (1) business understanding,(2) data understanding, (3) data preparation, (4) modelling, (5) evaluation and (6) deployment. There are iterative steps between phases 1

Figure 3.1: The phases and execution order of the CRISP-DM reference working procedure [Chapman et al., 1999].

and 2, and between 3 and 4, as well as a loop back to the starting phase from phase 5 (see figure 3.1).

There is nothing special about these phases and they seem like common sense. Next, we will go through the main steps with a very short explanation and a summary is included in the following table 3.2.

The starting phase "business understanding" is essential in order to form project objectives and business requirements. All the essential information inside a company should be gathered. This should include essential information on potential business cases, customer requirements, utilised resources, the needed expertise, possible risks etc. All this information should be discussed among all stakeholders and the findings of those discussions should be converted into a data mining problem and a preliminary plan. The second phase, "data understanding", includes the collection of the data sources, data quality and the initial insight to the data. This is an iterative operation with the first phase. These findings should be considered to refine the problem and plan.

The third phase, "data preparation", tackles the data quality issues from a modelling perspective and converts the collected data into a form that can

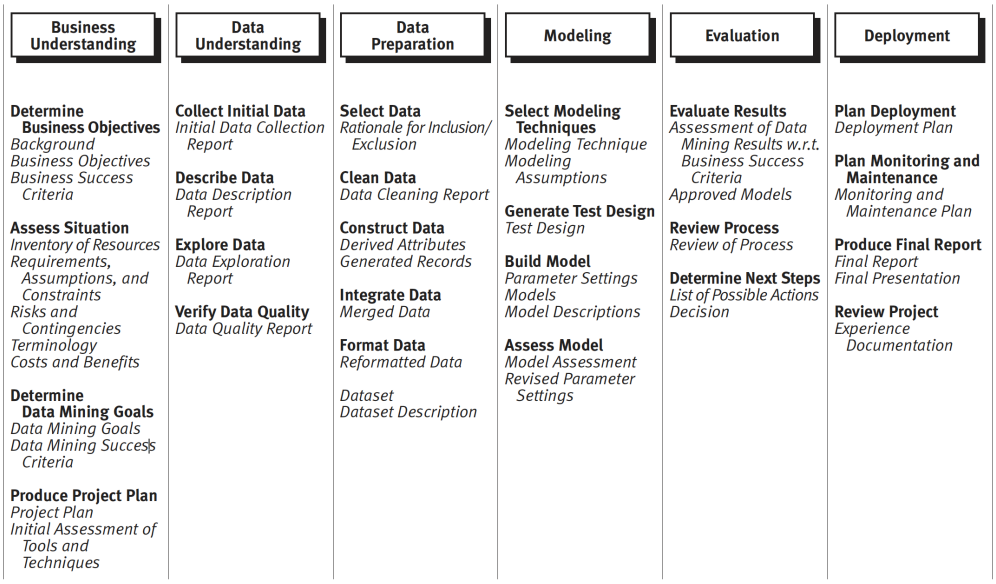

Figure 3.2: Generic tasks (in bold) and outputs (in italics) of the CRISP-DM reference model [Chapman et al., 1999].

be used to follow the current project plan. This is tightly coupled with the forth phase, "modelling", where a set of appropriate machine learning or data mining models are selected and applied to some part of the data. The model itself might have specific requirements from the data, thus an iterative step back is often necessary.

In the "evaluation" phase models are tested against the selected loss functions and generalised against unseen data. The best model is selected based on all key business requirements. This is the phase where the possibilities of the collected data often meet the original wishful thinking and it is time to refine business understanding and start the process again. However, if everything has gone according the original plan it is time to go the last phase: "deployment". This last step usually imports all the earlier data handling code to the company environment in order to produce new information on the original business problem when new unseen data arises. After a successful round of the process, new neighbouring opportunities often arise and a new project, based on the old findings, can be started.

3.2 The predictability of the phenomena

This section gives the background and principles of the methodologies used to get data understanding and to study the predictability of selected failure cases. The methods are old general visualisation techniques, time series plots, Q–Q plots, histograms and scatter plots. The abnormalities in data are further studied with experts in order to find any plausible explanation.

Time series plots gives the initial overall look onto the data. This study is continued by taking a deeper look into the data around selected failure cases. Q–Q plots compare whether measures collected before or after maintenance form statistically different distributions. A difference indicates predictability; however, this interpretation requires **independent** measurement. Scatter plots and histograms can be used to visualise the properties of the collected sample characteristics. The dependency of sample characteristic distributions to the distance from maintenance can be highlighted with multiple histograms. Scatter plots between all sample characteristics show bivariate correlation if it exists.

3.2.1 Initial anomaly detection

A key aspect of the anomaly detection is to understand the nature of the data [Chandola et al., 2009]. The aim is to detect any point, contextual or collective anomalies. The time series plot is one of the oldest versatile graphical representations of data. It is generally attributed to William Playfair's publications from the 1700s, even if earlier publications about it exist [Friendly and Denis, 2005]. Time series plots visualise the overview of the main measures as a function of date. In this thesis, we utilise time series plots for *monitor off* times and engine *motor on* times. The x-axis represents date and the y-axis represents the time the monitor was off or the engine was on during the turn. Maintenance call date information is included with vertical lines in the same plot. The colour of the vertical lines indicates whether the call was categorised as group A or another category (see 2.1). These figures reveal overall trends (collective anomalies), abnormal delay points (point anomalies) or time dependency (a contextual anomaly) that could require further technical explanation or studies. Changing the scales of the figures helps to identify possible smaller time or delay trends.

3.2.2 Q–Q plots

The basics of probability plots, like Q–Q plots, were already developed in 1930 [Wilk and Gnanadesikan, 1934] and the modern way to compare any

two distributions was known in the 1960s [Wilk and Gnanadesikan, 1968]. However, it is still a very powerful way to see whether two one-dimensional unknown distributions are similar. The idea is to evaluate the value of the same quantile points from the cumulative distribution of comparable measures sets, say x_i, y_i. Points (x_i, y_i) represent the two distributions, that is, one distribution on the x-axis and other on the y-axis. There are different ways to evaluate the point values (x_i, y_i) from cumulative distribution. With symmetrically sized large data sets, measurements can be ordered and measurements with the same indexes from both distributions often represent value (x_i, y_i). However, when the number of data points is not sufficiently high or not symmetric, some interpolation technique is used to evaluate the value in a quantile point. Distributions are similar when the points form an approximation of a line and are equal when the Q–Q plot forms a line through origin $y = x$. If the points do not approximate a line, the distributions can be claimed to be different. Special care needs to be taken in the interpretation of the last quantiles if the population size has not been sufficient.

To form cumulative distribution of measurements, an independence assumption simplifies the situation a lot. It is not typically completely true, but it gives a starting point. Moreover, it is often also the baseline assumption of initial prediction models before more advanced models are developed. Our approach was to use the simplified assumption that each turn is independent within the collected data interval. This enables us to combine events and samples for distributions that were utilised for Q–Q plots.

3.2.3 Histograms and bivariate analysis

Another basic visualisation of the probabilistic distribution of independent data points is histograms. Histograms were invented in the 1890s and they provide simple representation of one-dimensional numerical data [Pearson, 1895]. They estimate the probability distribution by dividing a value range into non-overlapping intervals and plotting the "relative frequency" of measurements in the corresponding bin to the y-axis. The frequency is relative to the width of the bin so that the sum of all bin width times' corresponding relative frequencies equals one. The essential information of the histogram is already in Q–Q plots, but for a small sample size, histograms (unlike Q–Q plots) do not include any interpolation, which could hinder interpretation. There are also advantages of histograms, for example the number of overlapping samples before and after maintenance can be clearly identified and location's effect on distribution can be visualised with multiple histograms. Bivariate analysis based on scatter plots is a simple visualisation of the cor-

relation properties. Scatter plots origins date back far in history and even the modern version of bivariate analysis can be dated back to the Herschel paper of the 1830s [Friendly and Denis, 2005]. In bivariate analysis pairs are formed between all characters and the corresponding scatter plots are combined in the same figure.

We utilised histograms and scatter plots for the sample characteristics. The data before and after maintenance are either put into different histograms or included in scatter plot colours/labels.

3.3 Prediction models and measures

This section outlines the history and working principles behind the selected supervised prediction models. The research interest dictates a comparison measure for the models, that is to say, prediction accuracy. The aim is also to understand the achievable accuracy of predictions, the nature of prediction errors and the earliest "time" when the predictions can be made. In the case of having limited data sets, predictive performance is typically estimated using a cross-validation (CV) principle.

For a supervised classification problem, the task is to train the selected models based on the historical data. Assumptions about the data guides the model selection. The data assumptions of **independent samples** simplify the problem to probability estimation (i.e. given the sample values X, they can estimate the probability that maintenance is needed "in the near future", i.e. $p(Y = 1|X)$). The "near future" in the training phase is defined by the data collection period's length. Samples collected before maintenance indicate needs (i.e. $Y = 1$). The number of events in the collection period (i.e. turns before/after maintenance) is one of the parameters of interest. For conditional probability estimation, discriminative models are a natural choice. Among them, for binary response, *logistic regression* is a clear starting point due to its simplicity and generally good performance. However, based on the findings of [Ng and Jordan, 2001], in the case of a limited amount of training data it is wise to test a generative model as well, in this case a *naive Bayesian model*. To achieve variety in our initial model set, and thus more reliability of the accuracy estimation, two additional models were selected. Based on a generally good performance in wide comparisons of machine learning models in [Fernandez-Delgado et al., 2014], two different, well-performing, simple types of model were selected: *support vector machines* (SVMs) and *random forests*.

Model and feature selection can be a difficult task and there are several ways of making the selection [Guyon and Elisseeff, 2003] [Vehtari and

Ojanen, 2012]. Typically, the best predictive performance among reasonable models can be achieved with Bayes model averaging [Piironen and Vehtari, 2017] over a candidate model set. This reduces the need for feature selection or feature creation, but is not practical for implementation. The feature creation methods, like principal component analysis, combine the features in some way and make the interpretation more difficult. Feature selection is typically utilised to make algorithms or models faster or easier to interpret, to improve accuracy and to reduce overfitting. There are three general classes of feature selection algorithms: filter methods, wrapper methods and embedded methods [Guyon and Elisseeff, 2003]. The filter method is independent of the machine learning algorithm and it is usually based on various statistical tests to observe correlation with the outcome feature. Embedded methods are built into the algorithms, for example via regularisation such as that provided by an elastic net. The wrapper methods model is trained to subsets of features and is based on the results' best subset, selected as in recursive feature elimination. With only seven features, complexity or time are not an issue, thus feature selection might only improve training or prediction accuracy. Due to the expected high correlation between features, the wrapper and embedded feature selection methods are tested for the models when applicable.

3.3.1 Predictive performance and cross-validation

Cross-validation (CV) is a well-known method that can be used for several purposes (e.g. for model selection or prediction accuracy estimation). The principle dates back to the 1930s and as a performance estimator it gained more popularity during the early 1970s and late 1960s [Arlot and Celisse, 2010]. The data is divided into training and testing (validation) parts. This is exactly as it is in conventional validation when estimating out-of-sample accuracy; however, in the case of limited data these sets might not be big enough and the model overfits the training data, giving a too positive estimate of accuracy. Thus, this estimate can not be used to represent general prediction error (or accuracy). Cross-validation reduces this generalisation thread by utilising multiple training and testing splits and it averages the prediction error estimate over all the selected data splits. This gives a less biased estimation of out-of-sample accuracy. Besides evaluating prediction error, it is also common to utilise the same procedure inside a training set for parameter tuning. However, the optimisation measure can then also be something other than out-of-sample accuracy.

This procedure introduces complexity to the calculation and it is often categorised according to computation complexity. The two most typical CV

procedures are called leave-one-out (LOO) CV and K-fold CV. K-fold CV makes random partitions of data into K-folds and utilises one at a time for testing and the rest for training. The LOO method is a more compute-intensive, special case of K-fold CV, where K equals the number of data points. Thus, it leaves one sample out (each in turn) for testing and utilises rest for training.

K-fold CV is often more practical due to creating a less heavy computational task than LOO CV, but selecting the correct K is not straightforward. There is trade off in the stability of one CV round's estimate and the number of folds that can be used in the average. More folds mean that there are more numbers to average over and more data to train; however, there is greater variance between estimates. There is no clear best split available, but five folds and ten folds are common selections. To keep the computation reasonable, we selected one version of K-fold CV for our performance estimate. The idea is to use a normal 80/20 split between training and testing data. This ensures that there is a reasonable amount of data with which to test results and get more stable prediction estimates. However, this introduces variability due to the different nature of such a small amount of data – the variation is reduced using four rounds of five-fold CV with a different data split. To get more samples with which to average over, it is common to use repetitions of CV with independent splits. This repetition is still less heavy to calculate than LOO CV and it brings down the variability among CV results better than some other methods (e.g. the bootstrap method [Kim, 2009]).

3.3.2 Logistic regression model

The origins of the logistic model can be traced back to the logistic function present in P. F. Verhulst's studies on population growth in the 19th century. This function gained popularity during the 1930s under probability studies, where it was typically referred as the probit function. However, it was only in the 1970s when the unique relationship of the model to log linear models was discovered [Cramer, 2003] [Nelder and Wedderburn, 1972]. This made the logistic regression model one of the generalised linear models (GLMs) used to predict categorical features. A GLM is a generalisation of an ordinary linear model; it assumes that the response feature y_i follows an exponential family distribution with the mean $E(y_i)$. Some (often non-linear) transformation (*link functions*) of the mean are assumed to be the linear function of explanatory features.

The special binary case (i.e. the logistic regression model) is briefly shown below. The mean of the Bernoulli distribution $E(Y|X) = p = Pr(Y = 1|X)$

depends on linearity of features X via an inverse logit link function (i.e. the logistic function). For binary valued models this means that logistic transformation of the probability (i.e. log odds) are linearly dependent on the predictor features.

$$Pr(Y = 1|X) = E(Y|X) = g^{-1}(X\beta) = \frac{1}{1 + exp(-X\beta)}, \tag{3.1}$$

$$g(E(Y|X)) = log(\frac{Pr(y = 1|x)}{Pr(y = 0|x)}) = \beta_0 + \beta^T x. \tag{3.2}$$

Often the regularisation term R is included in the logistic regression model and finding the optimal parameters gets more complicated than in some other cases (e.g. in the naive Bayes case). However, finding the optimal model parameters is still a relatively easy optimisation problem and regularisation reduces the possibility of overfitting.

$$\beta_{LR} = \min_{\beta,\beta_0} -[\sum_{i=1}^{N} y_i(\beta_0 + x_i^T\beta) - log(1 + exp(\beta_0 + x_i^T\beta))] + \lambda R(\alpha, \beta). \tag{3.3}$$

The model parameters (i.e. the optimisation problem) are then solved in a different manner, depending on the regularisation term and the software in use. Logistic regression is our primary target model due to its simplicity, thus we utilised it with and without regularisation. We can note that the regularisation introduces additional model parameters that are called "tuning parameters" λ, α. On the other hand, regularization can be selected so that it implicitly includes the feature selection in the model.

3.3.2.1 Feature selection and parameter tuning

Feature selection and model tuning can be a tedious task and they form a research field in themselves. Since logistic regression is our target model, two different feature selection methods are utilised. The projection method is found to be less vulnerable to overfitting than the other methods introduced in [Piironen and Vehtari, 2017]. This then is our other method and it is based on the idea of first finding the posterior of the reference model (e.g. the model with all features) and then finding a candidate model (a model with a subset of features) whose posterior distribution is as close to the reference model's posterior as possible [Piironen and Vehtari, 2017].

Implicit feature selection is a simple concept, where terms that favour small or zero coefficients are added to the optimisation function (3.3). This term is called the penalty function or regularisation function. An elastic net is a combination of the two most common regularisation functions L^2

norm (i.e. ridge) and L^1 norm (i.e. lasso) regularisation with a common regularisation coefficient λ.

$$R(\alpha, \beta) = (1 - \alpha)\frac{1}{2}\|\beta\|_2^2 + \alpha\|\beta\|_1. \tag{3.4}$$

The ridge and lasso methods have a lot of similar properties that they bring to the optimisation problem, but with different emphasis. Ridge regression is more effective at keeping all coefficients small, whereas the lasso method favours zero coefficients, however both of them help prevent overfitting to the training data. The tuning of the α, λ parameter can be done with a grid search. The prediction accuracy of the combination is used as a selection criterion and estimated with CV.

3.3.3 The naive Bayesian model

It is difficult to identify the orgins of the naive Bayes or simple Bayes classifier technique, but the roots date back to the invention of Bayes' theorem during the times of Thomas Bayes and Richard Price in the 18th century (see e.g. [Fienberg, 2006]). The naive Bayes method makes a strong independence assumption and utilises Bayes' theorem to form a probabilistic classifier. It assumes that all features are independent of each other given the classifier value. From a Bayesian statistics perspective in a simple two-class case, naive Bayes classifiers form a generative pair for logistic regression [Ng, 2001]. A discriminative classifier (e.g. logistic regression) tries to directly model $p(Y|X)$ observed data and make fewer assumptions, but it depends more on data quality. A generative classifier tries to learn the model that generates the data by estimating the assumptions and distributions of the model. This is used to predict unseen data by calculating $p(Y|X)$ via Bayes' rule. In the naive Bayes model the joint probability function $p(X, Y)$ (generative distribution) is calculated assuming the functional form of $P(X|Y)$ and $P(Y)$. The parameters for them are estimated directly from the training data. Then, according Bayes' rule, the $p(Y|X)$ are calculated and the most likely option is selected for Y. Even a discriminative classifier should generally be favoured due to fewer assumptions and a direct approach, however, based on the analysis in [Ng, 2001], the naive Bayes method might be a better choice than logistic regression when the amount of data is small.

$$P(X,Y) = P(X|Y)P(Y), \quad P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}, \tag{3.5}$$

$$P(Y = y|X = x) \propto P(Y = y)P(X = x|Y = y), \quad given\ \ X, \tag{3.6}$$

$$P(Y = y|X) \propto P(X, Y = y_k) = P(Y = y_k) * \prod_{m=1}^{d} p(x^{(m)}|y_k), \tag{3.7}$$

where m is the feature space dimension and $k \in \{1, 2\}$. The last equation is based on a naive Bayes assumption of conditional independence and the chain rule of conditional probability. Assuming Gaussian distribution for joint probability given the target value, the distribution parameters can be calculated with the maximum likelihood principle

$$\mu_k = \frac{1}{n_k}\sum_{i=1}^{n_k} x_i \quad \sigma_k^2 = \frac{1}{n_k}\sum_{i=1}^{n_k}(x_i - \mu_k)^2 \quad p(y_k) = \frac{n_k}{n}. \tag{3.8}$$

Thus, naive Bayes prediction is the most likely option for the y_k. That is

$$y_{NB} = \underset{y_k}{\operatorname{argmax}}\, p(y_k) \prod_{m=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_{k,m}} exp(-\frac{(x^{(m)} - \mu_{k,m})^2}{2\sigma_{i,k}^2}). \tag{3.9}$$

Thus, this is a straightforward optimisation for maximum likelihood estimation. Given the simple assumption, mean and variance are calculated for each feature in both classes, together with the corresponding class probability. If the independence condition holds, the naive Bayes method is an optimal classifier. However, for small samples it typically achieves good performance, even if the independence assumption is not true [Domingos and Pazzani, 1996].

3.3.3.1 Feature selection

The naive Bayes model with the Gaussian assumption does not require any parameter tuning, thus the only thing to decide before the optimisation problem is to select the correct features. Feature selection is done using recursive feature elimination. The exact algorithm utilises CV with repetitions and includes all subset sizes. For each CV round, sensitivity – that is, receiver operating characteristic (ROC) curves – are conducted for each feature and area under the curve (AUC) is evaluated. For each subset size, the most important features are selected based on AUC values and the model is trained for them and an out-of-sample prediction measure is calculated. Prediction accuracies are used to determine the best subset of the features.

3.3.4 Support Vector Machine

In essence SVMs use hyperplanes to divide data into two class. The idea of using hyperplanes to separate data into different classes was studied in the 1960s, but utilising the kernel trick to get rid of the linearity of the hyperplanes made significant progress in SVM studies in the 1990s [Boser et al., 1992] [Cortes and Vapnik, 1995]. SVMs were originally targeted to two-class classification problems, but they have been generalised to multi-class situations as well. An SVM is a non-probabilistic model and in its simplest form it maps each training observation to a feature's space $x_i \in \Re^d$ and finds an "optimal" hyperplane $H \in \Re^{d-1}$ from which to separate the points into two classes. The "optimality" criteria of hyperplanes may vary according the algorithm. One common optimality criterion called the "maximal margin" maximizes the distance from the hyperplane to the nearest data points in both side (i.e. for both groups). The most simplified linear case can be expressed in mathematical form

$$\underset{w}{\text{minimize}} \, \|w\|_2, \quad \text{subjet to : } y_i(w \cdot x_i - b) \geq 1 \quad \forall i, \tag{3.10}$$

$$y_i \in \{-1, 1\}, b \in \Re, w \in \Re^d, x \in \Re^d. \tag{3.11}$$

$$y_{SVM} = sgn(w \cdot x_i - b). \tag{3.12}$$

Where y is the class, x is the feature vector, $H := (w \cdot x_i - b) = 0$ is the hyperplane and y_{SVM} is the prediction based on the new x. The minimisation of the coefficient w norm is the same as maximising the distance of parallel hyperplanes $H_\pm := (w \cdot x_i - b) = \pm 1$ at equal distance on both sides of the separating hyperplane H.

Data points are typically not linearly separable, thus, in order to be able to find the "optimal" separating hyperplane for "overlapping" groups, soft margins are introduced. Soft margins are represented by the hinge loss function, which assigns a penalty to the data points on the wrong side of a hyperplane. The penalty is proportional to the distance from the hyperplane. This means that in a linear case the method can be reduced to the following modified optimisation problem

$$\underset{w}{\text{minimize}} \, \big[\frac{1}{n}\sum_{i=1}^{n} max(0, 1 - y_i(w \cdot x_i - b))\big] + \lambda\|w\|_2^2, \tag{3.13}$$

$$y_i \in \{-1, 1\}, b \in \Re, w \in \Re^d, x \in \Re^d. \tag{3.14}$$

$$y_{SVM} = sgn(w \cdot x_i - b) \tag{3.15}$$

where new parameter λ is used to balance the margin size and ensure that data is outside the margin's hyperplanes $H_\pm$ (as it should be).

The separating hyperplane and the margin are defined by the model parameters w, b and the optimisation problem can be solved when the tuning parameter λ has been selected. However, this linear version is not typically used and special kernel functions are selected to map the feature space non-linearly to a higher dimension. The kernel function is an essential part of the model. One common selection for the kernel function is the Gaussian radial basis $k(x_i, x_j) = exp(-\gamma\|x_i - x_j\|)$, which introduces the new tuning parameter $\gamma > 0$ (i.e. the width of the distribution). Kernel functions are selected so that the actual transformation to higher dimensional space is not necessary – it is enough to compute the inner products of images for all pairs of data in the feature space (i.e. $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$). Training this kind of SVM is essentially the same as solving a quadratic optimisation problem [Joachims, 2006]

$$\underset{\alpha}{\text{minimize}} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha, \tag{3.16}$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C \quad i = 1, ..., n \quad y^T\alpha = 0. \tag{3.17}$$

Where $e \in \Re^n$ is the vector of ones and $Q \in \Re^n, Q_{i,j} = y_i y_j k(x_i, x_j)$. This corresponds to solving the Lagrangian dual problem instead of the primal problem. The constant is defined by the selection of the tuning parameter λ i.e. $C \leq \frac{1}{2n\lambda}$. To form a SVM classifier we need to find one index, k, for which $0 < \alpha_k < \frac{1}{2n\lambda}$ and $\phi(x_k)$ lies in the boundary of the margin of the transformed space (i.e. we need to find one boundary point in the transformed space with the selected margin distance from the hyperplane). When the dual problem is solved using the primal dual relationship, the optimal w satisfies

$$w_{opt} = \sum_{i=1}^{n} y_i \alpha_i \phi(x_i). \tag{3.18}$$

Utilising this relationship of optimal w, the classifier for new x_t is defined by

$$-b = \big[\sum_{i=1}^{n} \alpha_i y_i k(x_i, x_k - y_k)\big], \tag{3.19}$$

$$y_{SVM}(x_t) = sgn(\big[\sum_{i=1}^{n} \alpha_i y_i k(x_i, x_t)\big] - b). \tag{3.20}$$

For further details, one example procedure for solving the SVM parameters is described in [Rong-En et al., 2005].

3.3.4.1 Feature selection and parameter tuning

For the hyper parameters, some tuning methods need to be selected. The simplest one is a rough grid search of γ and C values. The natural selection criterion for the best parameter combination is the out-of-sample classification error. This can be estimated via CV (as explained in the earlier chapters). The tuning of the parameters needs to be done each time the model is trained to the training data.

Besides tuning these parameters, the feature selection can potentially improve the performance. For variable selection, the same principle explained in chapter 3.3.3.1 is followed due to the relatively easy implementation.

3.3.5 Random forests

The decision tree-based models are based on the principle of finding the most significant features that can generate the most homogeneous split of root population among categories. Each category represents a branch of the decision tree. The most common case is to divide each node into two categories based on one feature. This process is continued in each new node until a certain level, which is usually predefined. The leaf in the bottom level will determine the category.

The decision tree principle is old and while it is difficult to attribute it to one author, Fisher used the principle in 1930 with his famous iris flower linear discriminant analysis paper [Fisher, 1936]. Significant progress was made with tree-based decisions in 1960 when the first regression tree algorithm (i.e. the algorithm for continuous value target features) was introduced. The next major cornerstone was in 1990 when Breiman introduced ensembles as classifiers [Loh, 2014]. The essence of ensemble learning is to train multiple trees differently and use some decision rule, among different proposed predictions, which typically enhance the predictive performance. Random decision trees use an ensemble learning algorithm originally developed in the 1990s [Ho, 1995]. Ensemble trees can be modelled like any supervised machine learning algorithm optimisation problem using the bias variance trade-off of two components: training loss and some measure of model complexity (like regularisation).

$$\underset{f_k \in F}{\text{minimize}} \sum_{i=1}^{n} l(y_i - \hat{y_i}^{(t)}) + \sum_{k=1}^{t} \Omega(f_k), \tag{3.21}$$

$$\hat{y_i}^{(t)} = \frac{1}{t} \sum_{k=1}^{t} f_k(x_i). \tag{3.22}$$

Where each F is a functional space of all classification trees, f_k is single tree solution for k:th data and feature combination, $l(,)$ is a penalty function (e.g. squared loss), Ω is the regularisation function (e.g. ridge or the number of trees) and t is the number of trees. Depending on the algorithm, the shown linear combination of votes can be replaced with a more sophisticated combination of votes. The utilised random forest implementation follows Breiman's paper [Breiman, 2001].

A bootstrap algorithm (i.e. taking samples with a replacement) is utilised to generate t different data sets. For each data set a tree classifier is constructed with a random subset of features. In other words, training data is sampled with replacement t times, for each sample M features are selected. For each subsample of data and features, an optimal tree is constructed. The classification error is used as a measure of top-down construction of the one specific decision tree (for other options see [Rokach and Maimon, 2005]).

3.3.5.1 Feature selection and parameter tuning

The typical random forest value for hyper parameter M is $\lceil col/3 \rceil$ or $\left\lceil \sqrt{col} \right\rceil$. The more trees there are, the better the performance and the number of trees can be decided based on training time; there are no other drawbacks to having more trees in the forest. However, according to practical guidance in [Oshiro et al., 2012], the performance can typically be achieved with 128 trees. The feature selection for the random forest can be done in a similar manner to that described in chapter 3.3.3.1, but reducing the features from seven might not improve the performance.

3.3.6 Error analysis and ROC curves

Out-of-sample prediction accuracy gives one measure of performance but the classification error types often have unequal costs and then one measure is not enough. Thus, in addition to accuracy, receiver operating characteristics (ROC) curves are also shown. The ROC curve measures the classifiers ability to rank positive instances related to negative ones. This is important in any cost-sensitive machine learning algorithms and the ROC curve is a widely-used visualisation technique in this area. As with many graphical techniques, it is not easy to define when it was used for the first time. A variant of the technique was proposed (e.g. in statistics) long before 1950, but in literature ROC curve analysis first appeared in the transactions of the 1954 Symposium on Information Theory. A longer explanation of the history can be found in [Swets, 1973]. The name comes from signal processing in radar technology,

but since that time it has been widely used in engineering, quality control, medical diagnosis, psychology and later in machine learning. The idea of a ROC curve is based on basic measures calculated from a contingency table (see 3.3). It plots the true positive rate against a false positive rate and depicts relative trade-offs between benefits (TP) and costs (FP). The diagonal

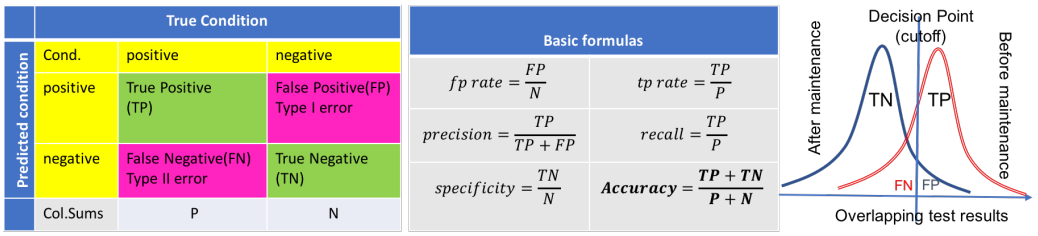

Figure 3.3: A contingency table and basic measures calculated from the table. The right side shows overlapping distributions of negative and positive samples with one random decision point (the cut-off point). These distributions can be combined with the probability of class.

line $y = x$ in the ROC curve represents the strategy of randomly guessing a class. Each sample of predictions forms one contingency table (a.k.a *the confusion matrix*) and thus one point in a ROC curve. Thus, it is possible to form a ROC curve from the total population by ordering all predictions from the most reliable to the most unreliable. Depending on the prediction model, reliability scores can be probabilities or any model internal metric reflecting probability. If true probability values are available, a ROC curve can be formed by sliding the decision point through possible values and calculating the curve points with different cut-off values (see figure 3.3). In practice, the ROC curve is formed by plotting cumulative TP rates vs FP rates in decreasingly ordered probabilities of class one; more details on the algorithm and analysis can be found from [Fawcett, 2006]. ROC curve information is often reduced to single scalar (i.e. the AUC, which, as the name suggests, is the calculated area under the ROC curve). Thus, for a random predictor, AUC $\in [0, 1]$ is 0.5 and for any reasonable model exceeds that.

Chapter 4

Implementation

This chapter describes the implementation of the work. It is divided into three parts explaining the implementation of the working method, data collection and analysis. The first section describes the circumstances in which the work was conducted, how the work phases proceeded and the background to changes made to work plans during the project. The second part describes how we obtained and combined the data from different systems and selected it for analysis. The last but most important part of the chapter enlightens the implementation of the analysis. It explains the logic behind choosing the R programming language and each essential R packet that is used to implement feature selection, CV and all the selected prediction models.

4.1 Working procedure

The circumstances in a company usually dictate the working procedure. This work was ordered from a department that was responsible for service development and deployment. The thesis was written at a different site of the company than that in which the rest of the team, the supervisor and the expert were located. Thus, the information exchange was mainly via web meetings, emails and phone calls, and it was rarely face-to-face and in the same location. The trend inside the company is towards increasing the utilisation of cloud services, which clearly affected the working habits. The approach was reactive, but roughly followed the main principles of CRISP-DM [Chapman et al., 1999] (see chapter 3.1). Implementation is explained in the coming sub-chapter with the help of CRISP-DM principles.

4.1.1 Business and data understanding

This thesis had two major drivers at the beginning of the work. First, the long-lived internal rumour that switch failures could be seen beforehand from the engine behaviour and ownership of the railway signalling log files that were currently being moved to cloud storage. Better business considerations were replaced the time momentum of the topic that was in place in Finland (i.e. considering the delays caused by turnouts via co-operation between different entities [Saha, 2017]).

The original target was decided inside the software team: to predict failure of the railway point engine. The work was to be done in the IBM Bluemix Cloud Service with "IBM Streaming Analytics" tool. This would be one-person work and the team would guide tool environment issues. The plan was to get the first result out as soon as possible and an expert would be consulted inside company on an as-needed basis.

Based on that, the planned work started by getting familiar with the IBM Bluemix and the Streaming Analytics tool. At the same time, maintenance-related data was requested from the other company by an expert co-operating with that company. The first finding was that the Streaming Analytics tool was only suitable for streaming data and not for initial analysis. In order to get some results, essential information of log files was transferred to a laptop and the tool was changed to RStudio. The maintenance-related data was also received and combined for initial anomaly detection. Two crucial findings – that there was only one case of engine failure and some of the abnormal behaviour did not match the maintenance data – led to an iteration round of business thinking. In the iteration round we involved more company internal signalling experts in the discussion, emphasizing the data mining work phases and the need to define potential business targets in the discussion. The maintenance data was found to be incomplete and the decision was to continue analysis, categorise the maintenance requests, concentrate on some of them and ask for more information about the maintenance-related efforts. Categorisation was done by consulting the expert on unclear issues while they were inquiring more maintenance information from the other company. The discussions of inaccuracies in data and classification later led to the decision to only concentrate on three common failure cases until more information on maintenance was available. This was the most crucial phase in the project – defining the problem. The selection was based on the understanding of the experts regarding the predictability of failure classes, as well as on findings from the literature regarding their commonness.

4.1.2 Data preparation and modelling

This is typically a straightforward phase in regard to working procedure. The data preparation and model selection are normally done iteratively and almost simultaneously. The major work can be done alone, but small meetings with the thesis supervisor or separately with signalling experts were organised in order to make decisions about the modelling and data collection. Due to the lack of further knowledge on maintenance, the decision was made that only data before and after the selected maintenance cases would be utilised for predictability studies. The data processing is described in 2.3. The decision to move from log measures to samples was based on the assumption that further history data is needed for predictions, which was verified by the initial analysis. Simple data merging was preferred due to the lack of example cases and the possibilities for easy implementation later. That led to another modelling assumption between switches (i.e. an event-based approach with calibration).

The decision to utilise several modelling techniques was made with the supervisor and this follows the Bayes' principle [Piironen and Vehtari, 2017]. Thus, several variables were selected to represent a sample in order to have the possibility to select the best representation of the data for each model separately. The model assumptions, testing and assessment criteria were discussed in 3.

4.1.3 Evaluation

The work evaluation phase started with a face-to-face meeting of all the stakeholders in the project. The aim was to go through all the assumptions and results, and revise the business potential. The work was presented, the results were interpreted and discussion continued afterwards via email.

There were two revised categories for the business goals. One was trying to predict random maintenance requests in order to get more time to plan the maintenance. This followed the original intention and the results could serve this purpose. However, there was another goal that some of the stakeholders thought of, namely trying to change the periodical maintenance requirement so as to be regulated according to need. These results could not serve this goal.

Throughout the email discussion it was understood that before the deployment phase there was a need to co-operate activities with the maintenance company. More understanding of the different maintenance incidents, predictions time limitations and accuracy requirements are needed. Thus, if this project continues, phase one should be restarted and other compa-

nies should be involved in the studies. From the perspective of this thesis, the work concluded here; however, other actions were planned in order to continue the work.

4.2 Data handling

The R language was selected as the initial analysis language due to its open-source licence and several available statistical modelling packages. RStudio in the laptop environment is a freely available handy tool for initial analysis. Thus, it was a natural working environment after the decision to use laptops (see chapter 4.1.1). The maintenance-related data was received in PDFs and Excel files, which were pre-processed and combined into one Excel file. This file was read in RStudio, where it was combined with the log information. The ALM logs were originally stored in IBM Bluemix Biginsight Hadoop Distributed File System (HDFS) cloud service as ALM text files. They were converted to HIVE tables for easier readability. The appropriate information from these tables were transferred to the laptop. A direct RStudio to HIVE table connection did not work due to JAVA incompatibility issues between the cloud server and laptop that were found during the implementation. Thus, the problem was circulated using Python script, utilising the Beeline subroutine to execute SQL commands in HIVE tables. These results were stored as CSV files on laptops and read by RStudio for further processing.

4.2.1 Combining and selecting data

Two district from Ilmala and data from two separate main servers we re-selected for further study. During the past seven years, these servers had 36–45 instantaneous class A maintenance cases. Clear classification was difficult and for this analysis all the unclear cases were removed. The typical reason for unclarity was that a maintenance report included a combination of actions done without a clear indication of the main problem, or for example they included class A actions (see 2.1.2) but the reason was stated to be a rock between rails. There were also cases where only part of the problems were corrected on the first visit, thus these were removed due to the clean data assumption after maintenance. The clean data assumption was "ensured" by requiring that there were no maintenance calls during the data collection interval. To ensure that data represented a working switch after the maintenance, the requirement was stretched to twice the data collection interval. After removing all unclear cases, eight switches and 19 needs-based maintenance calls was left from two servers. This was assumed to be a suffi-

cient amount of data and it included all the different maintenance aspects of group A. The selected eight switches were physically similar, which allowed us to combine the data in theory. However, as described in 2.1, the logging system of different switches might cause switch-specific variation in OPC logging times. Selecting two different servers ensures that the combination of available data takes all the necessary actions. This ensures that the steps required for implementation are considered at some level in this initial phase.

There are two types of variation in OPC logging times. One is variation due to commands arriving at different times to the logical cycles in the logic components – this is assumed to be random. Another variation in OPC logging times is assumed to be due to the different hardware and software components utilised and this combination might also vary over time. However, based on the assumed reasons behind these differences in recorded times, they can be taken care of with an additional constant. To evaluate that constant, the smaller values of the delay distribution should indicate variation due to architecture, not due to the current condition. Based on the initial analysis (see e.g. 5.1, 5.2), a lower quantile could be a very stable way to estimate the constant. Thus, the constant was evaluated based on the lower 0.25 quantile. The quantile was a bit bigger than typical in order to avoid problems that might occur due to the reactivation of engine in smaller problem cases. This leads to *motor on* times having wider small-value spread. The constant is calculated from data collected during both *monitor off* and *motor on* times after the maintenance for each studied case. To study the assumption behind this calibration, two calibration constants ($C1_{switch_i}, C2_{switch_i}$) were calculated. To avoid unnecessary additions, we utilise the baseline measures 3 sec and 3.4 sec for *monitor off* and *motor on* (i.e. $C1_{switch_i} = 3 - X_{switch_i}$ and $C2_{switch_i} = 3.4 - Y_{switch_i}$). If the assumption on the constant difference between switches is true then the calibration constants $C1_{switch_i}, C2_{switch_i}$ should behave similarly between selected cases. To ensure that there is one calibration case for one maintenance case, the final calibration constant C was the one that proposed smaller change to the distribution location. The different calibration constants are listed in 4.1.

To gain a better statistical interpretation we collected roughly similar-sized samples before and after maintenance. Samples and events were assumed to be independent. This is very simplified assumption, but it makes analysis simple. Q–Q plots can be created and they show whether the data before and after the maintenance request is different. Moreover, to be able to combine switches that have a very different usage, an event-based approach was introduced. Removing the time information, reduce the usability of results, thus table 4.1 was created to guide conversion back to the time perspective when needed. It includes average time that 30 or 150 turns took

Railway point	Maintenance Day	Class	150 events hours	30 events hours	C_1	C_2	C
V601	2015-05-24	A	65.2	16.7	0.0	0.0	0.0
V601	2015-07-21	A	53.7	9.8	0.0	-0.1	0.0
V603C	2013-10-14	A	50.3	8.7	0.1	0.1	0.1
V616C	2016-08-23	A	346.1	90.4	0.1	0.1	0.1
V616C	2016-09-22	A	242.3	55.1	0.1	0.1	0.1
V619A	2011-02-15	A	175.4	11.4	0.1	0.2	0.1
V619A	2013-09-03	A	181.2	25.0	0.1	0.2	0.1
V730A	2014-11-27	A	264.1	42.7	0.0	0.0	0.0
V730A	2015-04-27	A	323.0	63.4	0.0	0.0	0.0
V730A	2015-06-10	A	299.5	37.3	0.0	0.0	0.0
V730A	2015-08-19	A	229.9	32.3	0.0	0.0	0.0
V730A	2016-12-11	A	377.7	58.9	0.1	0.1	0.1
V730C	2015-11-22	A	289.5	50.8	0.0	0.0	0.0
V731C	2012-07-27	A	310.8	42.3	0.0	0.0	0.0
V731C	2013-02-09	A	142.3	24.2	0.0	0.0	0.0
V731C	2013-05-08	A	208.5	32.4	0.0	0.0	0.0
V731C	2014-03-28	A	242.5	40.5	0.0	0.0	0.0
V744A	2015-05-28	A	1661.3	241.9	0.1	0.0	0.0
V744A	2016-03-03	A	2572.3	1199.0	0.1	0.0	0.0

Table 4.1: Selected group A switches and details of the maintenance day, time-to-event conversion and calibration information. The time conversion example is the average time of the 30 or 150 closest events before and after maintenance. The calibration constant C was used for the corresponding maintenance event.

before after maintenance call.

4.3 Analysis

The R language includes several statistical packages, which motivated us to use R language in the analysis. Another reason is the RStudio environment that is convenient for initial visual data studies. This section lists the main functions and packages used in R to do the analysis. In general, the *data.table* package and corresponding format were used for data whenever possible. This package provides an extension to the R data frame format, and conversions from that format to others (e.g. matrix format) was done if required. This format is stable and widely used in the R community. The package provides a *fread* function for the fast reading of CSV files, and for Excel files *readxl* packages provide the necessary functions. The conversion of text to time or date was based on defining the R internal *POSIXct* format or, when

convenient, using the *lubridate* package. Visualisation was mainly done using default graphical functions and *ggplot2*. However, for the feature selection and error analysis *ROCR* and *bayesplot* packages were also utilised. The feature selection itself was the widely used *caret* package. For our primary model (i.e. logistic regression), more effective feature selection was done with *projpred* and *rstan* packages. The predictability study utilised the basic R functions *plot, pairs* and *hist, qqplot*. For histograms, equal-width bins are used to ensure easier interpretation. Besides these, background information on the predictability was gained with a hierarchical clustering function *hclust*. However, those figures fall out of the range of this document due to a lot of repeated information. The main method for the predictability conclusion of the event-based measurements was *qqplot*, whereas other methods were utilised to visualise sample characters.

The prediction models used in the study are implemented in different packages. Logistic regression utilised the *glmnet* package, whereas the naive Bayes and SVM models used the *e1071* package. Random forest implementation was selected from the *randomForest* package. Additionally, Bayesian logistic regression from the *arm* package and the xgboost tree model from *xgboost* were implemented and compared to the other models. However, these models were used to get more confidence in logistic regression and random forest, and were not tuned to the best performance, thus they were not included to the final version of the thesis. In the following sub-sections, we highlight the principles used to construct the prediction models.

4.3.1 Feature selection and prediction accuracy

To avoid any positive bias in prediction accuracy calculations, selecting the correct set of features for the model should be done at the same time as when the model parameters are selected for the training data. However, this would lead to slower implementation and complicate the feature selection presentation in the thesis. Thus, feature selection was carried once for all data, for each selected sample size. This might introduce a small advantage for out-of-sample prediction accuracy measurement, but according our tests the advantage is negligible. A hint of the magnitude of this simplification can be seen by comparing logistic regression with and without elastic net regularisation. The parameter selection of the logistic regression is done in advance to all available data, whereas elastic net does it implicitly every time the model is formed.

For logistic regression, feature selection is done by using the *rstanarm* packet, which provides an R interface to the Stan C++ library. The development environment includes GitHub implementation of an additional

rstanarm-based package "projpred", which implements projective feature selection for logistic regression. As mentioned before, elastic net regularisation does feature selection implicitly each time a model is constructed. For other prediction models the *caret* package is used for feature selection. This package includes the recursive feature elimination *rfe* method that can be used with the caret internal functions *caretFuncs, svmRadial, rfFuncs, nbFuncs* to select the best features. The method was tuned to use ten-fold CV with ten repetitions and it used prediction accuracy as a selection criteria for an optimal set. Additionally, the *caret* package utilises the *klaR* package extension in naive Bayes implementation.

For the out-of-sample prediction accuracy estimate, five-fold CV with ten repetitions was used. CV sets were created using the *caret* package function *createFolds*, and the average and standard deviation of the results were presented. For those models that had tuning parameters, the parameters were optimised in each iteration round separately, using either ten-fold CV or a rough grid search.

4.3.2 Prediction models and tuning

Each of our prediction models has a ready-made R package that is utilised in the implementation. In this section, the parametrisation of the working principle and the parametrization of the corresponding functions are introduced. If the model includes additional internal parameters besides the one estimated directly from data, the tuning principles are explained. For the tuning model's internal parameter, the package's own functions are preferred since they are usually faster.

For making the prediction, general the R-internal function *predict* is used. The parametrisation of the function is similar for all models with a small difference in the required format of the data for different models.

4.3.2.1 Logistic regression

The logistic regression models utilise the *glmnet* packet. The general linear model functions *glm* or *cv.glmnet* are parametrised for the *binomial* family for the classification problem. To solve the parameters used in logistic regression, the algorithm needs to solve the optimisation problem 3.3. Without regularisation, the linear co-efficient are determined by using maximum likelihood equations and this algorithm uses a quadratic approximation of the log-likelihood function and a coordinate descent method to find the optimal point of the function in a regularised regression case. When selecting the regularisation parameters, λ, α are solved separately. The fixed value of $\alpha = 0.5$

is chosen in order to give equal weight to Lasso and ridge regression. This is assumed to be good enough for this point of the analysis, thus this is not further tuned. Each time model is fitted, *cv.glmnet* internal ten-fold CV is utilised to tune the optimal value of λ. The function search sequence of 90 λ values based on internal criteria and select the optimal value of λ based on mean CV error.

4.3.2.2 Naive Bayes

The simple naive Bayes model is implemented in the *e1071* package in *naiveBayes* function. Implementation assumes Gaussian distributions for each feature and estimates both the mean and variance from the equation 3.8 and the prediction simply involves calculating the maximal value of equation 3.9.

4.3.2.3 SVMs

The SVM is implemented in the *e1071* package using *svm* or *best.svm* function. The function is parametrized for *C-classification* using a radial-based kernel. To solve the quadratic optimisation problem 3.16 requires implementing the iterative sequential minimal optimization (SMO) type of decomposition method, which is shown to have fast convergence [Rong-En et al., 2005]. To be able to solve the optimisation problem some parameters for γ and C need to be selected. For the selection of reasonable values for γ and C a grid search is utilised every time the model is constructed with the *best.svm* function. The grid is formed with values $\gamma \in 10^{(-6:1)}$ and $C \in 10^{(-3:2)}$. This is a very rough grid, found with couple of random tests, sufficient for a first estimate; if the study continues, the most promising area of the grid needs to be made more dense.

4.3.2.4 Random forest

A Breiman and Cutler's random forest is implemented in the package *randomForest*, with the function having the same name. The function is parametrised for a supervised mode with classification using 300 trees. Based on theory, this tree number is assumed to be sufficient and yet still very fast to calculate. The algorithm uses $\sqrt{p}$ features and from population size N it takes N samples with replacement to ensure different training data for each tree. The number of features in each training set were not optimised, even if the package included the tuning possibility *tuneRF*, but for such a small set of features, the default value was assumed to be sufficient. The tree order is the based-error rate among the out-of-bag (OOB) portion of the data.

4.3.3 Error analysis

The error analysis data is naturally produced while evaluating prediction accuracy. Thus, it is the product of five-fold CV with ten repetitions. However, each of the prediction functions needs to be configured to output the probability of maintenance need. The ROC curve calculation and visualisation are implemented in the *ROCR* package [Sing et al., 2005], which includes the functions *prediction* and *performance* in addition to a modified interface to *plot* functionality. The performance function can also be parametrized to calculate other error statistics (e.g. AUC values). As explained in [Swets, 1973], ROC curves from different samples can be combined into one curve or averaged over both axes. The average gives a bit more information on the uncertainty since it allows calculation of the standard deviation of the curve point values with different decision boundaries. Organising the data to matrix form (each column equals one test set) or vector form (all the data together), both of these can be deduced with the same principles. From both methods AUC values are also produced and, in addition, precision, recall and specificity are calculated from the contingency table of the combined data using decision boundary 0.5, which was used in the prediction accuracy calculations.

Chapter 5

Results

In this chapter, we describe the major results of the project. It is divided into four parts: the first describes the challenges and learning of the working procedure, the second shows the initial view of the data for the selected switches, the third visualises the predictability of maintenance, together with sample character properties, and the last concentrates on the models' capabilities by showing the prediction accuracy and error profiles.

5.1 Challenges and learnings

There are some general multi-site project problems that we will mention when we cover the challenges experienced during the work, but we will concentrate on the problems and solutions arising due to first utilisation of a data scientist in a department. The challenges arise from the different expectations and initial viewpoints held among the stakeholders, which are emphasised when there is a lack of understanding of the data mining work phase. In this situation data science work can be thought of as a black box, used to achieve an undefined target, and intermediate results are not utilised to their full potential. For example, emphasising the CRISP-DM procedure to all stakeholders at the beginning of the work would have helped the situation. Moreover, it is crucial to get correct stakeholders to follow the first two phases of the work.

We will follow the CRISP-DM work phases in the coming sub-chapters and show some of the challenges we faced and how they could have been diminished.

5.1.1 Business and data understanding

This is the most crucial phase in which to get every stakeholder to work towards a common goal. In our case, the challenges arose from different starting points, expectations and a lack of end-user experts among stakeholders.

A lack of end users in the planning phase can cause difficulties in defining the correct problem to solve. In our case the lack of maintenance experts among stakeholders led to selecting the wrong starting problem, which we changed after first data evaluation i.e. first visit in phase 2. Since the experts were in a different company it was not reasonable to assume that they would have been in the planning phase, however, due to close co-operation with those experts we could have organised a poll among them on the most essential failures and how predictions could help them. A combination of different expectations, starting points and the lack of understanding of data mining working phases in the starting phase can make project planning challenging. In our case, the ongoing implementation aspect dictated the tool selection. In a cloud environment, selecting the tools and handling the data between them is not trivial [Fisher et al., 2017], but understanding the data mining work phases could have prevented the wrong tool selection. The wrong tool was replaced with the temporal solution of utilising a laptop, which could have been avoided by discussing issues throughout the project with some data mining experts that have been working with IBM cloud services. The cloud-based data mining environment was found too late and the change to the working environment of the "IBM Data Science Experience" happened after phase 5, when the main work had already been carried out. The lack of "customer" understanding was another reason that the targets were not commonly shared among stakeholders. Together with the lack of data mining working phase knowledge, the needed commitment from the experts to assign their time for the coming working phases was ignored. This resulted in unnecessary delay to the process and fewer possibilities for experts to benefit from the intermediate results that the data mining process brings.

5.1.2 Data preparation and modelling

This is most straightforward part of the process. The analytics can follow the guidance from other colleagues and common research practises described in 3. Utilised small meetings with experts to understand differences in data to be correctly combine it were found sufficient. Thus, no procedural change is proposed. The small meetings with experts used to understand differences in data and correctly combine it were found to be sufficient. However, explaining the details of data selection and the expected results to all stakeholders

could have pre-harmonised the thinking for the evaluation phase. That could have improved the decision making in phase 5.

5.1.3 Evaluation and the next steps

More precise target is, easier the evaluation. Thus, sloppiness in phase one will introduce same discussions in the evaluation phase. For this phase one or two meetings targeting to decide usability and next steps with all shareholders is sufficient. The work was seen usable to assist maintenance planning, but it would be good to test prediction model for current log file and check the condition of corresponding switches. If this gives positive results, we should suggest co-operation project to Finish Transport Agency.

5.2 An overview of the selected data

The selection of study cases was based on the quality of maintenance information (see 4.2.1). From two monitoring areas in Ilmala, 19 class A maintenance cases from eight different switches were selected for closer study. In this section, an overview of that data is provided in visual and numerical form.

The times between turns can vary based on usage, and the delay distribution location also varies according to software and hardware architecture, as seen in table 4.1. As the table shows, the calibration constant might vary depending whether it was calculated from *motor on* or *monitor off* times; this slightly reduces the trust in the calibration process. However, the trend remains the same between the two distributions, thus calibration was utilised after the initial analysis. Another theoretical assumption was that the logging process would indicate a max 0.2 sec variation in the time measurements depending on in which phase of the logging cycle the data arrives to the equipment. This seems to be a relevant assumption based on tables 5.1 and 5.2, which show that 50 percent of data occur within 0.1 sec after the maintenance. Moreover, it also shows that the same applies to data collected from the whole studied logging period, even though the distribution location might differ slightly in this case.

These tables show that both distributions are very concentrated but that they have tails. The tails are even wider for *monitor off* than they should be based on the reactivation procedure of the engine. However, this reactivation should not allow *motor on* times that are longer than 12 secs, thus *motor on* times that are over 12 secs could be studied further as exceptional behaviour. The tables even provide a lot of information on the correctness of our initial

Statistics	V601	V603C	V616C	V619A	V730A	V730C	V731C	V744A
Min	0.4 (2.9) (2.8)	0.4 (2.9) (2.8)	0.7 (2.9) (2.8)	2.8 (2.8) (2.8)	2.8 (2.8) (2.8)	2.8 (2.9) (2.9)	2.3 (2.9) (2.9)	2.8 (2.8) (2.8)
1st Quad	3.0 (3.0) (3.0)	2.9 (2.9) (2.9)	2.9 (3.0) (2.9)	2.9 (2.9) (2.9)	2.9 (2.9) (2.9)	2.9 (3.0) (3.0)	3.0 (3.0) (3.0)	2.9 (2.9) (2.9)
Median	3.0 (3.1) (3.1)	3.0 (3.0) (3.0)	2.9 (3.0) (3.0)	2.9 (2.9) (2.9)	3.0 (3.0) (3.0)	3.0 (3.0) (3.0)	3.0 (3.0) (3.0)	3.0 (3.0) (2.9)
Mean	3.31 (7.47) (4.56)	3.24 (3.41) (2.95)	3.18 (7.87) (2.98)	3.87 (7.70) (14.2)	3.81 (6.53) (3.73)	3.42 (15.5) (2.99)	3.57 (5.04) (3.08)	14.72 (25.2) (3.44)
3rd Quad	3.1 (3.1) (3.1)	3.0 (3.0) (3.0)	3.0 (3.1) (3.0)	3.0 (3.0) (3.0)	3.0 (3.0) (3.0)	3.0 (3.0) (3.0)	3.0 (3.0) (3.0)	3.0 (11.4) (3.0)
Max	1395 (338) (111)	6130 (59.1) (3.1)	1016 (55.2) (3.1)	4143 (217) (4143)	14850 (129) (290)	18072 (992) (3.1)	1445 (273) (69)	11900 (2538) (109)

Table 5.1: Selected switches with group A problems (see chapter 2.1.2 and Table 4.1) and their summary statistics of *monitor off* times over the log collection period and measurements collected 0–150 turns before and after maintenance calls (in parenthesis). The table includes the minimum/maximum, first, second and third quantile and mean value of the measured *monitor off* times. The calibration constant is calculated as the 1st quadrate point of measurements collected 0–150 turns after maintenance calls.

assumptions. A comprehensive view of the situation is visualised in time series plots 5.1 and 5.2. These figures show when and how often these special cases happen and whether there is any clear trend in them. The figures show the same data twice, with one plot having the y-scale limited to below 12 secs in order to see the most interesting area of the data. The vertical lines (31 of them) represent the maintenance calls that we have information about and that could have an effect on switch-related log times.

The figures show two things clearly: there seems to be some trends, both horizontal and vertical, in the delays and the collected maintenance cases do not explain all the exceptional cases in the data. The second issue was the basis for only considering the data close to the collected maintenance cases. Moreover, these exceptional cases are either quite rare individual cases or they arise as a cluster (i.e. they are a vertical trend). The rare cases are not considered within this thesis and we focus on the situation below 12 secs. We will run through the trends with a bit more analysis in the following sub-section.

Statistics	V601	V603C	V616C	V619A	V730A	V730C	V731C	V744A
Min	0.0 (2.5) (2.2)	0.0 (2.9) (3.3)	2.1 (2.9) (3.2)	2.0 (2.0) (2.7)	2.2 (2.2) (2.9)	0.0 (2.2) (3.3)	0.0 (2.9) (3.0)	2.1 (2.4) (2.6)
1st Quad	3.4 (3.5) (3.4)	3.3 (3.3) (3.3)	3.2 (3.3) (3.3)	3.2 (3.2) (3.2)	3.4 (3.4) (3.4)	3.4 (3.4) (3.4)	3.4 (3.4) (3.4)	3.4 (3.3) (3.4)
Median	3.5 (3.5) (3.5)	3.4 (3.4) (3.4)	3.3 (3.3) (3.3)	3.3 (3.3) (3.3)	3.4 (3.4) (3.4)	3.4 (3.5) (3.5)	3.5 (3.5) (3.5)	3.4 (3.4) (3.4)
Mean	3.48 (3.80) (3.66)	3.40 (3.43) (3.38)	3.29 (3.81) (3.31)	3.35 (3.85) (3.43)	3.49 (4.01) (3.50)	3.46 (4.13) (3.45)	3.51 (3.76) (3.47)	3.79 (4.28) (3.46)
3rd Quad	3.5 (3.6) (3.5)	3.4 (3.4) (3.4)	3.3 (3.4) (3.3)	3.3 (3.3) (3.3)	3.5 (3.5) (3.5)	3.5 (3.5) (3.5)	3.5 (3.5) (3.5)	3.5 (3.5) (3.4)
Max	44.2 (8.1) (17.9)	48.4 (8.0) (3.5)	18.8 (18.2) (3.4)	22.0 (10.9) (8.1)	73.0 (24.2) (16.1)	32.4 (32.4) (3.5)	24.7 (8.1) (8.0)	24.5 (24.5) (8.1)

Table 5.2: Selected group A switches (see chapter 2.1.2) and their summary statistics of *motor on* times over the log collection period. The table includes the minimum/maximum, first, second and third quantile and mean values of the measured *motor on* times. The calibration constant is calculated as the 1st quadrate point of measurements collected 0–150 turns after maintenance calls.

5.2.1 Trends in the delays

In each of the eight selected railway points, the time series plots 5.1 and 5.2 reveal some vertical trends. These trends can mean periodical maintenance breaks or any testing period of a railway point. Information on those events is necessary when constructing the online predictions. However, this was not available and confirms our conclusion that we have incomplete maintenance data. Thus, vertical trends were not studied further, but would be relevant in discussions with the maintenance company.

Figures 5.1 and 5.2 show the trends, but the widely varying y-axis limits the comparison. To see trend-related details, the main points of the figures are collected in 5.3. The distribution is first divided into four areas: the main area is centred around the medium value, there is an area before the main interval, an area after the main interval but below 12 secs and an area after the main interval but over 12 secs. The high percentage of results in the "over 12 secs" category for *monitor off* times tells us that there is most likely a high number of vertical trends, which can be confirmed from the

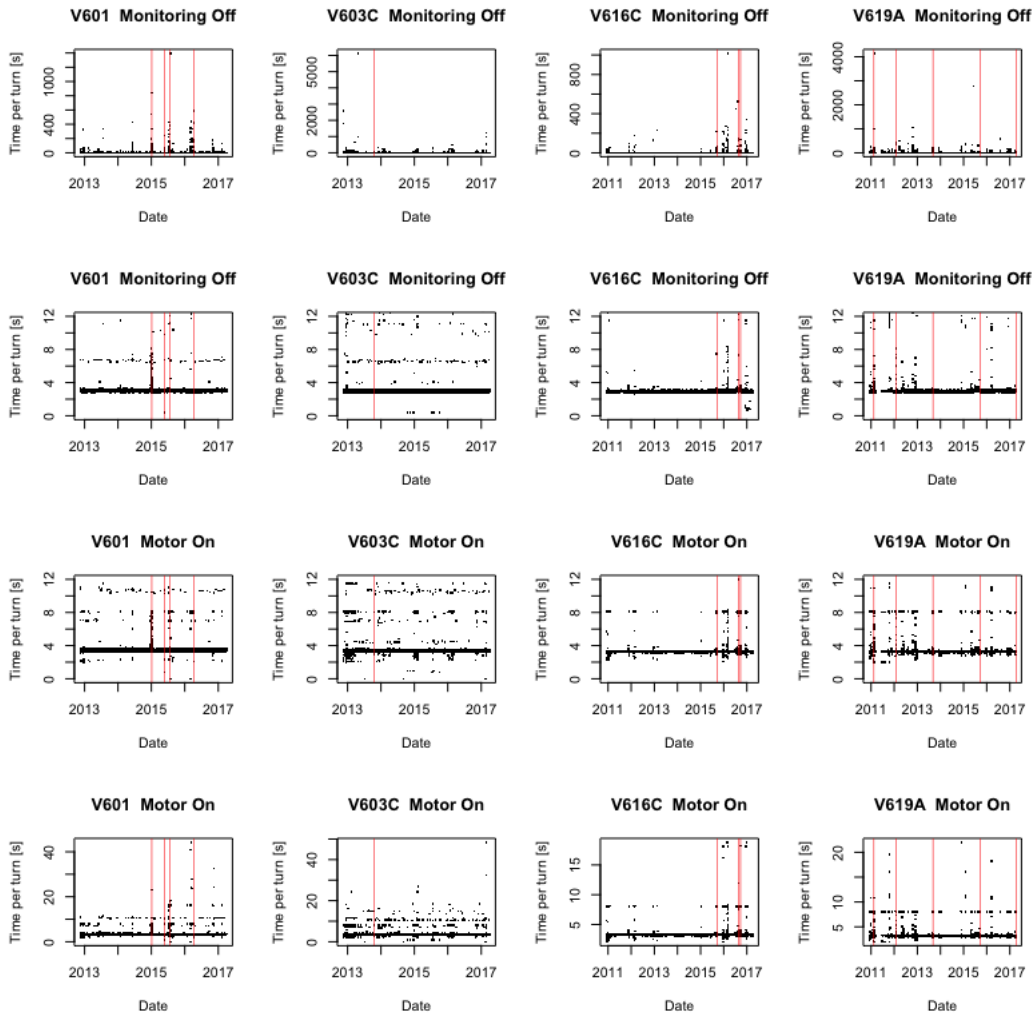

Figure 5.1: All measurements from the selected switches: part one. Each switch has four figures below each other, two for both measures and additionally two focused on the situation below 12 secs. The y-axis has the time that monitoring is off or that the motor is on, calculated from the log file. The x-axis includes the date and the vertical lines are the maintenance calls that we have information and should have an impact on the measurement times.

figures. "Motor on" (E in the table) times also exceed 12 secs on average 4/10000 turns – the reason for that is assumed to be some manual usage of the engine. The "main" and "before" intervals are not that interesting but are included for completeness. From horizontal trends perspective, the

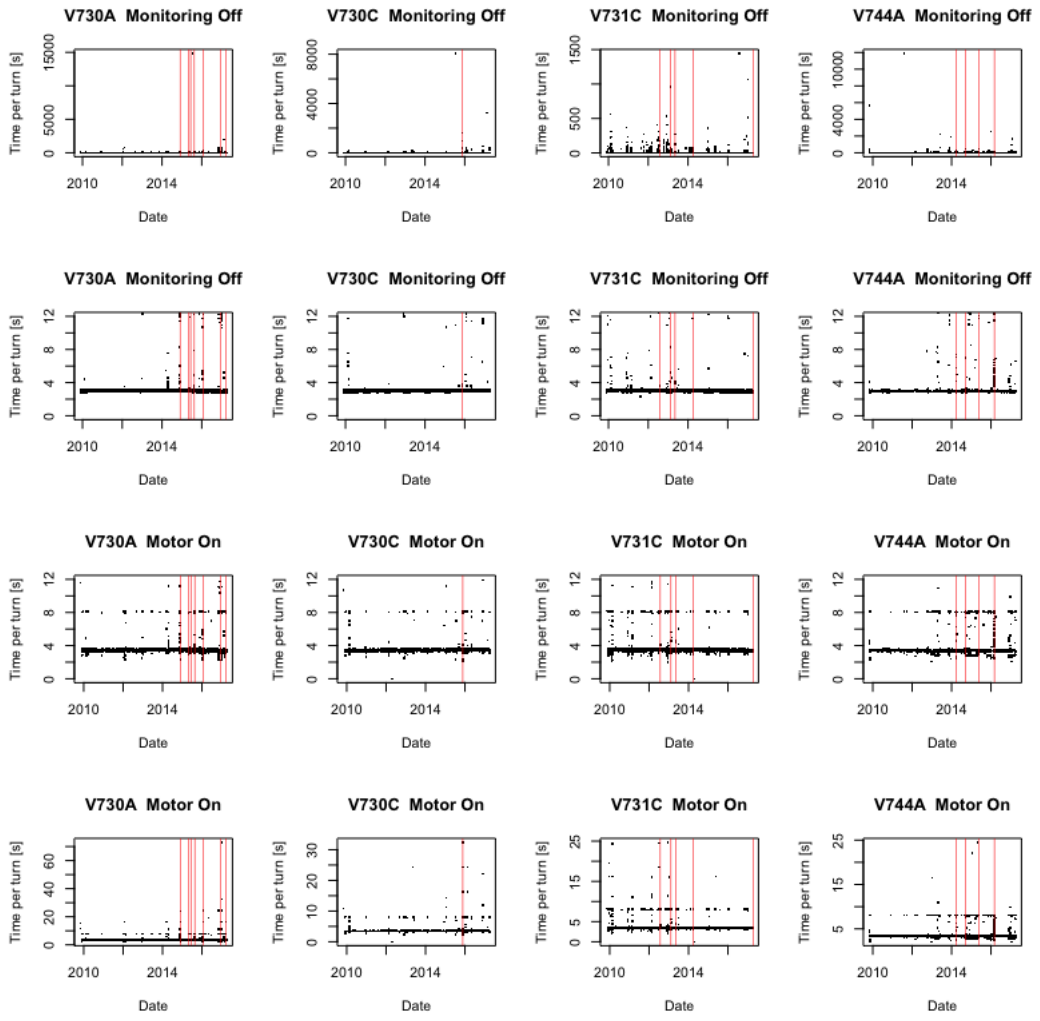

Figure 5.2: All measurements from the selected switches: part two. Each switch has four figures below each other, two for both measures and additionally two focused on the situation below 12 secs. The y-axis has the time that monitoring is off or that the motor is on, calculated from the log file. The x-axis includes the date and the vertical lines are the maintenance calls that we have information about and should have an impact on the measurement times.

interval between typical measurements in "main" and "before 12 sec" is the interesting interval. This interval is further divided into four distinct time intervals ($H1 - -H4$) to roughly catch possible horizontal trends. The numbers for H values are proportions of the data in the "after" interval. The $H3$

trend is clear and found to be a trigger system in the engine that can be seen as a critical error in log files. For switches $V601$ and $V603C$ the *monitor off* times have clear trends during the time interval H2–H4 no clear explanation has been found for that yet. Other exceptions in *monitor off* times are in $H4$ where the exceptions are a reflection of vertical trends (except in case $V603C$). This exception has no clear explanation either.

RW Point,	BF	Main	After	After 12	H1	H2	H3	H4
V601 M	1.1E-3	99.3	0.2	0.4	7.1	36.1	0.5	3.6
V601 E	0.15	99.6	0.6	0.03	4.4	6.2	68.9	8.6
V603C M	5.4E-3	99.5	0.1	0.4	5.9	53.3	0.0	20
V603C E	0.28	99.5	0.5	0.04	0.7	6.3	70.9	14.5
V616C M	28.3E-3	99.4	0.2	0.4	7.1	4.3	4.3	7.1
V616C E	0.17	99.6	0.5	0.02	5.6	1.4	70.4	0.0
V619A M	0.0	98.2	0.4	1.4	7.3	4.1	2.4	16.3
V619A E	0.44	99.2	1.8	0.02	2.8	0.9	81.5	2.4
V730A M	0.0	99.0	0.1	0.9	6.0	0.0	2.0	16.0
V730A E	0.25	99.6	1.0	0.04	1.1	1.1	89.1	1.8
V730C M	0.0	99.6	0.1	0.3	6.2	9.4	0.0	21.8
V730C E	0.07	99.9	0.4	0.02	4.1	2.6	87.6	1.0
V731C M	2.2E-3	98.9	0.1	1.0	10.4	10.4	2.1	4.2
V731C E	0.53	99.3	1.1	0.03	1.2	1.2	89.2	2.3
V744A M	0.0	90.3	1.8	7.9	16.7	12.1	1.5	10.6
V744A E	5.4	92.8	10.3	0.08	1.1	2.1	84.6	0.3

Table 5.3: The distribution divided intervals the main (within 0.4 sec from medium), "BF" is "before main", "after" is "after main" but smaller than 12 secs and "after 12" more than 12 secs. The horizontal trend is expressed as a percentage of observations between "main" and "12 secs". That includes four intervals: H1 (i.e. 4–4.2 secs), H2 (i.e. 6.5–7.2 secs), H3 (i.e. 7.8–8.2 secs) and H4 (i.e. 10.2–11.5 secs).

5.3 Predictability

In this section, the main results of the visual study of predictability are shown. From this point on the studies concentrated on data that was collected in the close vicinity of the selected maintenance cases. The data before and after the maintenance cases are compared to see whether they differ so significantly that it can be assumed that they come from different distributions.

Based on the summaries in the previous section, *motor on* and *monitor off* measures might have long tails in their distribution, this makes the selection of scales in the visualisation difficult. However, since software component trigger alarm before the *motor on* time reaches 8.5 secs and typically shuts the engine, higher engine times can be seen as problematic cases. Similarly, *monitor off* times that are over 12 secs indicate some problem and another trigger is recorded on the log. Thus, these cases are assumed to be included as "critical alarms" and for visualisation all *motor on* and *monitor off* measurements that are more than 8.5 or 12 sec are marked with those limiting values. Q–Q plots are a robust visual method for seeing the similarity of two different empirical distributions. These plots are first utilised with the measurements from each switch and then for a combination of data from all switches before and after the maintenance. These figures show that there is a difference in distributions, but also a clear overlap in individual measurements. Thus, consecutive measurements are collected together and characterised with seven features whose properties are visualised with histograms. The effect of the distance from maintenance is also clarified via histograms, and lastly the bivariate correlation of the sample characters is enlightened with a pairwise plot.

5.3.1 Comparing event distributions

The tables 5.2 and 5.1 show that a large portion of the distributions before and after maintenance is located in the same small main area. Thus, it is expected that most of the quantiles are close to the $y = x$ line when the value is smaller than 3.5, but that a difference would occur after that. However, more samples give a better view of these rare quantile points, thus the combined case gives a better view of the differences between distributions.

From figure 5.3 we can verify previous expectation. All of the distributions differ in their biggest quantiles and overlap with their smaller quantiles. The number of tail quantiles in which the difference can be seen varies between cases, but all of them are clearly different when all values from a distance of 150 events are collected together. For completeness figure 5.4 shows the same statistics for the *motor on* times. One clear difference is that that there can also be a difference between distributions in the small delay values. However, the difference in the maximal values is more appreciated, since the small values are due to the restart of an engine after some trigger has required it (at around 8 sec). The difference is as clear in the same switches as it is in the *monitor off* case.

Both of these measures indicate that median values before and after maintenance are similar, thus quantiles close to the median are not very good

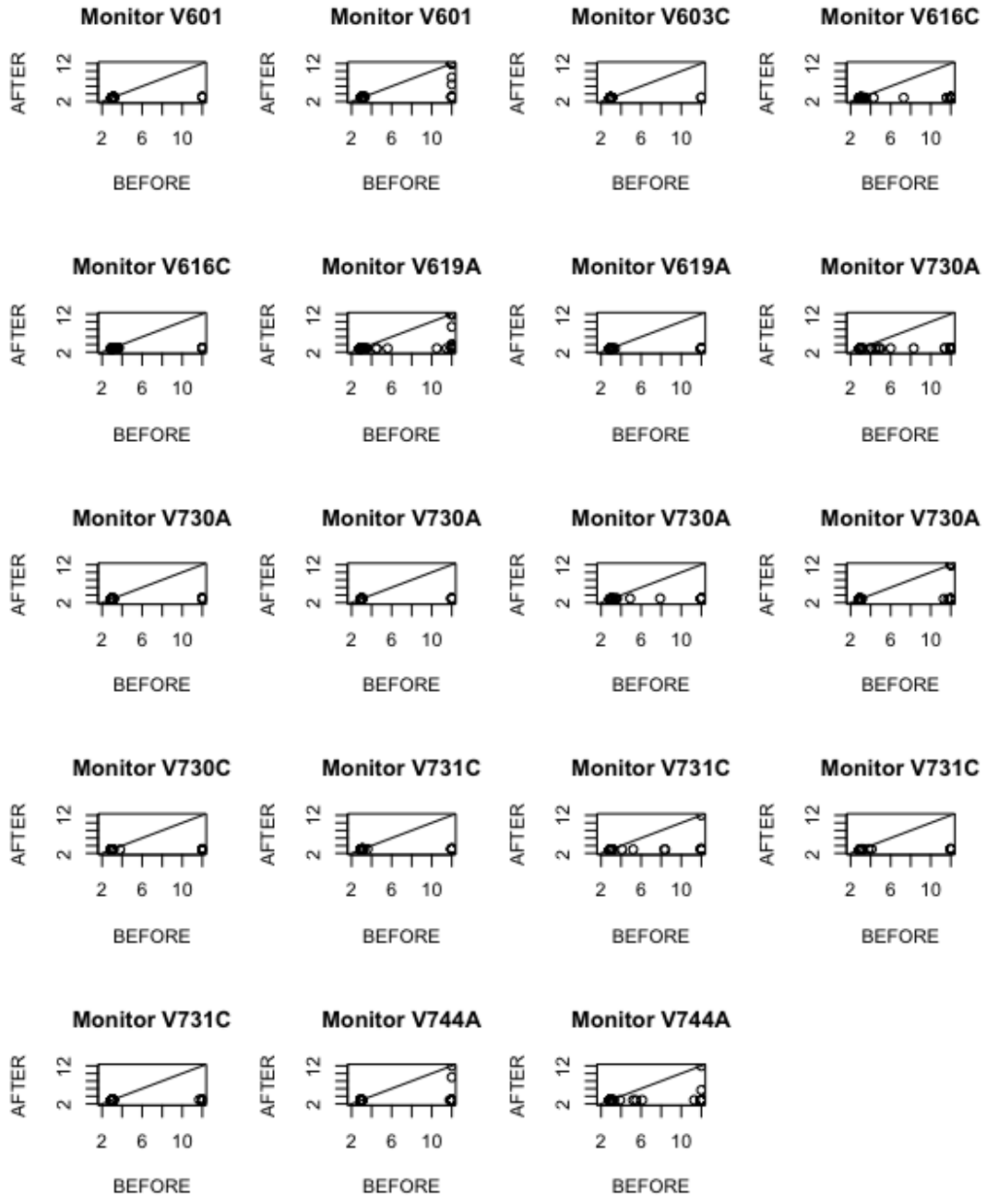

Figure 5.3: The Q–Q plots of the *monitor off* measures of different switches before and after maintenance. The measurements that are over 12 secs are replaced with "12 secs".

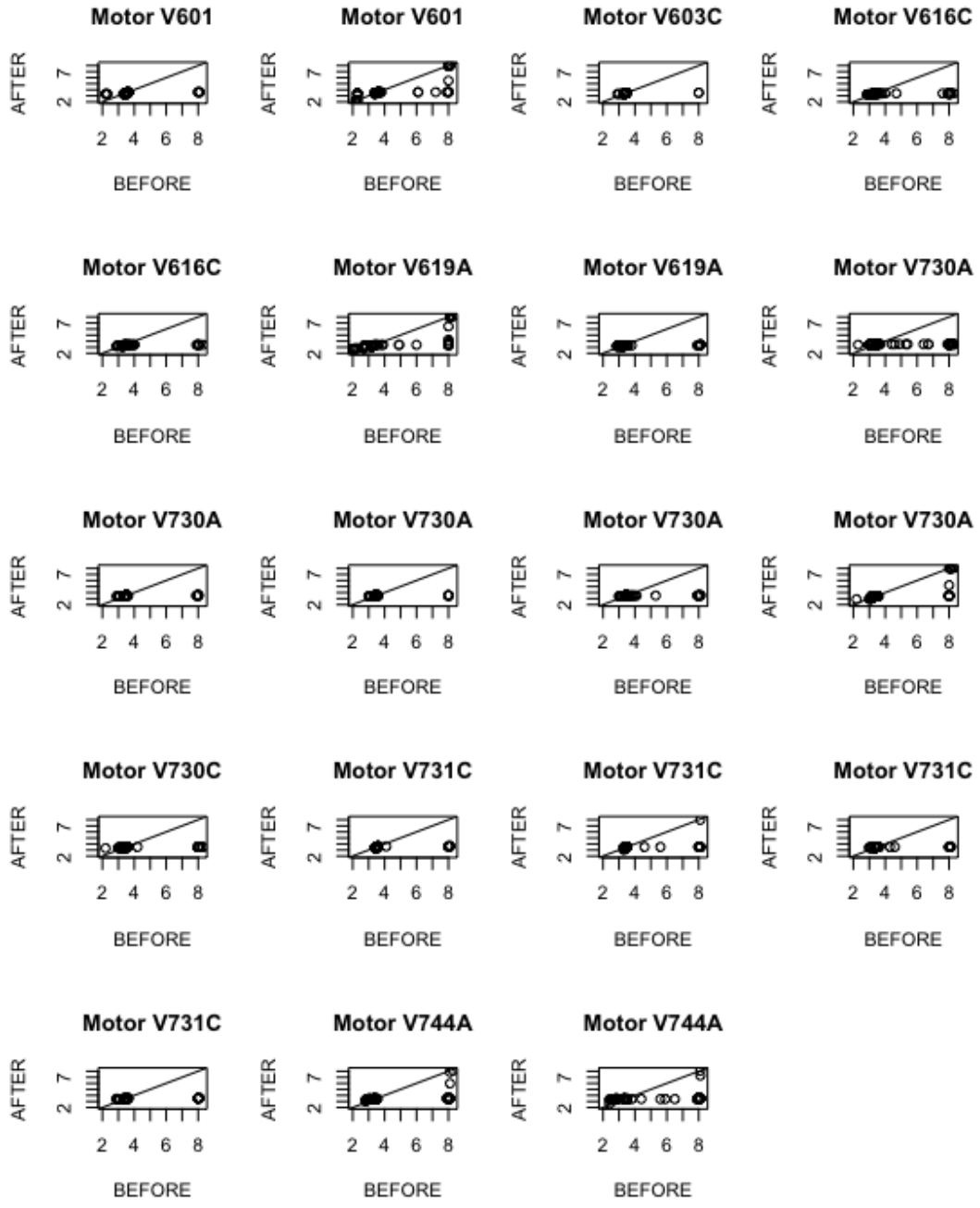

Figure 5.4: The Q–Q plots of *motor on* measures for different switches before and after maintenance.

characteristics with which to separate these distributions when compared to

the maximum value or the spread of the distribution. It is also clear that if a sample were to include the time period of 150 events, it should be possible to separate the two distributions. Combining measurements from the different switches confirms earlier findings (see figure 5.5). From the figures it is clear that distributions are different, however they have plenty of overlapping samples values. The figure shows that the distributions can be separate in the spread of different values and the amount of "maximum" values. These forms the basis for the sample characteristics.

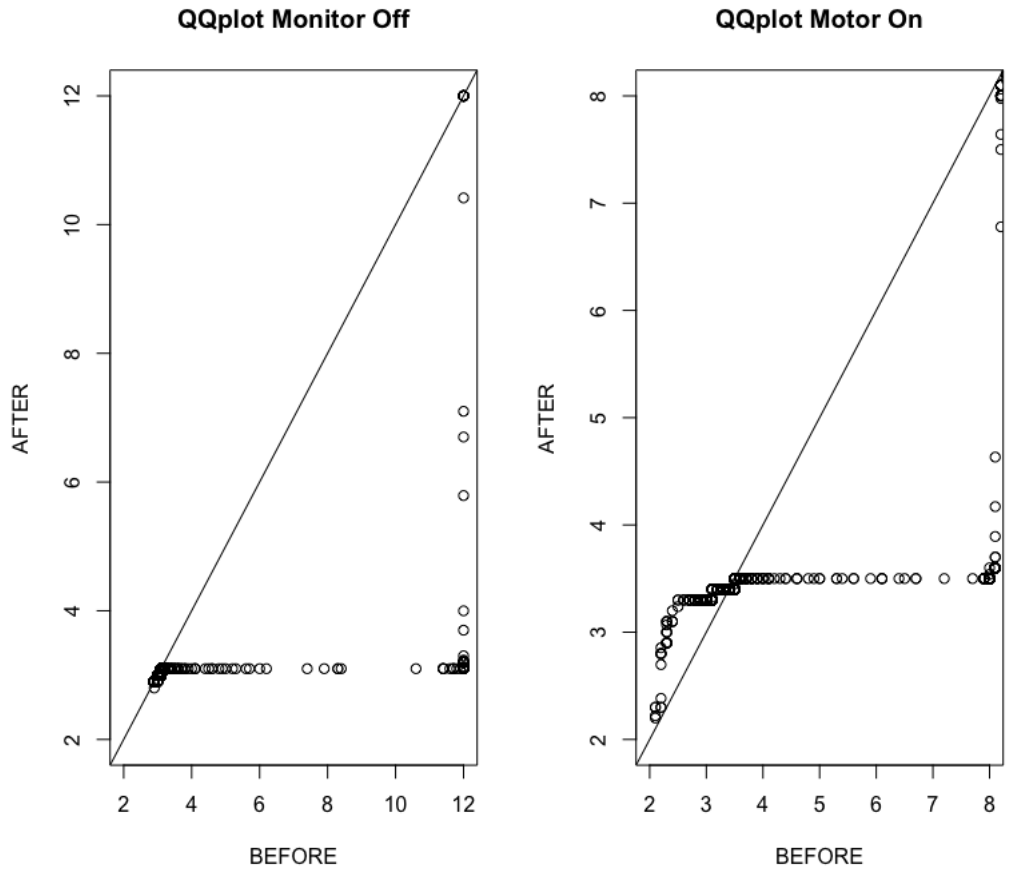

Figure 5.5: The Q–Q plots of *motor on* measures for different switches before and after maintenance.

5.3.2 The properties of the sample characteristics

The sample characteristics are the ones we are using in our testing models, thus it is important to see how different they are before and after samples. Moreover, they have two important parameters (i.e. sample size and location) whose behaviour needs to be understood. The assumption of the independence of samples implies that location does not matter. The assumption, generally not true, but allows us to study the predictability and prediction accuracy. However, it might be true in close vicinity of maintenance call and with histograms we can study that visually. The sample size is assumed to have an impact to the accuracy of the prediction model, thus at this point it is not considered further. There is one fixed sample size (15 events) in histograms and other sizes are considered with prediction models. The sample characteristics are described in section 2.3.1.1.

For figure 5.6 two characteristics were chosen to study three phenomena: whether distribution after the maintenance stays similar in time, how long is it before distribution changes as a function of location and how much overlap there is between the "before" and "after" characteristics' distributions. The stable distribution after maintenance seems to be true, even if there is slight difference in distribution immediately after maintenance. Thus, in relation to that, the independence assumption holds. "Before" maintenance distributions have a clear difference between "0–30" and other distributions, and the change is quite slow after that until the 150-event limit we have selected. Similarly, the "after" and "before" distributions get closer to each other the further the location from maintenance (as they should). The overlap is present in each location, however there is still slight difference, even in "120–150" distributions. This is studied further in figure 5.7.

The characteristics $X1 and X4$ behave quite similarly, as do $X2 and X5$ (see 5.8), thus the study continues with the characteristics $X3, X6 and X7$. Figure 5.7 shows the distribution change of the characteristics at both ends of the 150-event collection period. It confirms that even in further locations the distributions before and after maintenance are different, but hugely overlapping. The overlap is roughly 75 percent of sample characters, whereas just before maintenance the non-overlapping part is roughly 75 percent with one characteristic. Figures 5.6 and 5.7 show that $X3 and X6$ are the best individual charters with which to make predictions. Also, the histograms of sample size 30 show that roughly 50 percent of the samples could be predicted correctly with only one characteristic. However, for half of the samples the results are overlapping. Thus, to improve the prediction accuracy, one indicator might not be enough and a combination of factors is required. The more independent the characteristics are, the better the multivariate prediction usually

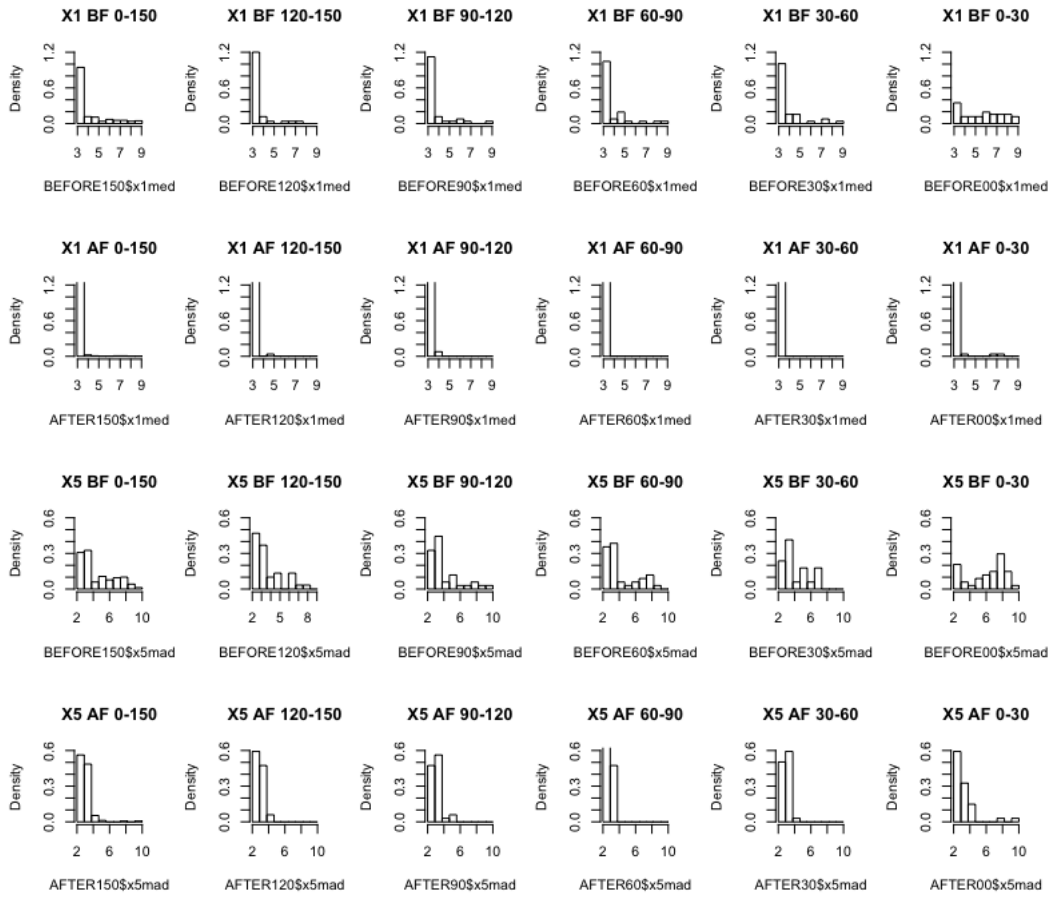

Figure 5.6: The histograms of sample characteristics with equivalent bin widths and limited y-axes, separated for both characteristics. X1 is the mean of "monitor off: times and X2 is the number of different values in *motor on* times with a sample size of 15.

is. The bivariate analysis in figure 5.8 gives information on the correlation between two characteristics. Since bivariate analysis contains some similar information to histograms, the sample size is changed to 30 events and all data inside 150 events from maintenance are utilised. Similar conclusions to those from histograms can be made with bivariate analysis, with the addition of pairwise correlation. There is clear linear correlation between X1, X4 and X7 (see 5.8). It also shows that no matter which two variables are selected, some cases are difficult to separate. The same conclusion can be made using hierarchical clustering and all characteristics. However, figure 5.8 also gives hope that not all characteristics are fully dependent and the combination of

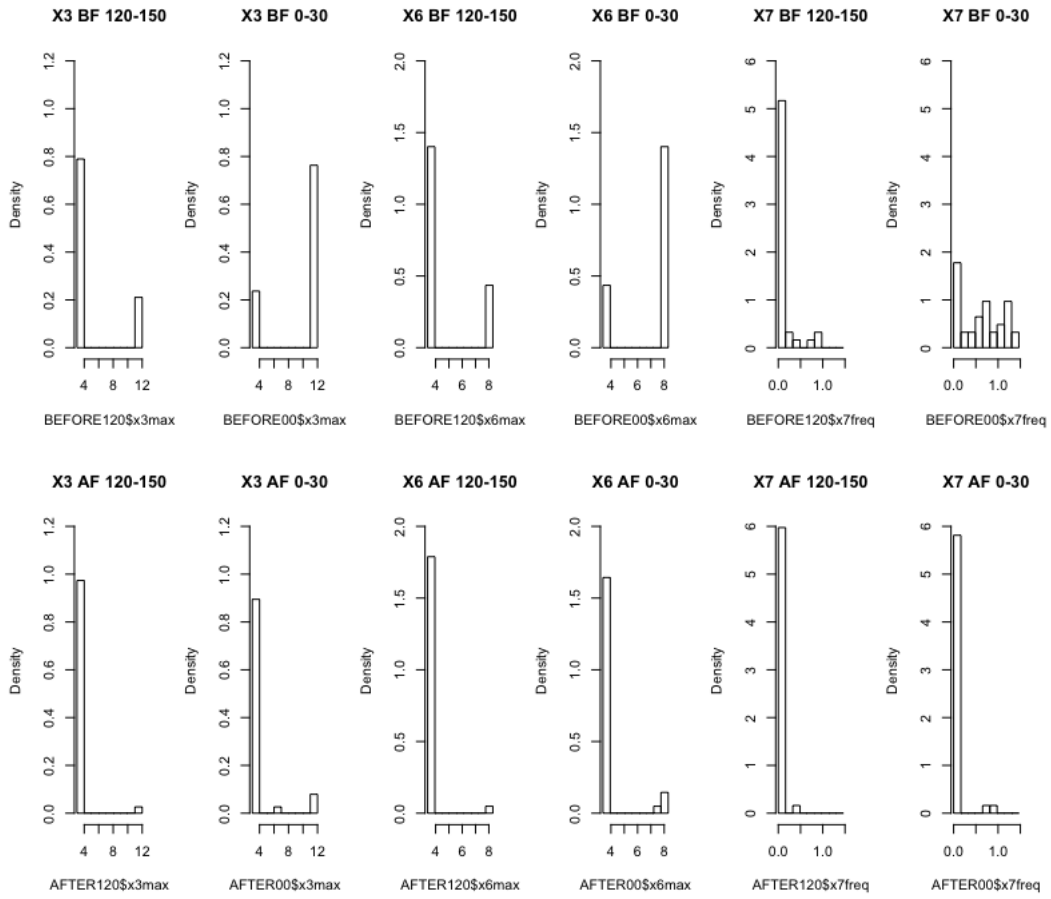

Figure 5.7: Histograms of sample characteristics with equivalent bin widths and limited y-axes, separated for each characteristic. X3 and X6 are the maximum *monitor off* and *motor on* times with a sample size of 15. X7 represents the frequency of critical errors reported in the log file.

them could improve the predictions.

5.4 Predictions

5.4.1 Feature selection and training accuracy

Table 5.4 shows that the optimal feature set depends on both sample size and prediction model. The maximum *motor on* time is the only feature that is present in each of the combinations. Besides, the most important feature is $X6$ – this makes it difficult to select the best variables for implementation.

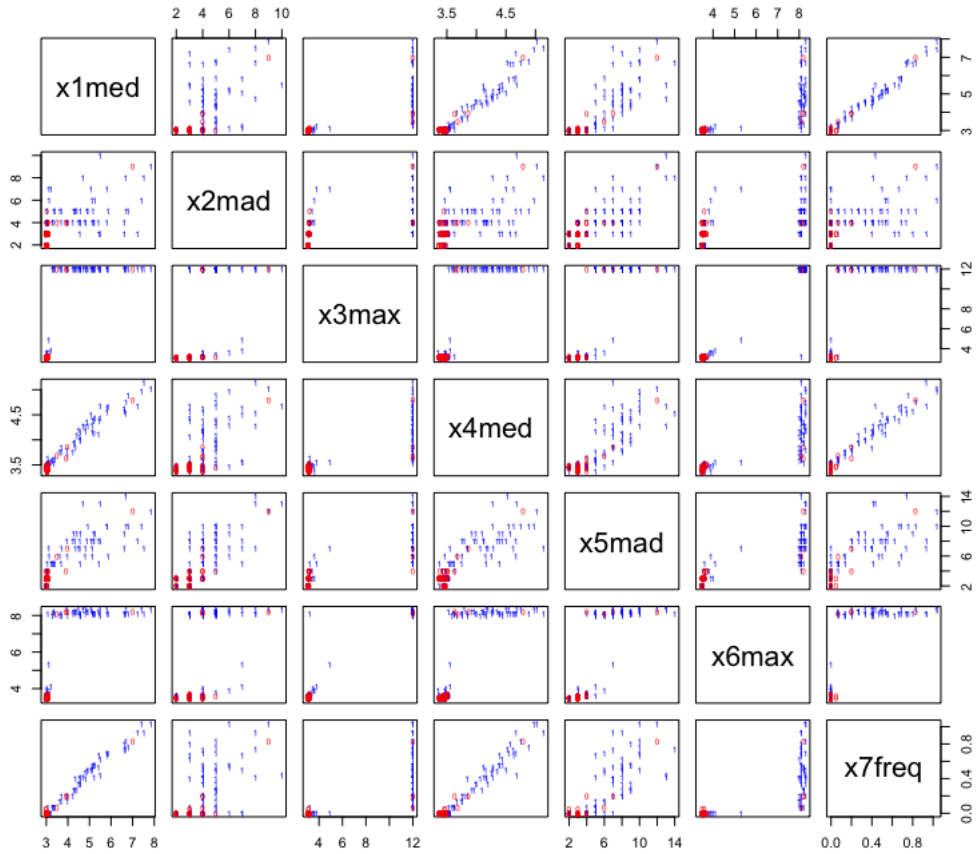

Figure 5.8: Bivariate analysis of sample characteristics when sample size is 30 and distribution width is 150 events. Values after the maintenance are marked with a red "0" whereas values before it is marked with a blue "1".

The reason could be also being the number of examples or very correlated variables.

Figure 5.9 shows that the suggestions given by the algorithms are vague in the sense that most of the variable combinations give similar prediction accuracy. This means that a variable combination is not expected to improve the prediction very much and almost the same results could be achieved with simple threshold of $X6$. The most obvious explanation for this is that all the variables are highly dependent. However, we use the variable combination,

[1]For this table, a logistic regression model with an elastic net is fitted to the whole data and the number of active variables is given for the reference. For this model, implicit feature selection from all available features happens each time the model is constructed.

Sample Size Prediction Model	Size: 30	Size: 20	Size 15	Size: 10
Logistic Regression	3: X6, X3, X4	3: X6, X3, X4	5: X6, X5, X4, X1,X2	1:X6
Log. Regr. EN[1]	3: X6, X3, X5	3: X6, X3, X5	3: X6, X3, X5	3: X6, X3, X5
SVM	4: X6, X4, X5, X3	6: X6, X4, X5, X3, X1, X7	6: X4, X6, X5, X3, X7, X1	6: X6, X4, X5, X3, X1, X7
Naive Bayes	3: X6, X4, X5	5: X6, X4, X5, X3, X1	2: X4, X6	6: X6, X4, X5 X3, X1, X7
Random Forest	7: X1, X5, X6 X4, X3, X7, X2	2: X6, X3	3: X6, X5, X4	5: X6, X3, X4 X5, X7

Table 5.4: The proposed feature selection of the implemented algorithms for variable sample sizes.

as suggested by the feature selection tools [2].

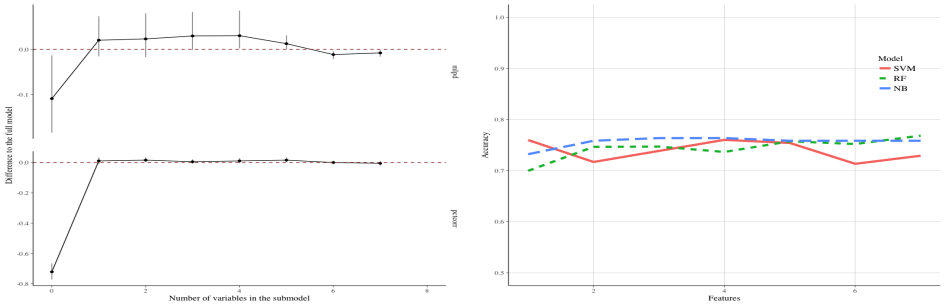

Figure 5.9: Feature selection with sample size 30. On the left is a logistic regression with relative mean log predictive density (MLPD) and mean model accuracy as a function of the selected variables, as calculated by variable selection implementation. On the right is the mean model accuracy of the SVM, random forest and naive Bayes models.

5.4.1.1 Training accuracy

Training accuracy shows the limits of the selected prediction model with the given data. For several model types this gives a meaningful upper bound to the prediction accuracy, however, for decision trees without feature selection for example, it is rather useless. It is clear that at least random forest can be overfitted to any training data.

[2]The caret recursive feature elimination (RFE) function always suggests the smallest number of features if the same results are achieved with a higher number of features.

Statistics	Logistic Regression	LR. EN	SVM	Naive Bayes	Random Forest	Comb.
S10: 0–150	0.65	0.67	0.73	0.66	0.76	0.66
S15: 0–150	0.70	0.69	0.69	0.67	0.75	0.69
S20: 0–150	0.71	0.72	0.72	0.71	0.76	0.71
S30: 0–150	0.74	0.75	0.77	0.74	0.94	0.75
S10: 0–60	0.74	0.74	0.75	0.73	0.87	0.74
S15: 0–60	0.76	0.74	0.77	0.74	0.85	0.74
S20: 0–60	0.81	0.79	0.86	0.79	084	0.80
S30: 0–60	0.86	0.83	0.88	0.84	0.97	0.84
S10: 60–120	0.61	0.62	0.62	0.61	0.75	0.62
S15: 60–120	0.68	0.64	0.64	0.64	0.76	0.64
S20: 60–120	0.65	0.64	0.72	0.65	0.70	0.65
S30: 60–120	0.74	0.68	0.75	0.68	0.95	0.68

Table 5.5: The training accuracy of the selected models with the given parameters and proposed feature selection. The accuracy is given in three different distribution locations.

Table 5.5 shows two important issues about data quality and sample size. The effect of size on the best achievable prediction accuracy quantifies the improvements that increased "history information" brings to the prediction. This is why sample size 30 is selected for the further analysis. The other more important issue is the data quality. All the prediction models, except random forest, have limited training accuracy of around 75 percent. This hints that the data quality will not permit better predictions. This might improve with more accurate timing measures, but this is not certain since the related literature only describes results that contain total power measurements.

5.4.2 Estimated prediction accuracies

The bigger the sample size, the better the training accuracy (as shown in table 5.5). Thus, the predictions accuracies (i.e. out-of-sample estimates) are calculated with sample size 30. The accuracies are calculated from samples whose closest event from the maintenance is reflected in the x-axis of figure 5.10. It shows that the feature selection smooths the prediction performance of logistic regression that otherwise, without regularisation, tends to overfit to the training data, leading to slightly worse prediction accuracy. It also shows that for random forest the performance clearly drops from the training accuracy. However, the major issue in the figure is the trend showing how prediction accuracy changes when we approach the maintenance event. The event that causes the maintenance call is always neglected in the data.

Figure 5.10 reveals that even when the trend in predictions of all models

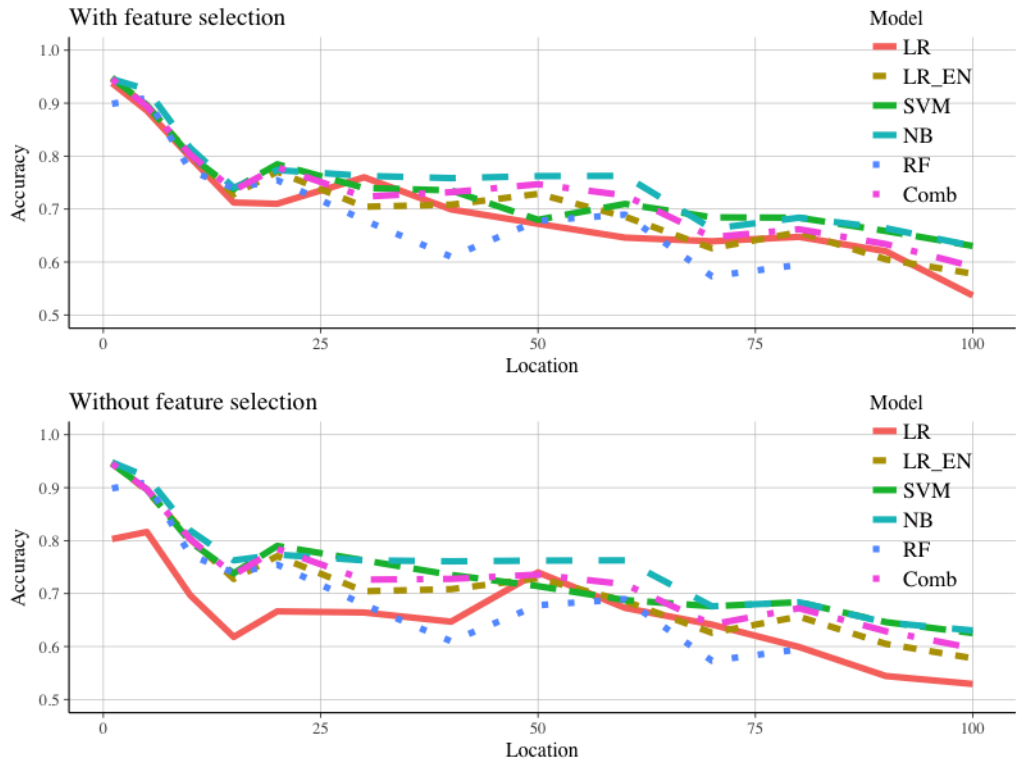

Figure 5.10: The mean prediction accuracy of the samples collected at different distances from maintenance. In the upper figure models use the proposed selected features whereas below all the variables are used in the models. Sample size is fixed to 30 events.

is similar, there are slight differences in prediction accuracies. Surprisingly, with this data naive Bayes estimates provide some of the best average performances. The prediction accuracy is over 60 percent already 100 turns away from the maintenance call and exceeds 70 percent around 60 turns away. Then, the performance is quite steady until the accuracy increases to over 80 percent around 10 turns away and over 90 percent five turns away. These results show that with these models and parameters there is a possibility to get max 80–90 percent accuracy when predicting failure just before it happens. However, these results are based on a very limited data set, thus we should be careful about drawing strong conclusions. Error analysis could give more insight to the usefulness of these results and possible action strategies, but before that we summarise the prediction accuracies with different sample sizes to table 5.6. Table 5.6 and figure 5.10 show that the prediction accuracies are similar to the training accuracy for all models except random forest.

Statistics	Logistic Regression	LR. EN	SVM	Naive Bayes	Random Forest	Comb.
S10: 0–150	0.65 ± 0.03	0.66 ± 0.03	0.66± 3	0.66 ± 0.03	0.69 ± 0.04	0.66 ± 0.03
S15: 0–150	0.69± 0.04	0.68 ± 0.05	0.68 ± 0.04	0.67 ± 0.04	0.68 ± 0.05	0.69 ± 0.05
S20: 0–150	0.70 ± 0.05	0.71 ± 0.05	0.71 ± 0.05	0.71 ± 0.05	0.75 ± 0.06	0.71 ± 0.05
S30: 0–150	0.75 ± 0.05	0.74 ± 0.06	0.74 ± 0.06	0.74 ± 0.06	0.76 ± 0.06	0.74 ± 0.06
S30: 0–120	0.78 ± 0.06	0.76 ± 0.05	0.76± 0.05	0.76 ± 0.05	0.78 ± 0.06	0.76 ± 0.05
S30: 0–90	0.80 ± 0.09	0.79 ± 0.08	0.79 ± 0.08	0.80 ± 0.08	0.80 ± 0.07	0.80 ± 0.08

Table 5.6: The out-of-sample prediction accuracy and standard deviation of different sample sizes. These are calculated with five-fold CV with 10 random repetitions. On the bottom are two additional distribution widths for sample size 30.

That gives us reason to believe that the prediction accuracies are limited by the data quality and there is no reason to try to find better models for this data. From table 5.6 we can also conclude that the accuracy of all the proposed models is at the same level.

5.4.3 Prediction error analysis

The cost of an error often depends on the error type. In case of two categories these can be shown in a contingency table (see 3.3). The false positive rate, a.k.a. false alarm, is the probability of predicting a maintenance request in no-maintenance-request-needed cases, whereas the true positive rate tells us that probability among maintenance-needed cases. The ROC curve plots these probabilities with different decision boundaries and is shown in figure 5.11. On the left of the figure all repeated and cross-validated cases are included in the same figure, whereas on the right the average is plotted with the standard deviation (STD) for different cut-off values. The STD values both in x and y values are bigger in the small values and highs for the random forest. Moreover, the right side shows that the random forest is the only clearly different profile predictor, even with the combined data, and also hints that logistic regression could have a slightly different profile.

Error analysis is summarised in table 5.7. The AUC value on the left corresponds the left side of figure 5.11, whereas the distribution values on the right correspond to the averaging done on the right of the figure. The AUC value summarises the ROC curve, in other words the models' ability to rank true positive (maintenance-needed) cases relative to the negative instances. This means that the true positive values correspond to maintenance predictions cases with a high probability. In essence, the AUC is the probability that model will rank a randomly chosen positive instance higher than negative. The ROC curve shows the model's performance at different probability

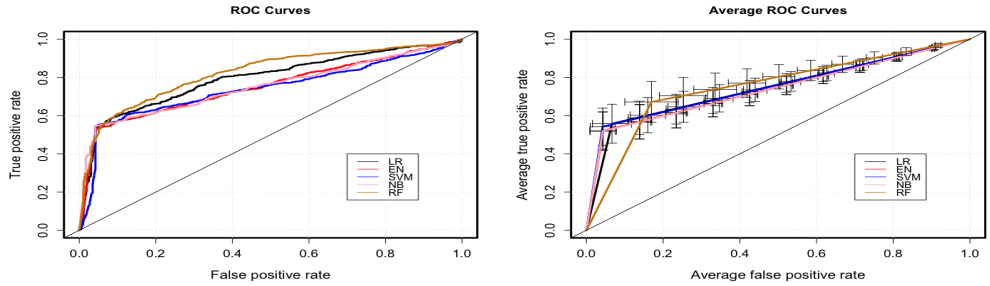

Figure 5.11: ROC curves combined from five-fold CV with ten repetitions. On the left is the ROC from the combined probability statistics, whereas on the right each case is handled separately in order to get the STDs that are added with a mean value line to the figure. The data collection width was 150 events and the sample size was 30.

points.

On average random forest is a slightly better classifier for ranking positive instances relative to negative instances than other classifiers. Whatever they are based on, the produced distribution values the order might vary based on the training–testing split of the data. The precision values tell that among positive predictions over 90 percent are correctly predicted for lower AUC value models, whereas for random forest the value is 80 percent. On the other hand, from all true positive values less than 55 percent are predicted correctly (recall), whereas for random forest the percentage is over 65 percent. The "specificity" column shows the true negative ratio among all negative instances and random forest is clearly the worst among the compared models. The accuracy of the predictions seems to be at the same level for all models and a typical error is a false negative prediction (i.e. a prediction that there is no need for maintenance even when the data is collected before a maintenance call). The proportion of that type of error is smaller in the random forest case. This error type could be due to having a too wide distribution width – even figure 5.10 and histograms indicate that the change might not be that dramatic. The proportion of error types FP/FN remains a constant 0.15, 0.5 or 0.15 for logistic regression, random forest and others with different distribution widths 150–90, after that error type proportion of other than random forest increases.

Measure Model	AUC	Precis.	Recall	Specif.	$\mu_{Acc} \pm \sigma_{Acc}$	$\mu_{AUC} \pm \sigma_{AUC}$
LR	0.79	0.89	0.56	0.93	0.75±0.08	0.81±0.07
LR EN	0.75	0.92	0.52	0.96	0.74±0.08	0.79±0.08
SVM	0.74	0.93	0.54	0.96	0.75±0.08	0.79±0.08
NB	0.75	0.92	0.52	0.96	0.74±0.08	0.79±0.07
RF	0.82	0.80	0.67	0.67	0.75±0.07	0.83±0.07

Table 5.7: A summary table of model measures and error analysis with a distribution width of 150 events and a sample size of 30. The four columns on the left are produced from combined data. A cut-off of 0.5 of the score functions is used to calculate precision, recall (true positive rate, sensitivity) and specificity values.

Chapter 6

Discussion and Conclusions

This thesis has presented the creation challenges of a data mining project in a fresh environment. It utilised an existing combination of maintenance information and railway signalling logs, and showed the limits in predicting the instantaneous need for maintenance in three major failure categories of railway points.

A data mining project's outcome should be more than the final results; the understanding gained during the process can be almost as valuable and the problem might even change due to intermediate results, as was the case in this thesis. To emphasise this and improve success possibilities of new data mining project on this topic in similar conditions, three work aspects are highlighted in the thesis. The first aspect is the data mining working procedure and possibilities to enhance it. This was studied with the help of CRISP-DM reference model. The second aspect is data quality, its limitations and possibilities to improve it. The thesis goes into the origins of the data sources in order to seek the needed actions to improve the quality and construct the correct data format to build a real-time maintenance planning assistance application. The third aspect is the main problem: the feasibility analysis of the predictability of the railway point failures. The thesis followed order of working procedure and visualised the initial situation in time series plots of the selected measures (the *motor on* and *monitor off* times, calculated from log files). In the analysis of this thesis it is assumed that *turns* in close vicinity are *independent* and turn behaviour is visualised before and after selected maintenance calls using Q–Q plots. To improve the predictability in this thesis, consecutive turns were combined in samples and characterised them. It used bivariate analysis to revel their properties and showed model-specific feature set selection. It also showed the achievable prediction accuracies, error types and dependency on the distance from the maintenance call using four different prediction models (*logistic regression*,

naive Bayes, *SVM* and *random forest*). The thesis includes the history, construction principles and R language implementation of these models. The highlights and the most important findings, and improvement suggestions for all three phases are included below. The working procedure's description can serve as a pre-planning step and a tool to achieve the correct participation, resourcing, better result utilisation, improved problem description and harmonised understanding of results. Thus, explaining the data mining work phases to all stakeholders at the beginning of the project has great potential to improve the success of the work. In our project, the first two phases were found to be the most critical and engaging the correct experts at that time could have potentially improved the work plan, the problem description and the utilisation of intermediate results inside the company. It is especially important that persons understand the environment and the way potential prediction results would be used. The other aspect of the work description that would have helped relates to tool selection – in our case it would have allowed usage of favourable tools in the initial analysis and sped up the definition of the final target. Moreover, it would have indicated the need for further co-operation and data inquiry from the maintenance company. The final tool and programming language are most likely different to those utilised in feasibility analysis. The bridge from procedural issues to data-related issues is that the understanding of working phases could have hastened understanding of the log data collection process. In this project, the details of the time stamp were clarified as a consequence of the analysis of differences in distributions between switches.

The data quality, format, collection procedures and improvements in them are important for understanding the results and enabling fluent future work. This thesis lists improvements for all of them. Understanding all maintenance-related actions, the reason for doing them, their cost and the manner of reporting them would have improved the correct targeting of the work and the full utilisation of log data. During the project a combination of maintenance regulation and log file analysis revealed the incompleteness of the information. Moreover, in order to build any real-time predictive algorithm to assist repairmen in work planning, clear categorisation of problems and actions would be needed in the maintenance data in order to specify algorithm behaviour after each failure. The incompleteness of that data was the major reason that we could not evaluate the usability of the sliding window approach of our prediction models with full log data. The limited understanding of maintenance work made us concentrate on common, possibly slowly developing problems listed in group A that required the *tuning, cleaning* or *oiling* of the switch. In log file data quality problems were gathered around the time stamp of the logged events. The time of the logs were

included late in the protocol stack after several time-consuming steps, introducing a random additional delay variation of 0–0.2 sec. Moreover, this depends on the software and hardware components utilised at that time in that particular switch. Time was also stored in a format that had only 0.1 sec granularity. These inaccuracies should be eliminated in order to see the full potential of only utilising the time measures of log files to predict railway point condition. A minor improvement to data reading would enable changing the HIVE table date strings to date format to allow more flexible structured query language (SQL) utilisation in the database. The main part of the thesis is the feasibility analysis of the predictability of switch failures in selected failure classes. The visual inspection with Q–Q plots revealed that measurements from the log files are clearly different before and after the maintenance. However, distributions overlap widely, indicating that information from several past turns needs to be included in the prediction model. Analysis of the correlation properties of seven typical sample characters derived from two log measurements (and additionally from *log warning messages*) showed that typical characters are highly dependent. Model-specific feasibility analysis shows that the optimal feature set rarely included all seven features and the maximum time was typically the most important feature. The further feature selection study for logistic regression revealed that with this limited data measurement accuracy, the maximum *motor on* time inside samples would give almost as good prediction as any characteristic combination. This implies that a simple limit for that value would predict most of the cases correctly. In literature, these limits have been studied and found to be impractical, however, in those studies the target was the elimination of regulated periodical maintenance and it becomes as a safety issue [Garcia Marquez et al., 2003]. The aim of our prediction application was less ambitious – to produce a maintenance planning assistant – and so we do not have the same safety issue and thus the same conclusion cannot be drawn. The prediction accuracy of all models increases steadily when the sample size grows from 10 to 30 events. Moreover, the prediction and training accuracy are at a similar level for all models, [1] suggesting that it is possible that the data cannot provide better separation. The accuracy results show that the maintenance need is clearly visible just before the actual call, which is as assumed. The prediction accuracy drops from over 90 percent to 80 percent when receding from five turns distance from the call to 10 turns distance. Receding further it slowly decreases and is below 70 percent when the distance is 30–70 turns from the actual maintenance call. These figures are far

[1] Excluding random forest that has the capability to clearly overfit the model to training data.

from the over 97 percent accuracies achieved with the help of sensors in a test environment [Garcia Marquez et al., 2003] [Garcia Marquez et al., 2010]. Better accuracy was clearly expected since our approach only utilised a rough estimation of time duration of one sensor that reports the power usage curve instead of the whole information without even considering other sensor measures. However, since those results were conducted under test conditions it remains unclear how early these failures would be detectable. Besides prediction accuracy, the error profile is important for the feasibility of the models in maintenance planning assistance. Random forest is the only model that clearly has a different error profile. It has better AUC performance, over 80 percent, however, this does not imply superior usage for our purpose. Other models avoid false positive prediction at the expense of missing more warnings. Thus, they have better precision (over 90 percent) at the expense of worse recall (around 55) and also worse AUC (roughly 75 percent). However, for planning assistance the precision suits as a best error measure since it implies reliable assistance for an expensive extra maintenance condition visit to a railway point location. Even these models do not predict all faults but whenever they indicate a fault it is worth putting a visit to that switch at the top of the work pile. All in all, the thesis shows that there is potential to continue the work. Continuation should happen in a project including several players in the rail network industry. One general digitalisation goal would be the standardisation of failure categories among Finish maintenance companies and including them as one field in the maintenance report. Other targets are to take the feasibility study to the next level and start the working procedure from phase one and together with this group of companies define the most valuable outcome then, for example, take the wider railroad area, complete maintenance data, test accurate *motor on* times (measured with a sensor) or utilise the electricity consumption curves produced by test sensors. This would naturally include the dynamic extension of the prediction models. The correct institution in which to start the discussion of such a project would be Finnish Transport Agency.

Bibliography

S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

B. Boser, I. Guyon, and Vapnik V. Training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.

Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

K.P. Burnham and D.R. Anderson. Model selection and multimodel inference: A practical information-theoretic approach. In *Springer, USA ISBN 0-387-95364-7*, pages 1–47, Second edition 2002.

Longbing Cao. Data science: A comprehensive overview. *ACM Computing Surveys*, 19(3):Article 43 with 42 pages, 2012.

V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):Article 15, 2009.

CRISP-DM Consortium : P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. Crisp-dm 1.0: Step-by-step data mining guide, 1999.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20: 273–297, 1995.

J.S. Cramer. Logit models from economics and other fields. In *Cambridge University Press, Cambridge, England, pp. 149-158*, 2003.

P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classifier. *Proceedings 13th International Conference on Machine Learning*, pages 105–112, 1996.

T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27 (8):861–874, 2006.

M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundres of classifiers to solve real world classification problem? *Machine Learning Research*, 15:3133–3181, 2014.

S.E. Fienberg. When did Bayesian inference become Bayesian? *Bayesian Analysis*, 1(1):1–40, 2006.

D. Fisher, R. DeLine, M. Czerwinski, and S. Drucker. Interactions with big data analytics. *Interactions Magazine*, 50(3):50–59, 2017.

R. A. Fisher. The use of multiple measurements in taconomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

M. Friendly and D. Denis. The early origins and development of the scatterplot. *Journal of the history of the behavioral sciences*, 41(2):103–130, 2005.

F. P. Garcia Marquez and F. Schmid. A digital filter-based approach to the remote condition monitoring of railway turnouts. *Reliability Engineering and System Safety*, 92:830–840, 2007.

F. P. Garcia Marquez, F. Schmid, and J. C. Collado. A reliability centered approach to remote condition monitoring. a railway points case study. *Reliability Engineering and System Safety*, 80:33–40, 2003.

F. P. Garcia Marquez, C. Roberts, and A. M. Tobias. Railway point mechanisms: Condition monitoring and fault detection. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 224:35–44, 2010.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Machine Learning Research*, 3:1157–1182, 2003.

Tin Kam Ho. Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition 14-16 Aug*, 1:278–282, 1995.

V. J. Hodge, S. O. Keefe, M. Weeks, and A. Moulds. Wireless sensor networks for condition monitoring in the railway industry: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1088–1106, 2015.

T. Joachims. Training linear svms in linear time. *KDD ´06 Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226, 2006.

Ji-Hyun Kim. Computational statistics and data analysis. *Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap*, 53:3735–3745, 2009.

M. Kim, T. Zimmermann, R. DeLine, and A. Begel. The emerging role of data scientists on software development teams. *ACM 38th IEEE International Conference on Software Engineering*, pages 96–107, 2016.

N.D. Lawrence. Data readiness levels. *Cornell University Library, Computer Science Databases arXiv:1705.02245*, 2017. `https://arxiv.org/abs/1705.02245` Accessed 29.12.2017.

Liikennevirasto. Rautatieliikenteen täsmällisyys 2011, 2012. Liikenneviraston Tutkimuksia ja selvityksia 16 2012 `https://julkaisut.liikennevirasto.fi/pdf3/lts_2012-16_rautatieliikenteen_tasmallisyys_web.pdf`.

Liikennevirasto. Vaihdekäsikirja: Vaihteen huolto-ohjeet, 2016a. Liikenneviraston ohjeita 23/2016 `http://www2.liikennevirasto.fi/julkaisut/pdf8/lo_2016-23_vaihdekasikirja_web.pdf`.

Liikennevirasto. Ratatekniset ohjeet (rato) osa 14 vaihteiden tarkastus ja kunnossapito, 2016b. Liikenneviraston ohjeita 14/2016 `http://www2.liikennevirasto.fi/julkaisut/pdf8/lo_2016-14_rato14_web.pdf`.

Liikennevirasto. Rautatieohjeet, 2017. web page with updated instructions `http://www.liikennevirasto.fi/palveluntuottajat/ohjeluettelo#.Wd41Zy_TRTa`.

Wei-Yin Loh. Fifty years of classification and regression trees. *International Statistical Review*, 82(3):329–348, 2014.

J. A. Nelder and R. W. M. Wedderburn. Beyond independence: Conditions for the optimality of the simple bayesian classifier. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):170–384, 1972.

A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems*, 14, 2001.

Y.and Jordan M. Ng. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems*, 14:605–610, 2001.

T. Oshiro, P. Perez, and J. Baranauskas. How many trees in a random forest? *Proceedings of Machine Learning and Data Mining in Pattern Recognition 8th International Conference*, pages 154–168, 2012.

B. O. Oyebande and A. C. Renfrew. Condition monitoring of railway electric point machines. *IEE Proceedings - Electric Power Applications*, 149(6): 465–473, 2002.

Joern Pachl. Railway operation and control. In *VTD Rail Publishing, USA ISBN 0-9719915-1-0*, Second print 2004.

K. Pearson. Contributions to the mathematical theory of evolution. ii. skew variation in homogeneous material. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences.*, 186, 1895.

J. Piironen and A. Vehtari. Comparison of Bayesian predictive methods for model selection. *Statistics and Computing*, 6(3):711–735, 2017.

KDnuggest Methodology poll. What main methodology are you using for your analytics, data mining, or data science projects?, 2014. WWW page: `http://www.kdnuggets.com/polls/2014/analytics-data-mining-data-science-methodology.html`. Accessed 11 Oct 2017.

L. Rokach and O. Maimon. Top-down induction of decision trees classifiers: A survey. *IEEE Transaction On Systems Man and Cybernetics - Part C: Applications and Reviews*, 35(4):476–487, 2005.

F. Rong-En, C. Pai-Hsuen, and L. Chih-Jen. Working set selection using second order information for training support vector machines. *Machine Learning Research*, 6:1889–1918, 2005.

Jukka Saha. Master thesis: The strategy of track maintenance and the development of the track maintenance operations, 2017. `http://www.theseus.fi/bitstream/handle/10024/124802/Saha_Jukka.pdf`.

Colin Shearer. The crisp-dm model: The new blueprint for data mining. *Journal of Data Warehousing*, 5(4):13–23, 2000.

T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer. Rocr: visualizing classifier performance in r. *Bioinformatics*, 21:3940–3941, 2005.

J. A. Swets. The relative operating characteristics in psychology. *Science*, 182(4116):990–1000, 1973.

A. Vehtari and J. Ojanen. A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, 6:142–228, 2012.

M.B. Wilk and R. Gnanadesikan. The method of probits. *Science*, 79(2037): 38–39, 1934.

M.B. Wilk and R. Gnanadesikan. Probability plotting methods for the analysis for the analysis of data. *Biometrika*, 55(1):1–17, 1968.

N. Wright, R. Gan, and C. McVae. Software and machine learning tools for monitoring railway track switch performance. *IET Railway Condition Monitoring Conference Birmingham, UK, September 27-28,2016*, 2016.

F. Zhou, M. Duta, and M. Henry. Remote Condition Monitoring For Railway Point Machine. *ASME/IEEE Joint Rail Conference Washington, DC, USA, April 23-25,2002*, 2002.