MASTER

Optimizing maintenance decision-making based on imperfect predictions

Derks, T.E.H.

Award date:
2017

Eindhoven, April 2017

Optimizing Maintenance Decision-making based on Imperfect Predictions

by
T.E.H. Derks

BSc Industrial Engineering & Management Sciences – TU/e 2014
Student identity number 0748484

in partial fulfilments of the requirements for the degree of

Master of Science
in Operations Management and Logistics

Supervisors:
dr. H. Blok, TU/e, OPAC
dr. A. Akçay, TU/e, OPAC
dr. G. Xiao, TU/e, OPAC
dr.ir. M. Boosten, Royal Philips N.V.

TU/e School of Industrial Engineering.
Series Master theses Operations Management and Logistics

Subject headings: Proactive Maintenance, Failure Predictions, Capital Goods, Decision-Making, Imperfect Information, Mathematical Modelling, Newsvendor model, Value of Information

"Live as if you were to die tomorrow.
Learn as if you were to live forever."

–Gandhi

ABSTRACT

In this research, we developed a decision support model for Remote Monitoring Engineers (RME). Failure prediction models generate predictions for upcoming equipment failures. RMEs receive these predictions and aim to initiate proactive maintenance actions to prevent failures. However, the predictions are not always good. Predictions can be false or the prediction models can miss failures. The decision support model covers the decision to initiate actions on such predictions accounting for the imperfectness. Furthermore, the model supports in the timing decision of the maintenance actions.

The imperfectness of a prediction is measured by the probability that this prediction is true. We developed a newsvendor solution to find an optimal probability threshold to act on a prediction or not.

Keywords: Proactive Maintenance, Failure Predictions, Capital Goods, Decision-Making, Imperfect Information, Mathematical Modelling, Newsvendor model, Value of Information

MANAGEMENT SUMMARY

In this report, we present the results of our research on maintenance decision-making based on imperfect predictions conducted at Philips. Philips develops advanced equipment for the diagnosis and treatment of patients in hospitals. Philips wants to keep the equipment up and running as much as possible by employing a proactive maintenance policy. Such policy aims to conduct maintenance only when necessary to prevent the equipment to fail.

Problem statement

Philips developed failure prediction models to predict future failures of the customer's equipment. These models analyze data generated by the software running on the equipment. Based on this data, the predictive models can generate an alert if it is likely that a failure will occur in the near future. Remote Monitoring (RM) needs to judge these alert and make the decision to take proactive actions to prevent the failures. However, the information generated by predictive models is not perfect. The models can generate false alerts and they can miss failures. It is not fully understood how RM can account for this information in their decision-making. This motivated us to define the following main research question:

How can Philips optimize the proactive maintenance decision-making by Remote Monitoring, accounting for the imperfectness of information on machine conditions?

Analysis of the current situation

We conducted an analysis on the current situation to gain knowledge on current practices and characteristics of the maintenance policy. The results of the analysis are used as an input for the design phase of the project. We identified which service requirements are relevant for the proactive decision-making by Remote Monitoring Engineers (RME). It is important that RMEs consider these requirements in their decision-making to make sure customers are treated according to their service contract.

In order to understand the imperfectness of the predictive models, we did some research on the development process of predictive models and identified why the models are imperfect. Understanding this imperfectness provides a starting point from which the predictive models can be improved.

Maintenance costs are very important in a maintenance policy. We conducted an analysis on the costs in the current maintenance policy. An important aspect is the analysis of the costs of imperfectness of the predictive models. Figure 1 shows the comparison of proactive maintenance versus reactive maintenance in terms of hours spend. In proactive maintenance, RM initiated maintenance actions that prevented the failure. In reactive maintenance, a failure occurred that was not predicted by predictive models or RM did not initiate proactive maintenance actions for a valid alert. We showed in Figure 1 that the costs of reactive maintenance are more than twice as high as the costs of proactive maintenance. Therefore, it is not desirable that the predictive models miss many failures. The costs of false alerts were found to be significantly lower.

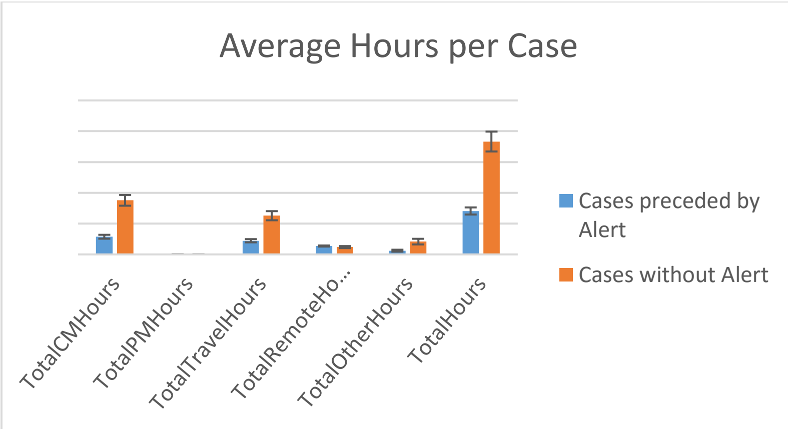


Figure 1: Costs comparison of proactive maintenance and reactive maintenance

Decision-support model

A mathematical model has been developed to provide RMEs with support in their maintenance decisions. It provides information on the short-term consequences in terms of expected costs and expected downtimes of possible actions they can take. The model supports in the decision to initiate maintenance actions or not, and when the maintenance actions should be performed. Some customer contracts include an uptime guarantee. We recommend for these contracts to make the decision to minimize the expected downtime. For customers with other contracts, we recommend to make the decision that minimize the expected costs. We developed a newsvendor solution on how credible an alert should be to take actions on it. We tested this solution under several circumstances and it is found to be an optimal probability threshold. The results are visualized in Figure 2. This figure shows that for more credible alerts (i.e. a higher value for P), it is beneficial to initiate maintenance actions.

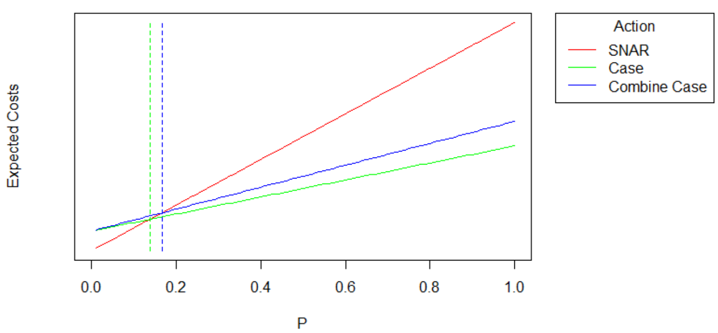


Figure 2: Influence of P on expected costs

With a case study on the Flat Detector, we tested the decision support model for customers with different characteristics regarding their service contract. The decision support model provides customer specific support to RMEs.

Value of information

We conducted research on the value of information generated by predictive models. This value can be accessed by calculating the difference in expected costs or downtime of having information available or not. It provides a guideline on how much Philips can pay to receive such information. We calculated the value of information under different levels of imperfectness. Even unreliable information about a future failure can be valuable in terms of both expected costs and downtime savings. Predictions with a certainty of just 20% are already valuable and can bring savings in expected costs and downtime. The value of information can differ among customers with different service contracts. These differences are shown in Figure 3.

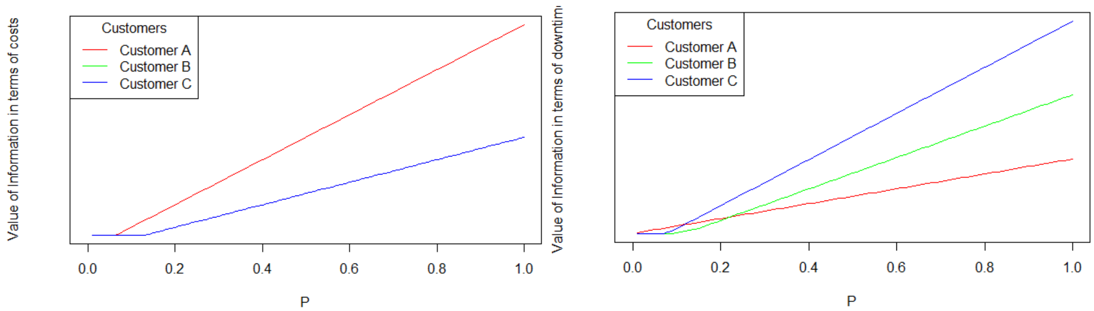


Figure 3: Value of information in terms of costs and downtime

This figure shows for different customers how valuable an alert with different probabilities that it is true is. It provides insights in how much Philips can pay to make such information available.

PREFACE

This report is result of my graduation project as part of the Master Operations Management & Logistics at Eindhoven University of Technology. The project took place at Royal Philips in Best from September 2016 to March 2017 and was part of MANTIS. I am glad that I had the opportunity to do this project in the inspiring company that Philips is.

The Master thesis marks the end of my master and my student life and represents the start of a completely new phase in my life. It has been an inspiring journey in which, I learned a lot about the academic world, business, and about myself. I would like to use the remainder to express my gratitude to a number of people that played a role in my thesis and my study life.

First of all, I would like to thank Herman Blok. Our discussions about my thesis were always very constructive. You helped me a lot with the mathematical modeling, which really enriched my thesis. I really appreciated your willingness to help me with all the problems I faced during my research. Herman, thanks for all your help.

Secondly, I would like to thank my second supervisor, Alp Akçay. Alp, you gave me the opportunity to conduct my thesis as part of the MANTIS program. During my thesis, you were really engaged in my project and you were more than a second supervisor to me. Çok teşekkür ederim Alp!

I would also like to thank Guanlian Xiao for her role as third assessor. Although your official role was limited to just assessing the final version of my thesis, we had constructive discussions about the further research to be conducted in Philips.

Next, I would like to express my gratitude to a number of people from Philips that were involved in my project. Many thanks go to my company supervisor Marcel Boosten. During our (almost) weekly meetings, we had good discussions on all kind of topics related to my thesis. Your enthusiasm really motivated me throughout my entire research. I would also like to thank the people from remote monitoring and from data science. You were always helpful to answer the many questions I had during my thesis. Without your input, this thesis was never established as it is now.

Moreover, I want to thank all my friends from Eindhoven and Overloon for the enjoyable time next to the working hours in Philips and the university. Especially, I want to thank my 'friends' from 'Huize Kruis'. After a stressful day, it felt always good to come home and having much fun with you. Thanks Jenna, Stan, Rene, Laura, and Jiayi!

Last but not least, I would like to thank my family and my parents in particular. It always felt good to come home to Overloon in the weekends. You always gave me unconditional support in my study and in my life. I cannot thank you enough for having you in my life!

Tom Derks

LIST OF ABBREVIATIONS

CBM: Condition-Based Maintenance

cdf: Cumulative Distribution Function

CI: Confidence Interval

CM: Corrective Maintenance

FSE: Field Service Engineer

IMS: Intelligent Maintenance Systems

iXR: Interventional X-Ray

KPI: Key Performance Indicator

LSO: Local Service Organization

pdf: Probability Distribution Function

PdM: Predictive Maintenance

RM: Remote Monitoring

RME: Remote Monitoring Engineer

RSE: Remote Service Engineer

SLA: Service Level Agreement

SNAR: Seen No Action Required

SVM: Support Vector Machine

LIST OF FIGURES

List of Tables

CONTENTS

1 INTRODUCTION

"The beginning is the most important part of the work." – Plato

This report presents a research on maintenance decision-making based on imperfect predictions conducted at Royal Philips. Royal Philips develops medical systems (e.g. MRI scanner or CT scanners) that are used in hospitals for diagnostic and treatment purposes of patients. Maintenance costs are usually a big part of the total lifecycle costs of such high tech systems. Therefore, companies try to find strategies to keep these costs as low as possible while keeping the systems up and running as much as possible. Unexpected downtime often leads to excessive costs, loss of revenues, or for Royal Philips' customers, delayed treatments of the patients.

Over the years, these maintenance strategies evolved from run-to-failure strategies to strategies that monitor the 'health' or condition of the system and try to conduct maintenance only if the system is expected to fail in the near future. Such strategies are often referred to as Condition-Based Maintenance (CBM) strategies. In such strategies, the condition of the system can be accessed in various ways. This can be done by periodic inspections or by continuously monitoring certain characteristics of the behavior of the system (e.g. vibrations or acoustics). Continuously monitoring requires special equipment to be installed on the system that measures such characteristics with sensors.

Nowadays, these systems are operated by software allowing the systems to generate data of its usage, and its (malfunctioning) behavior. Such data is stored in log files, processed and used to predict failures in the near future with state-of-the-art data science techniques. However, such failure predictions are subject to imperfectness due to various reasons. The predictions need to be evaluated and judged and maintenance decisions should be made upon them.

1.1 COMPANY INFORMATION

Royal Philips (commonly referred to as Philips) was founded in 1891 by Gerald Philips and his father Frederik Philips in Eindhoven. It started as a company manufacturing incandescent lamps and other electro-technical products. Later, Philips became one of the biggest light bulb producing companies in the world. Today, Philips is headquartered in Amsterdam, The Netherlands, and has evolved into a technology multinational, offering a variety of products. In 2015, the comparable sales grew to EUR 24.2 billion and the net income to EUR 659 million. Philips employs approximately 113.000 employees and provides sales and services in approximately 100 countries in the world. Philips' mission is to improve people's life through meaningful innovation. The company serves both professional and consumer markets throughout the world in areas of health systems, personal health and lighting solutions. Recently, Philips split into two stand-alone companies, Royal Philips, active in health technology and Philips Lighting, active in lighting solutions. This research takes place in Royal Philips. For the rest of this thesis, we refer to the company as Philips.

1.2 Problem solving methodology

This chapter discusses the research methodology used to solve the research questions. During the project, the regulative cycle discussed in van Aken, Berends, and van der Bij (2007) will be followed. This cycle is a method that is often used in business problem solving. The cycle is given in Figure 4.

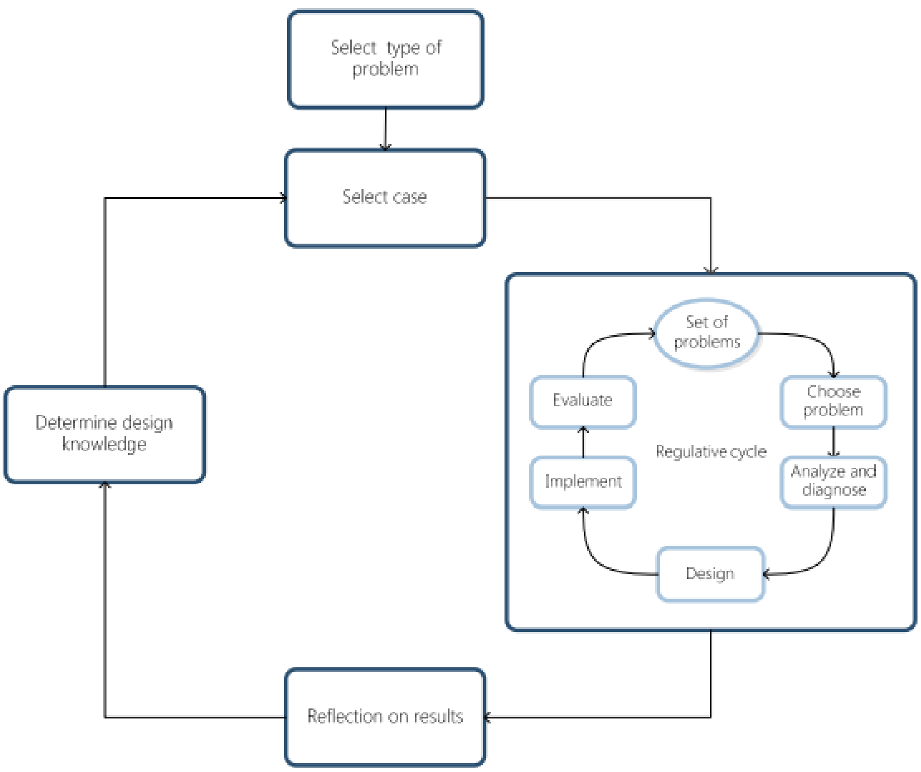


Figure 4: Reflective cycle including the regulative cycle (van Aken, Berends, & van der Bij, 2007)

Following this cycle, the first step is to select a type of problem. The type of problem in this project is optimization of maintenance activities based on imperfect information. The literature research revealed that this is a more or less undiscovered research area which makes it an interesting topic. The case selected for this research is the case of maintenance decision-making within Philips. The next step in the reflective cycle is to enter the regulative cycle. Maintenance decision-making is still a very broad problem so the topic of the thesis should be narrowed.

The specific problem chosen for this research is 'the accounting for imperfect predictions in decision making by RMEs'. It is expected that solving this problem will eliminate a large set of sub problems. This maintenance decision making is done by RMEs and the imperfect information is received from predictive and proactive models discussed before.

This problem will be further analyzed and diagnosed in the next step. Once the problem is analyzed and diagnosed, a design should be made to solve the problem. For this project, the design should optimize maintenance decision making for RMEs accounting for imperfectness of information.

The next step is to implement, the design in the organization. We will apply the decision support model in a case study. The actual implementation is not part of this thesis.

Once the design is implemented and employed in the organization for some time, the implemented design should be evaluated. This reveals existing problems with the design and its implementation such that they can be solved.

1.3 Predictive maintenance

According to Sharma, Yadava, Deshmukh (2011), maintenance can be categorized in the following three classes:

1. Preventive maintenance – The maintenance actions are carried out on a planned and periodic schedule.
2. Corrective maintenance – Unscheduled or repair maintenance actions. These are often carried out in case of a machine failure
3. Predictive maintenance – Maintenance actions are conducted based on information from modern measurement and signal processing methods to predict and diagnose the condition of machines.

Predictive maintenance is also often referred to as Condition-Based Maintenance (CBM) and it aims to conduct maintenance just before a failure arises such that failures are prevented and the equipment is used as long as possible to prevent unnecessary maintenance. This literature study will focus on the use of data on machine condition prognostics in related maintenance decisions.

In a CBM policy, these decisions are made based on the observed health or condition of the machine. The main goal of CBM is to assess equipment real-time in order to make maintenance decisions that reduce unnecessary maintenance and related costs (Gupta & Lawsirirat, 2006). It attempts to monitor this health based on condition measurements without interrupting the operation of machines (Heng, Zhang, Tan, & Mathew, 2009). CBM consists of the following three steps: 1) Data acquisition, 2) Data processing and 3) Maintenance decision-making (Jardine, Lin, & Banjevic, 2006). Step 1) and 2) will lead to a better understanding of the current condition of the machine and it is important input for the decision making (Lewandowski & Oelker, 2014).

Machine fault diagnostics and prognostics are important topics in CBM (Jardine, Lin, & Banjevic, 2006). Fault diagnostics is related to the detection, isolation, identification of the machine fault. Fault detection means to indicate whether something is wrong, isolation to locate the faulty component and the fault identification means the determination of the nature of the detected fault. This fault identification is also often referred to as the determination of the failure mode. Each failure mode can have different triggers and a different deterioration pattern (Siborska, Hodkiewicz, & Ma, 2011). Prognostics deals with the prediction of machine faults. It tries to determine if a fault is impending and it tries to estimate when and how likely it is that a fault will occur (Jardine, Lin, & Banjevic, 2006). This can be worth the effort because 99% of the machine failures are preceded with by some malfunction signs or other indications that a failure is impending to occur (Bloch & Geitner, 1983). A fault occurrence triggers the fault diagnostics, while prognostics is done in advance of the occurrence of a fault.

The use of Intelligent Maintenance Systems (IMS) has been suggested to enable a proactive maintenance management strategy, which determines when maintenance should take place based on different condition indicators. IMS are embedded diagnostic and prognostic systems that try to forecast failures aiming to improve the related maintenance processes (Djurdjanovic, Lee, & Ni, 2003). The goal of IMS is to monitor the degradation status of a machine or components by sensors and embedded devices. Future failures are predicted based on this information and by using algorithms for health estimations (Frazzon, Israel, Albrecht, Pereira, & Hellingrath, 2014).

Salfner and Malek (2007) provide a method for effective online failure prediction. The focus of online failure prediction is to perform short-term failure predictions based on the current runtime state of the equipment. The time relations in online failure prediction are shown in Figure 5.

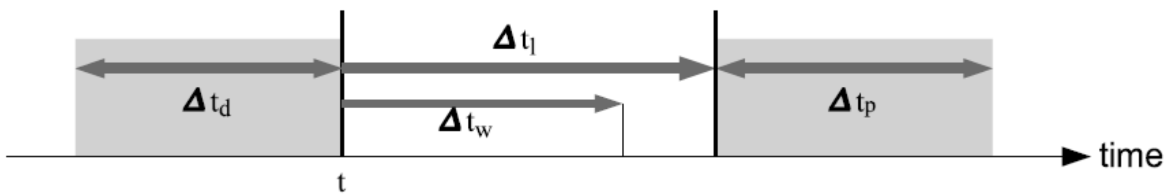


Figure 5: Time relations in online failure prediction

t represents the present time. Δt_d represents the data window size from which historical data is taken. Δt_w represents the warning time which is determined by the time needed to perform proactive action. Δt_l is the total lead time from the prediction to the moment the problem is solved and Δt_p represents the prediction which describes the length of the time interval for which the prediction holds. This terminology can be used to characterize the predictive models used in Philips.

The authors use hidden semi-Markov models (HSMM) and demonstrate the effectiveness based on field data. The basic assumption for the use of HSMM is that failure-prone behavior can be identified by patterns of errors. The authors show by an experiment that such an approach is very effective with respect to online failure prediction. Different methods are assessed and compared in terms of precision, recall, F-measure, false-positive rate, and computing time.

One type of online failure prediction uses log files from the advanced systems that are discussed earlier. When these log files are stored on a central database that can be accessed online, the information can be used to predict failures. Sipos et al. (2014) present a data-driven approach based on multiple-instance learning for predicting equipment failures by employing data mining techniques on the event logs. They use state-of-the-art machine learning techniques to build predictive models from log data. Their approach also utilizes data on service actions. The workflow of their approach is given in Figure 6.

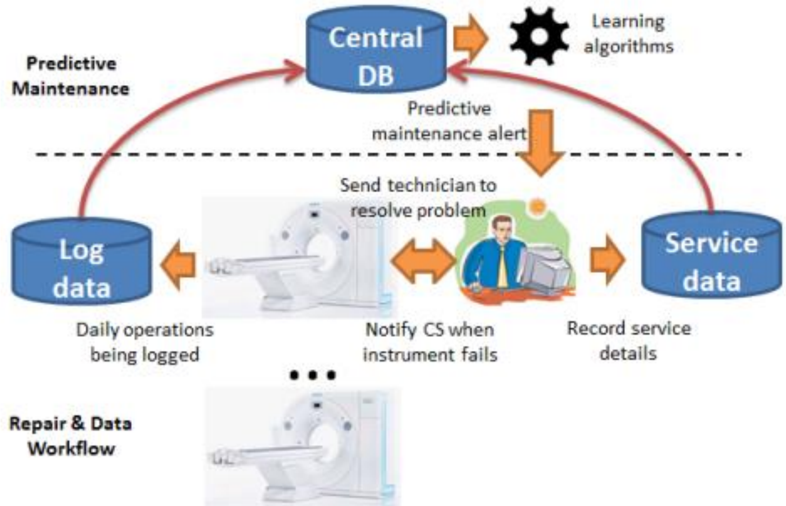


Figure 6: Log-bases predictive maintenance workflow (Sipos et al., 2014)

The predictive models generate alerts based on the log data that warn for a possible failure in a certain time frame. These alerts can be classified as true or false. In order to make log-based preventive maintenance useful and practical, Sipos et al. (2014) define the following requirements related to the timing of an alert:

- Predictive interval: a pre-defined time interval before a failure in which an alert occurance gives enough time to act upon it.
- Infected interval: a pre-defined time interval after a failure in which the equipment is breaking down or under repair.
- Responsive duration: a pre-defined time length for a real-life action for an alert.

The performance of these models can be evaluated by the precision and recall as defined in the next section. The actual maintenance strategy should be determined separately for different components with different models and is influenced by many factors.

1.3.1 Imperfect failure predictions

The previous sections discussed methods that handles a system of which the state cannot be directly observed resulting in imprefect estimations of the system's condition state. Another type of research deals with failure prediction methods that try to predict whether or not a failure will occur within a certain time period. The aim of these methods is to predict as many failures that arrive while generating as few false alarms as possible. These false alarms are ofter refered to as *False Positive (FP)* predictions. *True Positives* (TP) are predictions that predict a failure right. If the model misses to predict a failure it is called a *False Negative (FN)* and if no failure occurs and no alarm is given, the prediction is a *True Negative (TN)*. Any failure prediction belongs to one of these four cases which are covered in a confusion matrix as given in Table 1 (Salfner, Lenk, & Malek, 2010).

Table 1: Confusion matrix failure predictions

		Predicted	
		Failure	**No Failure**
Actual	**Failure**	**True Positive (TP) (correct warning)**	**False Negative (FN) (missing warning)**
	No Failure	**False Positive (FP) (false warning)**	**True Negative (TN) (correctly no warning)**

Metrics like the *precision* and *recall* can be calculated with this confusion matrix. Precision is the ratio of correctly identified failures to the number of all predicted failures so that:

$$precision = \frac{TP}{TP + FP}$$

The recall is often defined as the ratio of correctly predicted failures to the number of true failures so that:

$$recall = \frac{TP}{TP + FN}$$

There are also many other metrics that can be calculated with numbers from the confusion matrix.

According to Salfner, Lenk, and Malek (2010) often the trade-off has to be made between the FN rate and the FP rate. Reducing the FP rate often results in the increase in FN rate. Using previous formulas, we can say that improving the precision of a prediction model often results in a worse recall.

Candea, Kawamoto, Fujiki, Firedman, and Fox (2004) examined the trade-off between confidence in the correction of failure prediction and the costs of acting on the prediction in case of software failure management. The authors show that short reboot times (low cost of action) allow for higher false positive rates than slower restarts (higher cost of action). Although their research was focused on software failures, their findings might also be valid in other failure management areas.

1.3.2 Proactive decision-making

The aim of predictive maintenance strategies is to prevent failures by taking proactive actions. The decisions to take proactive actions are covered in a proactive decision making process. The decisions in such processes require output from prognostic and diagnostic models. A proactive decision support framework is provided by Bousdekis, Magoutas, Apostolou, & Mentzas (2015). This framework is shown in Figure 7.

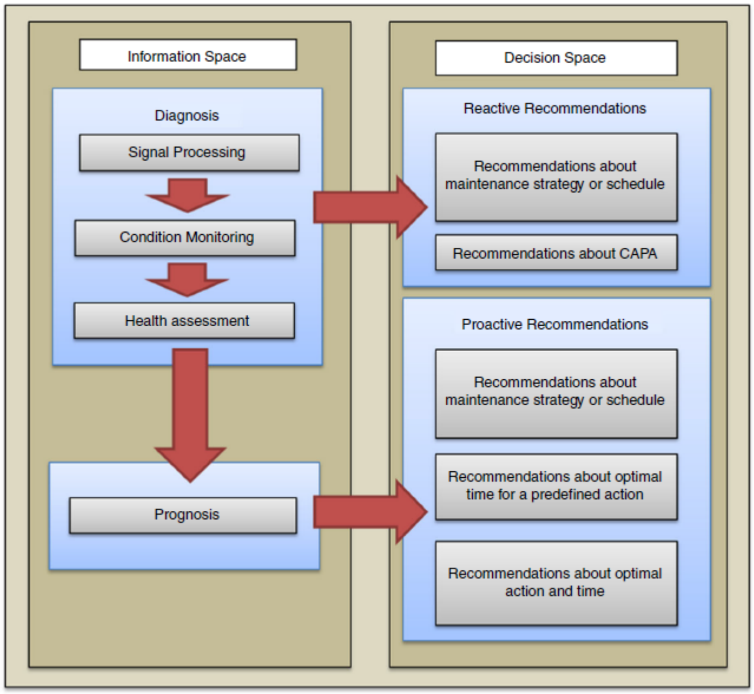


Figure 7: Proactive decision-making framework (Bousdekis, Magoutas, Apostolou, & Mentzas, 2015)

The information space in the framework consists of diagnostic and prognostic models. Information from both models is required in the decision space, which consists of reactive and proactive actions or recommendations.

1.4 Predictive maintenance in Philips

In Philips, the information space as defined in Bousdekis et al. (2015) consists of Failure prediction models. These models aim to predict component failures in the near future. The prediction interval differs among models, but are usually between 10 to 30 days. Chapter 4 provides more information about these predictive models. The predictive models analyze data sent by the software on the system, and generate alerts that contain failure predictions. These alerts contain both diagnostic and prognostic information about an upcoming failure.

Remote Monitoring (RM) reviews these failure predictions. RM is a team that consists of Remote Monitoring Engineers (RME), which are responsible for initiating proactive maintenance actions. Proactive maintenance actions are actions to prevent future failures instead of acting on failures. RMEs do not perform the actual proactive maintenance actions on the customers' equipment. They only send recommendations to the Local Service Organizations (LSO). RMEs can either reject an alert, or initiate proactive maintenance actions. Rejecting an alert is called SNAR, which stands for Seen No Action Required. If the RME wants to initiate proactive maintenance actions, he creates a case and sends this case the LSO. In the LSO, Remote Service Engineers (RSE) and Field Service Engineers (FSE) are responsible to execute the case according to the recommendations given by RMEs. The LSO is responsible for providing local services on customer's equipment in a specific country or region, while RM monitors customer's equipment globally. Whenever a customer experience problems with their equipment, they call the LSO in the specific region.

1.5 THESIS OUTLINE

Chapter 2 provide the research design that is followed during this thesis. In Chapter 3, the different service contracts used in Philips are explained and the modeling requirements related to these contracts are identified. Chapter 4 discusses how predictive models are developed and why they are not generating perfect information. Chapter 5 contains an extensive analysis of the current situation regarding the proactive maintenance processes. It contains an analysis of the costs and an analysis of spare parts decisions in proactive maintenance cases. Furthermore, we analyzed and modelled the component's expected remaining lifetime after an alert, and we modelled the time it takes until a proactive maintenance case is executed. An short-term decision support model is developed in Chapter 6, which is applied in a case study in Chapter 7. Chapter 8 discusses the value of information and the conclusions and recommendations are discussed in Chapter 9.

2 RESEARCH STATEMENT

"A problem well stated is a problem half solved." – Charles Ketterling

This chapter contains the research statement of the thesis and contains information from the full master thesis project proposal. Section 2.1 contains some background of the problem within Philips and Section 2.2 contains the research questions and scope of the thesis.

2.1 PROBLEM BACKGROUND

Healthcare Imaging Systems are essential for the diagnosis and treatment of patients. Due to high costs involved, it is not feasible to implement backup systems. Therefore, the system downtime needs to be minimized while keeping the costs low. It is very important that the maintenance policy employed for the systems is reliable and cost efficient. A Condition-Based Maintenance (CBM) policy aims to conduct maintenance just before a failure arises. As mentioned, RM continuously monitors the condition of some components of equipment in the field. Data is gathered, stored, analyzed and used to predict failures in the systems. However, these predictions are not always perfect and not every failure is predicted. Different failure prediction models are used and each of them has its limitations. There is no guidance for RM on how to account for the imperfectness in their decision-making.

Besides the problem of the imperfect predictions, it is not exactly clear for RM how they should account for the service contract of the customer in their decision-making. This is a relevant problem because it is important that service is delivered in accordance with the service contracts. Underperformance can result in a lack of customers' trust and over-performance can make customers not see the value of having better service contracts. Differentiating customers based on the level of customer value implied by their service contract enables to make customer specific optimal decisions.

2.2 RESEARCH QUESTIONS

Based on the research gaps found in the literature study and the current problems in Philips, several research questions can be constructed. Maintenance decisions by RMEs are now subjectively made based on their experience and their knowledge of the system. The maintenance decision making lacks an objective foundation. There is a need for some support in how RMEs can make better decisions accounting for the characteristics of the alerts and the service contract of the customer.

How can Philips optimize proactive maintenance decision-making, accounting for the imperfectness of information on machine conditions?

1. How should the RMEs account for the service level agreements?
2. How can RMEs account for imperfectness in predictions?
3. What are the relevant cost factors in the maintenance decision making by RMEs?

2.2.1 Scope

The scope of the project is the business-oriented maintenance decision making using the information provided by predictive models. We consider information generated by these predictive models as input for the maintenance decision-support model. The decision-support model covers two decisions. The first decision is to initiate proactive maintenance actions for an alert raised by a predictive model or not. The second decision is related to the timing of the alert. RMEs recommend when the maintenance actions should be performed. The decision support model should provide the RMEs with support in this decision. It should provide a recommendation on what is the best time to execute the proactive maintenance actions. Therefore, we define the decision scope of the decision support model by:

- Initiate proactive maintenance actions for an alert or not
- The timing of the maintenance actions

2.2.2 Modeling scope

The goal of the thesis is to create a decision-support model for the RMEs on how to act on alerts. Only actions on alerts are included in the scope of this thesis so that decision making by RSEs related to customer calls is initially excluded from the project scope. We expect that the decision scope of RSEs is very different from the decision scope of RMEs. Including both decision-making processes in the modelling scope is too broad. Therefore, we focus on the decision-making by RMEs.

It is important that the parameters that should be incorporated in the model are carefully selected. The different parameters of the model are listed below and explained afterwards.

- Correctness of information
  - Model reliability
  - Confidence
- Customer contracts
  - Service Level Agreements (SLA)
    - Availability
    - Response time
  - Entitlements
- Potential value of actions

- Action Characteristics
    - Costs involved with the actions
        - Working Hours
            - Traveling
            - Maintenance
                - Remote
                - On-site

2.2.2.1 Correctness of information

As explained earlier, the correctness of information can be assessed by two different factors. The first one is the model reliability, which is characterized by the models confusion matrix. These confusion matrices are considered as input in the model.

In addition, the confidence gives some information on how likely it is that the alert is true. Therefore, it should be incorporated in the model. It is important that both the model reliability and the confidence are taken into account together. More research needs to be conducted on if and how these different aspects are related and how they should be interpreted.

2.2.2.2 Service Level Agreements (SLA)

One highly important requirement of the decision support model is that SLAs with the customers are satisfied. Philip's customers can sign for different service contracts ranging from basic support contracts to premium contracts. Customers with premium contracts should be provided with better and faster services such that these customers experience more value from their service contract. However, there are no clear guidelines on how REs should make different decisions based on the customer value.

Besides that, under-performance of the SLAs should be prevented, it is also important that there is not too much over-performance. Over-performing can lead to the situation that customers do not see the value of having premium contracts when the service is as good as for lower contracts. This should be prevented as much as possible since premium customers are the most valuable for Philips.

Further research should be conducted to gain knowledge on which kind of agreements are incorporated in the SLAs. Possible agreements can be on the maintenance budget, response times and availability.

2.2.2.3 Potential value of actions

The potential value of actions should also be taken into account and should be interpreted as the prevention of costs in the future. It means that a certain action can prevent future downtime costs. Consider an example when an alert is received for a specific part. Not acting on the alert can result in a failure with all associated costs. The potential value of acting on the alert can be seen as the prevention of these failure costs.

2.2.2.4 Action characteristics

Every action has its own specific characteristics. An action can fail with a certain probability and there are also costs involved in performing a maintenance action. Each possible maintenance action can be characterized by a success probability and costs involved with the actions. The success probability of an action is the probability that, if the action is performed for a certain problem, it will actually fix the problem successfully. Examples of such actions are the execution of remote service actions and on-site service actions. In general, on-site service actions have more chance to be successful but are costlier. In this thesis, the success probability of certain actions is considered as input.

There are always costs involved in performing maintenance actions. These costs can range from the costs of a 15 minutes call with the customer to troubleshoot the problem, to the replacement costs of a component. For some actions the costs can be fixed (e.g. costs of a part) but for others the costs can be vary (e.g. repair times or time needed for on-site diagnostics). It can be the situation that several parts are ordered to fix a problem. A possible reason can be to increase the probability that the problem is fixed in the first visit. When just one of the parts is used, the others should be send back to the warehouse.

3 SERVICE REQUIREMENTS

"Customers don't expect you to be perfect. They do expect you to fix things when they go wrong." – Donald Porter

This chapter contains the analysis of service contracts used in Philips. The goal of this chapter is to analyze the service requirements that should be accounted for in the maintenance decision making by RM. As discussed earlier, it is not clear to RMEs how they should account for these contracts. This thesis aims to provide decision support for RMEs that is customer specific, such that service is delivered according to the customer's contract. Section 3.1 discusses the different service contracts with a short description of each contract. Section 3.2 provides the implications of the service contracts on the decision-making by RM.

3.1 SERVICE CONTRACTS PHILIPS

Customers of Philips can sign for different service contracts for the equipment they buy. Recently Philips made the transition from three different service contracts (Silver, Gold and Platinum) to a more customer oriented service contract portfolio called RightFit. Although this transition, the old contract types are still often used. The new contracts aim to provide a better fit with the customer's needs by providing more flexibility. The different types of RightFit service contracts are discussed in an increasing order of coverage in the next sections. All service contracts include standard unlimited technical telephone support from the Customer Care Solution Center. Philips experts are on call available to provide live-assistance, 24/7 remote monitoring and remote diagnostic services. Note that this chapter only provides the general entitlements of the contracts. The exact entitlements differ per key market, per country and per modality. In the last section of this chapter, the relevance for remote monitoring will be addressed.

RightFit Assist

RightFit Assist is the most basic service contract. It provides scalable coverage for customers that have in-house support. It also includes unlimited technical telephone support from the Customer Care Solution Center. Besides the core offering, the customer can select a range of service options that provide coverage for different kind of parts and labor.

RightFit Support

With the RightFit Support contract, the customer and Philips share the responsibility of maintaining the system. The customer's in-house engineering teams have access to OEM parts and technical expertise from Philips. It includes full part coverage and unlimited second-response on-site labor along with optional strategic part coverage and part and labor pools. RightFit Support aims to provide the customer OEM expertise and support for their in-house engineers. Philips-trained engineers are working side-by-side with the customer's engineers to improve the expertise of the customer's engineers. The customer can adjust the contract to match it with the staffing levels and the skills of the in-house engineers.

RightFit Value

RightFit Value is a contract for customers that are looking for creative ways to minimize their service expenses while hedging some risks. It includes some part coverage, planned maintenance, and corrective maintenance at a relatively low price. Corrective maintenance is covered with a bank of labor hours or a bank of parts if needed. Services as uptime guarantee and clinical phone support are excluded in this contract.

RightFit Select

RightFit Select is a flexible offering from the service portfolio. It offers quick response to meet agreed service levels at moderate costs to fit within the customer's budget. Several coverage options are available to add more protection.

RightFit Primary

RightFit Primary is a customizable offering in the service portfolio. It gives the flexibility to customize the service coverage to the unique needs of the customer. It includes full parts coverage and a 98% uptime guarantee along with a four-hour, on-site response time. The customer can choose from a wide range of options to sign for more service coverage.

RightFit Protection

RightFit Protection aims to provide the customer with complete protection. The weekday coverage is extended, strategic parts coverage is provided to protect the riskiest proprietary parts and parts are delivered according the earliest next-day delivery policy. Philips guarantees 98% uptime to ensure that the system is maintained according to the highest OEM standards. It provides a strong system support with quick response times and strategic part coverage to optimize uptime and performance.

RightFit Uptime

RightFit Uptime is the premier offering within the service contract portfolio for customers for which downtime is not an option. This all-inclusive agreement provides the highest standard of service delivery and Philips guarantees 99%[1] uptime of the equipment. It includes the fastest on-site response and part delivery to ensure that engineers and parts are on-site when required. In addition, the weekday coverage is extended and more flexibility to schedule maintenance activities is offered.

[1] This number depends on the region and modality

Figure 8 shows the relative frequencies of the different contracts on the equipment. It gives an impression on which service contracts are popular and which are not.

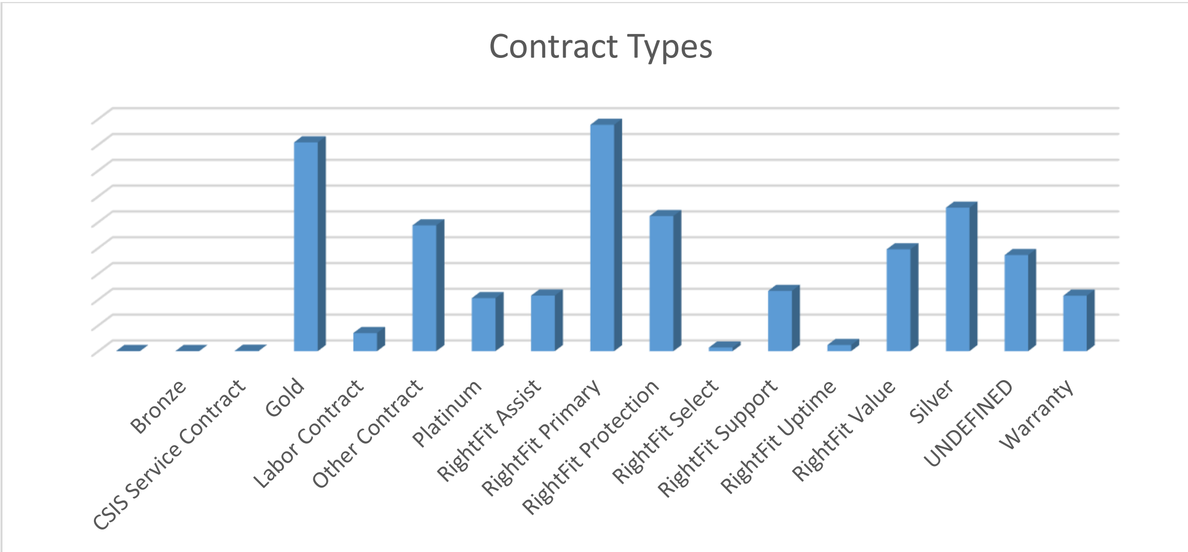


Figure 8: Contract types on equipment

The full default entitlements for each RightFit contract can be found in Appendix I. This appendix also includes the different response time options for each customer contract.

3.2 Learnings

It is important to understand the service agreements of Philips with the customers. As mentioned, every service contract provides 24/7 remote monitoring of the equipment.

The customer can also choose for options that provide coverage for specific services or parts. This can be relevant in the decision making for the RMEs. When the customer's contract covers certain service actions or parts, Phillips will pay these costs. The most important coverage option are shown in Table 2.

The first category of coverage options is coverage for labor and parts. It includes all coverage options related to actions executed by RSEs or FSEs in the LSO and coverage for part replacement. The first attribute of this category covers labor costs and travel costs of RSEs and FSEs related to maintenance activities on the customer's equipment. The second and third attribute are related to different kind of component of the system. All attributes in this category are relevant for both reactive and proactive maintenance.

The second category is called CM Service Window. It includes a time window in which Philips provides services to the customer. Examples of time windows are: Monday to Friday, 8:00-17:00 and 7 days a week 24/7. It depends on the operating hours of the customer if maintenance can be conducted outside of these operating hours. If maintenance activities are scheduled outside the working hours of the hospital, the customer incurs no loss of capacity due to proactive maintenance actions. Customers cannot treat their patients when the machine is under repair. Therefore, maintenance activities during working hours cause a loss of capacity. The entitlement for outside operating hours of the customer is noted by owh, and is set to 1 the customer is entitled for outside working hours activities and 0 if the customer does not have this right.

Table 2: Coverage options in the service contracts

Category	attribute	Incurred in case of	Type of option
Parts and Labor Coverage	Labor and Travel	Reactive, Proactive maintenance	Yes or No or Pool
	Normal Parts	Reactive, Proactive maintenance	Yes or No or Pool
	Strategic parts 1, 2, and 3	Reactive, Proactive maintenance	Yes or No or Pool
CM Service Window	Hours of coverage	Reactive, Proactive maintenance	Time window
Downtime	Downtime compensation	Reactive maintenance	Yes, No

The third category is related to unscheduled downtime of the customer's equipment. If a failure occurs, reactive maintenance is necessary. Unscheduled downtime of the equipment is compensated if the customer is entitled for this option.

As mentioned previously, some service contracts include a guaranteed uptime of the equipment. This means that Philips guarantees that a single equipment is able to run for a certain percentage of time. Maintenance policies should account for this guarantee and make sure that this service level is met. However, downtime on equipment is not always predictable by predictive models. These models only predict specific failure modes of components of the system while the uptime guarantee applies for the complete system. This makes it difficult to incorporate it in the maintenance decision model where the focus is on making the optimal decision on an alert.

4 PREDICTIVE MODELS

"Prediction is very difficult, especially if it's about the future." – Niels Bohr

Several predictive models are created that aim to predict upcoming failures of the equipment in the near future. Data scientists in Philips Research create these models. Currently there are 24 predictive models in use for iXR and 14 predictive models for MR. There are more models created but they are not deployed yet. Each predictive model tries to predict failures in a specific component of the equipment.

4.1 MODEL DEVELOPMENT

As mentioned, Data scientists from Philips Research develop predictive models. A RME joins the model development process such that the RME understands the model because they will use the model.

The model development methodology used in Philips is based on the method described in Sipos et al. (2014) and employs machine-learning techniques. To start the model development process, historical service data and daily equipment log data is collected for a target component. The log data contains so-called error messages that can also be referred to as events. Software on the system generates these log files. Examples of error messages are shown below:

- XSC: CLM flow switch opened
- Application error: Unable to communicate with GEOIPC. SID is unknown and no movements are available.

A log file can contain data on thousands of such error messages. In the daily log file, data is stored on how many times an error message occurred that day. The service data that is used contains information on part replacements. In addition, the date of the replacement is included in the service data. Data scientists make the assumption here that the part replacement is the consequence of a component failure. The consequences of this assumption will be discussed later.

Two data pools are created. In the 'bad' data pool, log files are collected that are generated on a time interval prior to a part replacement. In the 'good' data pool, log files are collected from days that are not in the interval. The predictive maintenance problem is to construct a binary classifier for predicting failures based on new equipment log data. The methodology to construct this classifier is based on a Support Vector Machines (SVM) algorithm. This algorithm aims to assign weights to the features in log data such that the training data is separated maximally. These weights represent a so-called hyperplane that tries to separate prior failure data from good data. A new data point is classified as prior failure when it is on the prior failure side of the hyperplane. Using this classification method, the model tries to predict future equipment failures based on equipment log data.

The data scientists can tune the model by constructing the hyperplane. They can select a hyperplane such that the model for example has a higher precision or recall. There is always a trade-off between the precision (% of correct alerts) and recall (% of discovered failures). A higher recall leads often to more false alerts and thus a lower precision. According to Interview 2, Appendix II the hyperplane is currently constructed according to the following optimization problem:

$$Max\ Precision = \frac{TP}{TP + FP}$$

s.t.

$$FP < 1\%$$

The parameters of the confusion matrix are calculated based on the training data. According to this optimization problem, the data scientists do not take into account the number of failures missed by the model.

4.2 IMPERFECTNESS OF THE PREDICTIVE MODELS

The predictive models employed are not 100% reliable. Not every alert raised by the model is true and the model can miss a failure. The next sections create understanding of why models are imperfect and how this imperfectness can be measured.

4.2.1 Sources of unreliability

As mentioned earlier this chapter, service data of part replacement is used and counted as a failure. However, it is also possible that other maintenance actions are conducted to solve a problem with the equipment of the customer. A frequently occurring maintenance action is calibration. When calibration is executed on the equipment, the log data should belong to the prior failure class. However, such maintenance actions are not included in the service data that is used in the development of predictive models. Therefore, it is possible that log data prior to a calibration action, is considered as good instead of prior failure data. A possible consequence of this situation is that the actual FN rate of a model is higher than calculated in the confusion matrix of the model. This is because this matrix is calculated based on test data that takes only into account the part replacements.

There are also other reasons for imperfectness in the predictions of the predictive models. These are due to uncertainties of different aspects related to the development of the models by data scientists. According to Interview 2, Appendix II the quality of the data used in the model development is not always of a good quality. Values can be missing or wrong due to various reasons. In addition, maintenance decisions are made by humans. Different decisions for similar cases can result in less reliable data. Also, the customers can act differently. Some customers can choose to 'live' with minor problems, while others call Philips immediately when they experience that there could be something wrong with the system. These differences can result in ambiguities in the data. It also happens that there is not that much useful data available. Using such small sample sizes makes it difficult to get reliable results in the model development. Larger sample sizes make it more likely that the reality is captured more adequately.

4.2.2 Measuring imperfectness

It is important to measure the imperfectness of the models because it can have huge impacts. False alerts can result in unnecessary service actions. From the other side, missing failures can result in downtime of system with high potential impact for the customer since patients' treatments are delayed as result of the downtime of the system. The imperfectness of the predictive models is currently measured by a confusion matrix as described in Section 1.3.1. It shows the relation between the predicted class and the actual class.

The matrix is computed by using the training and test data used in the model development and is not updated with new data after the deployment of the model. This means that possible anomalies in the trainings and test data can make the actual performance in the field deviate from the calculated confusion matrix. For example, the calculated precision in the model development can be lower in reality. Therefore, the aforementioned uncertainties do not only affect the reliability of the predictive models, they also affect the reliability of the confusion matrix.

The imperfectness of the predictive models is partly captured by the confusion matrix. The models also generate an imperfectness measure for each alert called the *confidence*. This confidence is typically a number between 0.5 and 1. It represents how confident the model is that the new data point belongs the failure-prone class. In other words, it says something about the distance of the new data point to the hyperplane. The bigger the distance, the higher the confidence of the alert.

The confusion matrix of a predictive model is currently not taken into account by RMEs. This matrix is not accessible for RMEs so they do not know the theoretical performance of the model. They only know that the FP of the predictive model is smaller than 1%. Since the FP is very small, the alerts are very likely to be valid.

RMEs also do not take into account the confidence of alerts in their decision-making. This number is visible for every alert, but it is not clear to them how they should interpret the number. Therefore, the RMEs do not use the alert confidence at all in their decision-making.

Currently, RMEs are not using objective measures of the imperfectness of the predictions in their decision-making. The two existing measures (confusion matrix and alert confidence) are both not used. When the RME needs to make a decision upon an alert, they only make a subjective judgement of how good the model is. Trust in the model is the leading factor in this judgement. However, the level of trust can differ among RMEs so some RMEs have more trust in a model than others do.

4.2.3 Probability of that alert is true

It is not very easy to access if an alert is true or not. RMEs can identify if an alert is false and caused by various reasons by checking operational information to reject alerts. For example, sometimes an alert can be considered as false if the FSE is on-site. When the FSE is on-site, it his very likely that he triggered the predictive model. This can be considered as an *operational error*. Operational errors induce false alerts resulting from the operational status or activities. Some of these alerts are AutoSNARed but others can be identified by checking some parameters.

Another type of errors that can result into false alerts are the *model errors*. Such errors are the result of the imperfectness of the predictive models, which is discussed earlier. Such false alerts are more difficult, if not impossible, to identify with a success rate of 100%. Currently, RM tries to identify such false alerts by making a subjective judgement of 1) the predictive model, 2) previous alert occurrences and 3) the log files.

The confidence of an alert is not taken into account by the RMEs because they do not know how to interpret this number. As mentioned earlier, the confidence is a measure of the distance of new data point (prediction) to the hyperplane that classifies data into good or failure prone data. The higher the distance the more likely it is that the classification is true. Therefore, the confidence of an alert can give valuable information on the probability that an alert is true. Platt (1999) provides more information on how probabilistic outputs for SVM can be obtained. He presents a method for extracting the probabilities $P(\text{class}|\text{input})$ from the outputs of SVM.

The judgement of an alert requires a high level of knowledge of the predictive models and deep understanding of the log files. Without this understanding, it is very difficult to access the probability that an alert is true. Therefore, it would be valuable to develop a method that accesses this probability in a more objective way. The goal is to make sure that the judgement of alerts does not rely solely on the knowledge and experience of the RMEs.

The required input parameters for the method that should be developed are:

- Confusion matrix of the predictive model
- Confidence of the alert
- Judgement of the log file corresponding to an alert
- Number of previous alerts

The development of this method is, due to its complexity and time restrictions, not included in the scope of this thesis. We only mention which parameters can be used to access the probability that an alert is true.

5 ANALYSIS CURRENT MAINTENANCE POLICY

"In God we trust, all others must bring data." – W. Edwards Deming

This chapter contains the analysis of the current maintenance policy. The required data is extracted from Vertica and analyzed in order to create valuable insights on the performance of the current maintenance policy. Section 5.1 contains an analysis of the costs relevant in the maintenance decision making. In Section 5.2 data regarding spare part decisions are analyzed. We analyzed different aspects related to time in Section 5.3.

5.1 COST ANALYSIS

This section analyzes the costs of the current predictive maintenance policy. First, the relevant cost factors are identified from the data, then the actual values are calculated for each of the factors. This provides information that give valuable insights for RMEs to account for the expected maintenance costs in their decision making.

5.1.1 Cost factors identified from data

The 'iXR_gdwhcv_jobs' table in Vertica contains information on the service actions along with its related costs that are conducted to solve a case. Three relevant data fields in these table are 'CostCm', 'CostRemote', and 'CostTravel'. These costs factors are the result of hours worked on 'maintenance or diagnostics', 'remote service' and 'traveling' respectively. Another relevant data field is 'CostOther'. These costs are the result of either 'technical support', 'waiting/delay', 'internal travel time', and 'picking or docking of a part or tool' and are directly derived from the hours spend on the activity. These hours are also stored in the same table, which allows us to calculate the costs per hour for each of these three costs factors. Note that these costs per hour depends on different factors like the region and modality.

Another important cost factor is the costs of a spare part. These costs can be found in the table 'iXR_gdwhcv_parts' in Vertica. This table contains data on which and how many parts are ordered during a job of maintenance. Two data fields in this table relate to the costs of a spare part. One field is called 'CompPrice' and the other is 'CompCost'. 'CompPrice' is the price of a single spare part ordered while 'CompCost' is the total costs of the ordered spare parts of a certain part. So, 'CompCost' and 'CompPrice' only differ when in one order two or more identical spare parts are ordered.

The aforementioned costs are the costs that can be billed to the customer depending on the service contract of the customer. Table 3 shows the cost factors that are identified from the table 'iXR_gdwhcv_jobs'.

Table 3: Maintenance cost factors

Cost factor	Activities	Description
CostCm	Diagnostics	Costs of diagnosing the problem of the system
	Corrective Maintenance	Costs of conducting the actual maintenance on the machine
CostRemote	Remote Service	Costs of providing remote service to the customer
CostTravel	Travel	Costs of traveling
CostOther	Technical support during CM	Costs of providing technical support during corrective maintenance actions (diagnostics or the maintenance itself
	Waiting/delay	Cost incurred because the engineer has to wait or is delayed
	Internal travel time	Costs of internal traveling. This includes for example traveling to a part pick-up point
	Picking or docking of a part or tool	Cost of activities related to the picking or docking of a part or tool.
CompCost	Costs of ordered parts	Costs of the ordered parts for maintenance actions. These include both used as unused parts.

Whenever an FSE or RSE in the local service market spend time on one of the factors for a case, he has to register the hours in an information system. In addition, he has to select the factor to which the hours spend belong. In this way, the costs of a maintenance case are stored in the information system for each factor. Since, these costs are entered in the information manually, it is sensitive to varying practices used by engineers. For example, some FSEs enter remote service as 'Remote Service', while others enter it as 'Technical support during CM'. Only information on values for the cost factors can be found in the data and not on the specific activities.

5.1.2 Analysis of costs

A dataset is extracted from Vertica to analyze the costs of acting on an alert. All alerts are selected that result into a case and coupled to the case in table *catmasterlist*[2]. For 4,505 alerts, the alert action is defined as "case". However, for some of these alerts, no CaseId is defined which makes it hard to couple it to a case. Besides that, some alerts contain a case number that is not valid (i.e. a CaseId of 9999999 or

[2] Note that data on iXR and MR are stored in different catmasterlist tables. Data from both tables is used.

0000). A CaseId can be considered as valid if the first three digits are 010. From the 4,505 alerts, 3,126 alerts contain a case number that starts with 010.

Now all the alerts are selected with a valid CaseId, the corresponding case should be coupled to the data of the alerts. Cases defined in the catmasterlist contain information about the costs made for that specific case. Therefore, the case from catmasterlist is coupled to the alert based on identical case id's. Note that the case id is called CaseId in the table with alerts and CaseID in the catmasterlist. Also, it should be taken into account that the CaseID in the catmasterlist is the same as the CaseId in the table with alerts without the first digit. Running the new SQL code, the set of 3,126 alerts reduces to a set of 1852 alerts coupled to a case. Other alerts cannot be coupled to a case in the catmasterlist and thus corresponding costs cannot be found. A possible reason for this reduction is that Vertica is a combined database, which is not directly coupled to the other databases. Data needs to be transferred to Vertica but it is not exactly clear when this is done. This can be a reason that some data on cases is missing in the catmasterlist.

As mentioned before, the costs are calculated by multiplying the hours spend on a cost factor with an hour tariff. Since this tariff differs per region, the choice is made to analyze the hours instead of the costs. For each model, the average hour per cost factor is calculated. Figure 9 shows these numbers for each iXR model and Figure 10 show these number for each MR model. The labels on the y-axis contain the names of the predictive models.

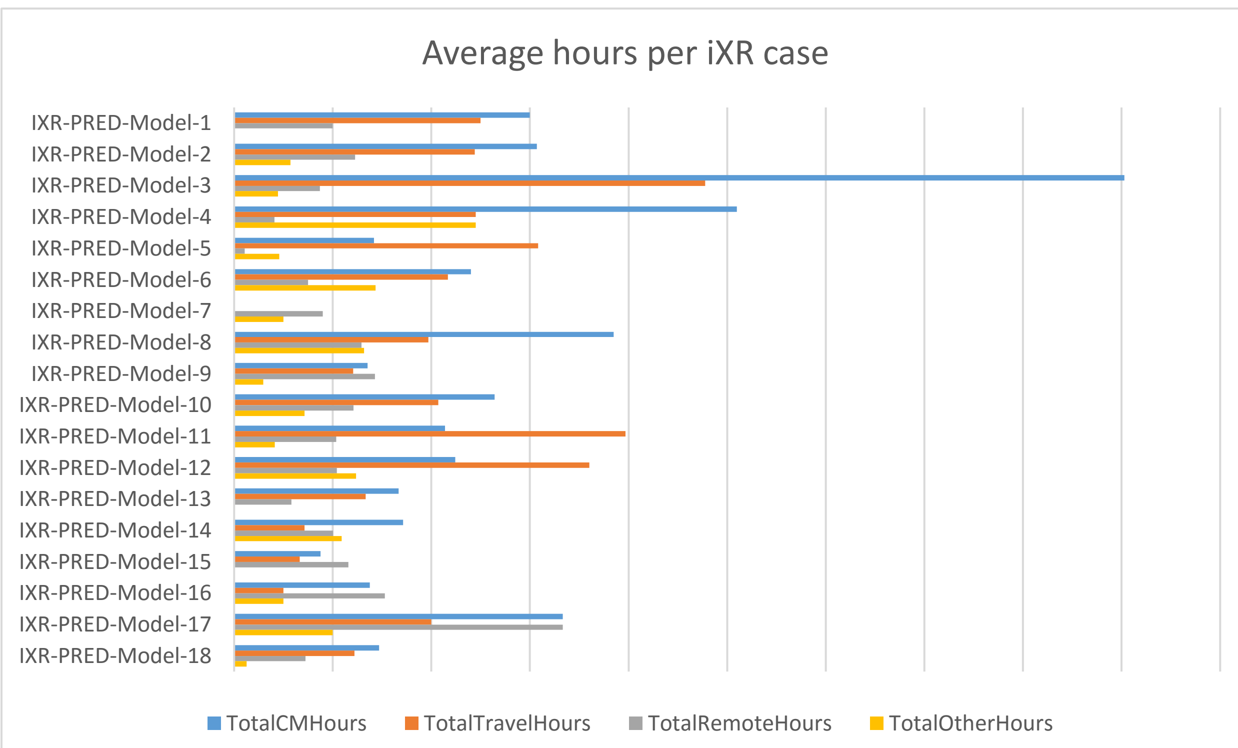


Figure 9: Average hours per cost factor for each iXR model

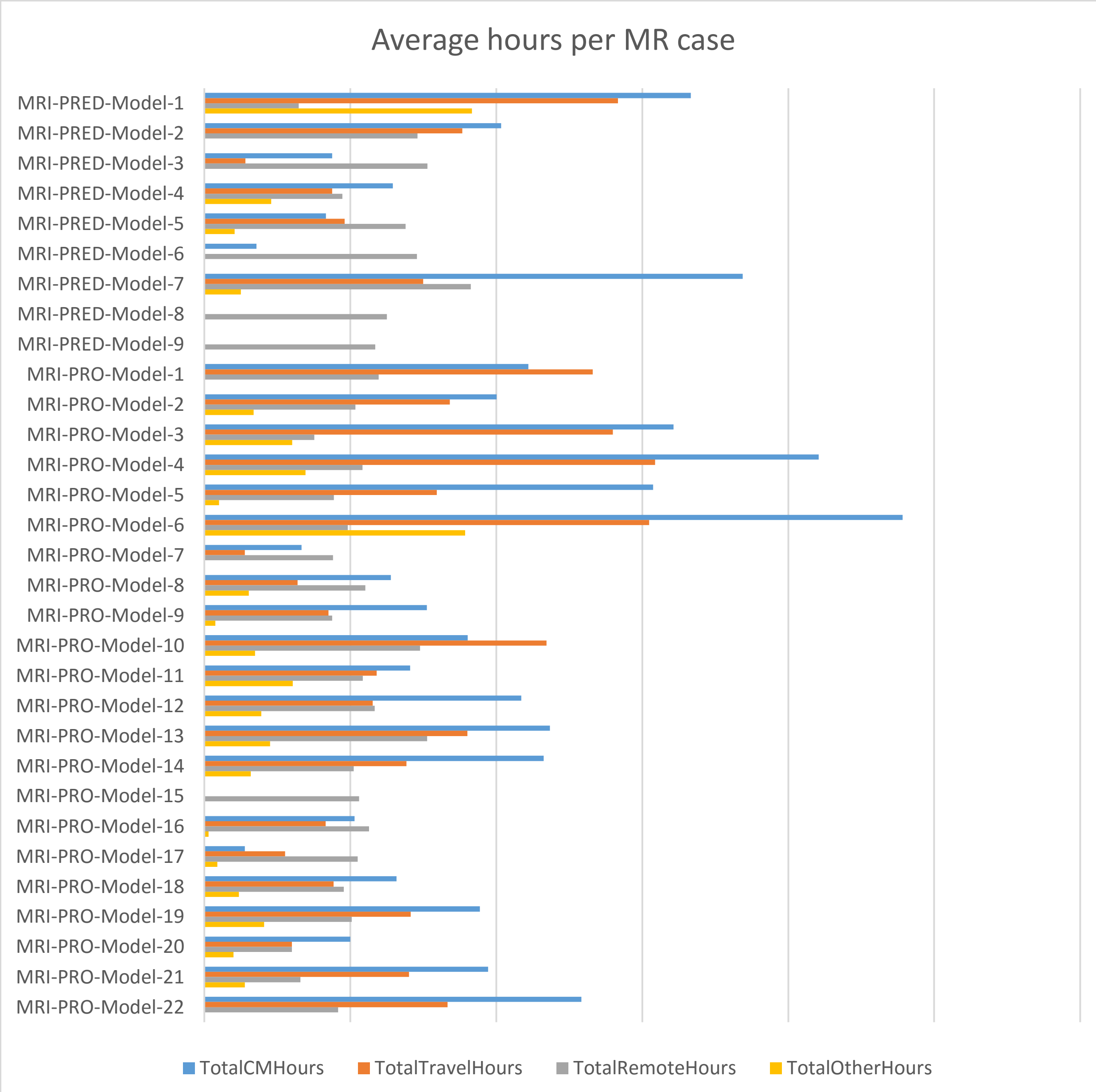


Figure 10: Average hours per cost factor for each MR model

Both figures show that the TotalCMHours is for almost every model the leading cost factor in a case. Also the TotalTravelHours are relatively high compared to the others.

As mentioned, the previous averages are based on a limited number of cases. Therefore, these averages do not provide a valid claim on the costs of future cases. The average values over all predictive models are shown and compared to the FN costs in the next section.

5.1.3 Analysis of costs of FN

Currently, the costs of missing a failure are not known. This is one of the reasons that the rate with which failures are missed is not taken into account during the predictive model development. There is no objective measure of the importance of this FN rate. In failure predicting, the importance of FN can be measured by its costs.

5.1.3.1 Identification of FN cases by safety questions

When the customer calls with a problem, a person in the LSO needs to ask some safety questions to the customer. These questions are:

1. Was the device in clinical use at the time the issue was discovered?
2. Was any patient or user harmed?
3. If the device has alarm/alert capability, did it alarm/alert as it should have at the time the issue was discovered?
4. Was this an out of box failure?

The answers given by the customer are stored in larger text field with information about the maintenance activities conducted to solve the case. With Question 1 and 2, the LSO asks if the problem with equipment occurred during the treatment of the patient and if the patient was harmed due to the problem

Question 3 checks if the problem should have been predicted by a predictive model. If the answer is yes, the predictive model misses the failure and the case can be considered as a FN. Analyzing the costs of these cases will reveal the costs of FN.

The answers on the questions are not stored in a separate data entry in the database. The answer on safety question 3 can be found in a bigger text entry in the data labeled as 'CustomerComplaint'. It contains among others information on all interactions of the LSO and the customer. The answer on question 3 needs to be extracted from this text entry in the database table. Because there are ten thousands of cases, it is not possible to find these cases manually. Therefore, we wrote a code in R to find the answer on question 3 for each case. This code can be found in Appendix III

The code aims to isolate and capture the text between question 3 and 4. In this way, the answer on question 3 is extracted from the text. We execute the code on a dataset of 10000 iXR cases. For 6608 cases, the code was able to extract the answer on question 3. The remaining cases did not contain information on this question, or the code was not able to identify the answer.

In the 6608 cases, the answer on question 3 was always "N/A". This suggests that engineers in the LSO never answer this question properly. This can be due to the newness of remote monitoring and engineers in the LSO do not know immediately, if the predictive model should have predicted the problem. Therefore, it gives no valuable information on which cases can be the result of a FN.

5.1.3.2 Identification of FN cases

The previous method of identifying FN cases was not successful because the required question is not answered correctly. Another way of identifying such cases is to look at cases of which the alert came too late. This means that the customer calls before the predictive model generates the alert. Practically, this means that the model missed the problem with the customer's system. Therefore, we can consider the case as a FN case. If the cases are identified, the average costs of such cases can be calculated.

Alerts that were generated too late, are SNARed by the RMEs. The RME has to select a reason for every alert that he SNARs manually. For late alerts, the RME selects the SNAR reason 'Late alert'. So, late alerts can be identified by selecting alerts that contain 'Late alert' as SNAR reason.

Since, such alerts are SNARed, they are not coupled to a case such that the costs of the cases cannot be identified immediately. After a SNAR, the RME does not fill in a case id. The only way to connect a case to a late alert is to couple cases conducted on a system to a late alert on the same system. This results in instances where many cases are connected to one alert since it connects all cases ever executed on the system to a late alert. In order to create more accurate connections, only cases are coupled to late alerts that are created up to 30 days before the alert is generated. Furthermore, we only use cases classified as CM and which are not preceded by an alert. Although this increases the chance that the right case is coupled to an alert, it does not guarantee that the right case is connected to a late alert because it is not linked directly. The SQL query we used can be found in Appendix IV. We exported the resulting data to Excel. In Excel duplicate cases and/or alerts are removed from the data. We only use the alert-case combinations, the smallest differences in time. This results in a dataset containing 668 usable cases.

The average hours spend per cost factor are shown together with their 95% Confidence Interval (CI) in Figure 11. We calculated the CI by:

$$95\%\ CI = 1.96 \cdot \frac{\sigma}{\sqrt{N}}$$

With σ the standard deviation of the average hours per case and N the size of the data.

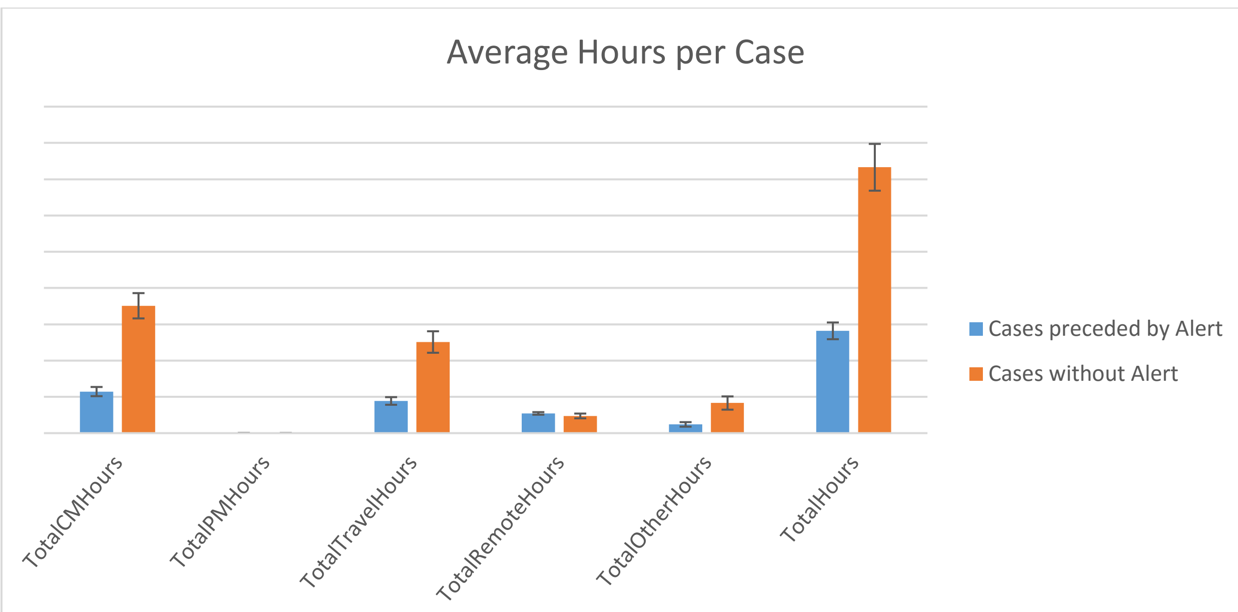


Figure 11: Costs of FN cases versus TP cases

5.1.4 Analysis of costs of FP

This sections deals with the costs of a false alert. We expect that it is quite hard to find cases of which the alert is classified as false. This classification can easily be found in the data but since the FP rate of the predictive models is very low (see Chapter 4), we expect just a few of such cases. We find 52 FP cases by running the SQL query for FP cases from Appendix IV. In 28 of those cases, there is no information about costs stored, which leaves us with 24 useful cases. The average costs of these FP cases are shown in 5.1.4.

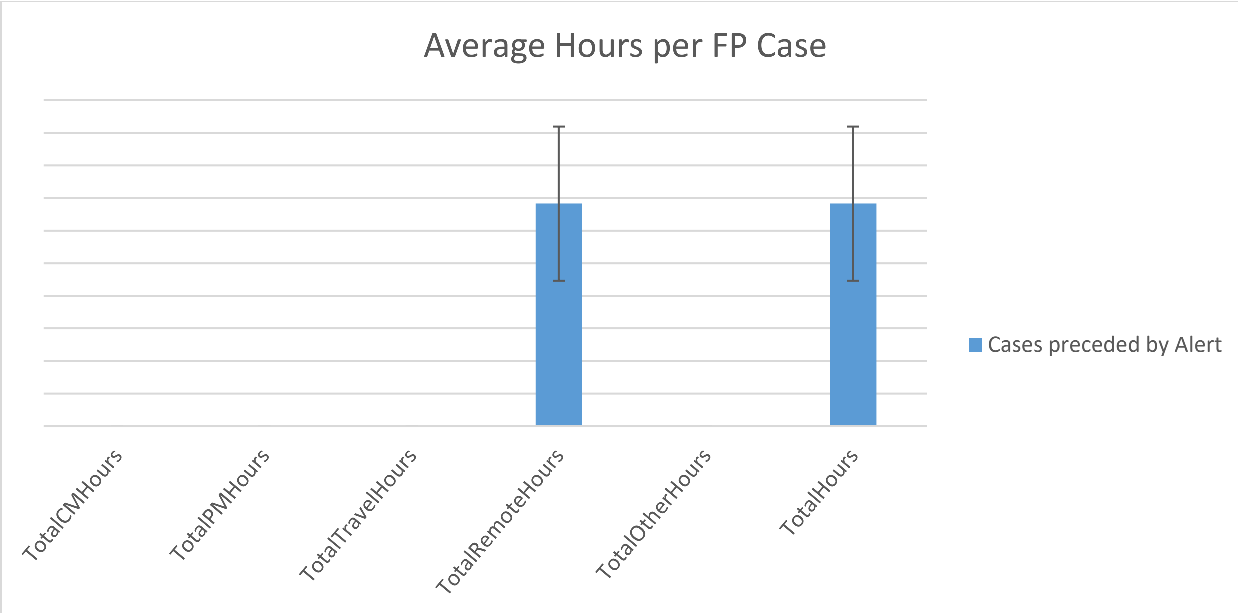


Figure 12: Average hours per cost factor for FP cases

In all 24 cases, the LSO only spend hours on remote services to find out that the alert and its case is false. The LSO never dispatched an FSE to the customer in these cases.

5.2 Analysis of spare part decisions

Currently, the decision of sending a spare part to the customer as action upon an alert is not made by RM. This decision is made by someone from the LSO. It depends on the region who in the LSO is responsible for this decision. First, it is investigated how many times spare part are send to the customer based on failure prediction. Second, two performance indicates are calculated related to this decision in order access the performance of the current way of working. These two performance indicators are the % of cases fixed in one visit and the number of unnecessary shipments.

Alert data is extracted from Development.ISDA_model_output_alert. Service data for iXR and MR are extracted from Development.iXR_fdvsv_catmasterlist and Development.MR_fdvsv_catmasterlist respectively. The data is analyzed in Excel.

5.2.1 Spare part decisions

As mentioned in chapter 4, the predictive models are build based on data on part replacements. This suggests that these models try to predict when a system requires a part replacement. However, the maintenance case resulting from alerts often do not include a part replacement. For each predictive model, it is calculated how many cases include a spare part replacement. This gives insights in how frequently a spare part is required to solve the maintenance case. Figure 13 shows the percentages of iXR cases where spare parts were ordered and Figure 14 shows these percentages for MR cases. The figures also show how many cases are used for each predictive and proactive model. These numbers are represented by N.

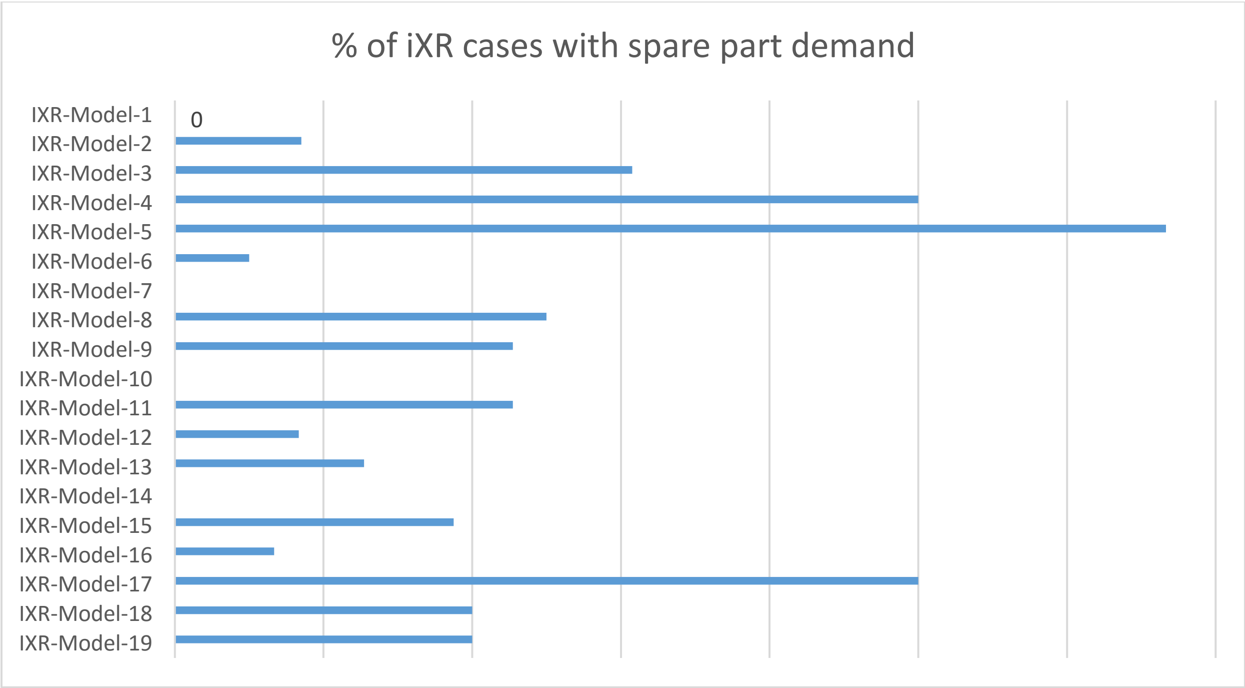


Figure 13: Percentage of cases with parts for each iXR model

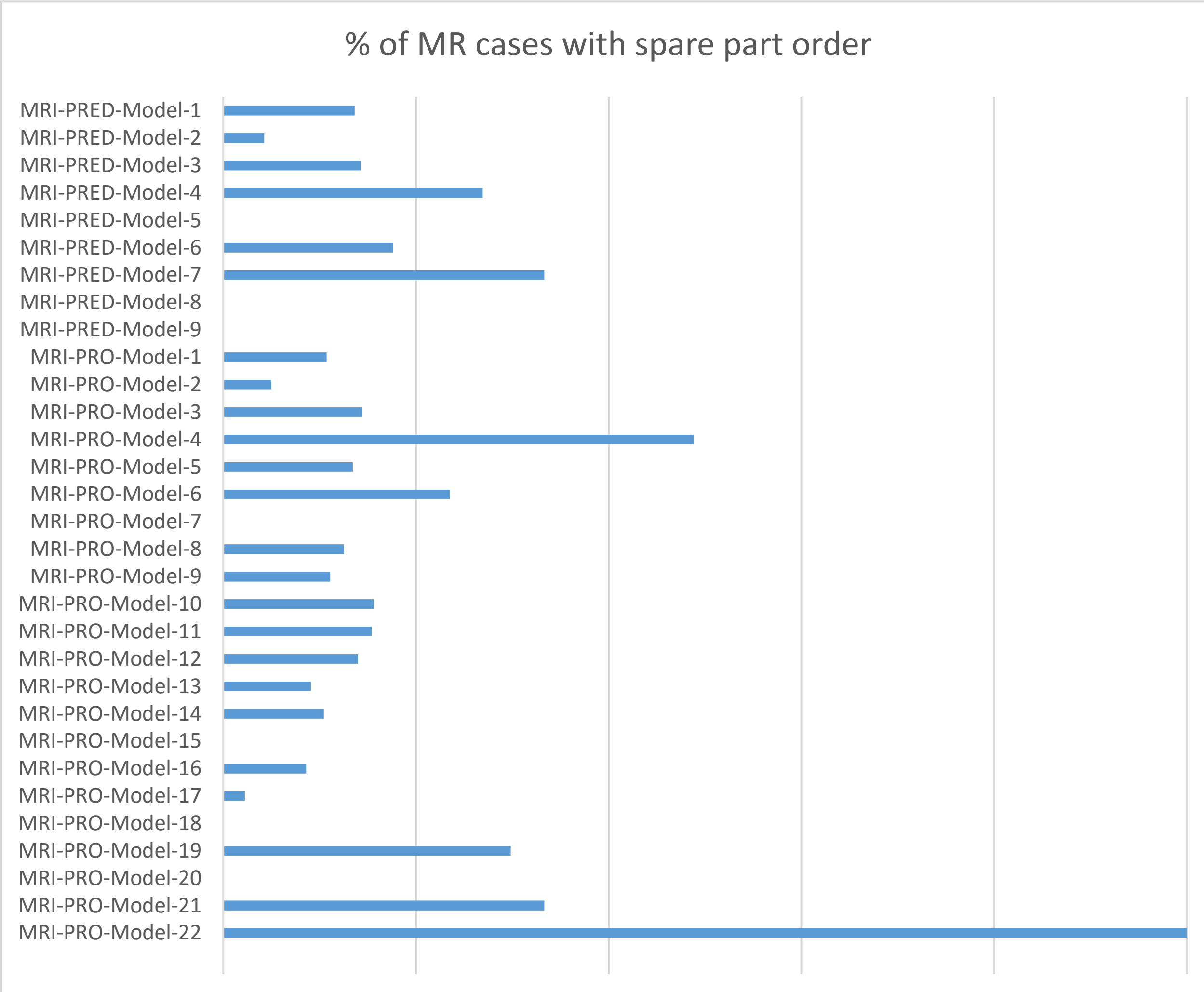


Figure 14: Percentage of cases with parts for each MR model

As shown in the figures, the percentages vary from 0% to 66.67% for iXR predictive models and from 0 to 100% for MR models. These MR models include the predictive models and the proactive models. The percentages give information on how likely it is for a predictive model that a part will be ordered. However, the percentages for each model calculated with a very small data set. The number of cases for each iXR predictive model range from 2 to 47. Such small dataset makes it very difficult to make good and valid claims about the likelihood that a part will be ordered during a case. Besides that, cases made upon a predictive model may require different spare parts.

5.2.2 First visit fix

The purpose of the analysis is to get insights in how many of the cases, the problem is not fixed in one visit. Such cases can have impact on the service level of the customer. One data sample contains all the cases and another sample contains the data of cases where parts were ordered. The percentage of cases that are fixed at the first visit are calculated for both samples and compared. The KPI that measures this is called *First Visit Fix percentage*. In order to calculate these KPIs, tables are created that show the frequencies of numbers of visits in the data. These frequency tables can be found in Table 4.

Table 4: Frequency table of the number of visits

Number of visits	All Cases	Cases with Parts
0	769	56
1	307	92
2	45	34
3	8	8
4	5	5
5	2	2
6	1	1
7	2	2

According to this data, no FSE is dispatched to the customer in 769 cases because the number of visits is zero for those cases. However, this is not completely true. A closer look on the data reveals that the travel hours are often greater than zero in those cases. This means that a FSE travelled to the customer, while the number of visits is zero.

We can conclude from these hours that the FSE visited the customer. We assume for those that the number of visits was one. In the calculations of the First Visit Fix percentages, only cases where the number of visits box is filled are taken into account. Therefore, the First Visit Fix percentage can be defined as:

$$First\ Visit\ Fix\ \% = \frac{Cases[Number\ of\ visits \leq 1]}{Cases[Number\ of\ visits \geq 0]} \cdot 100\%$$

The first visit fix percentages of both samples can be found in Table 5.

Table 5: First visit fix percentages

	All Cases	Cases with Parts
First Visit Fix %	94%	74%

We can conclude that cases that require a part replacement, face a lower first visit fix percentage than cases that can be solved without a part replacement. This is due to the fact that FSEs can find out during his first visit, that a part is required to fix the problem. Then he needs to order it, and visit the customer another time to replace a part. If the cases where a part needs to be replaced has one visit, the FSE brought the part already with him.

5.2.3 Unnecessary Shipments

Another interesting KPI to look at is the percentage of cases with unnecessary shipments. Such shipments can lead to excessive costs and thus the number of such shipments should be low. In the data, the number of unused parts is stored which indicates how many of the ordered parts, are not used during the maintenance actions. These are send back to the warehouse and unnecessary shipment costs are incurred. To calculate the KPI, a sample is created with all cases that involve parts. The KPI is defined as the ratio of cases that contain unnecessary shipments to all cases in the sample. It turns out that 15% of the cases where parts are involved, contain unnecessary shipments. Note that in some cases multiple shipments were unnecessary. The frequency table for the number of unused parts per case is shown in Table 6.

Table 6: Frequency table of unused parts per case

Total parts unused per case	Frequencies of TotalPartsUnused
0	230
1	31
2	5
3	1
4	0
5	1
6	1
7	1

The percentage of cases with unnecessary shipments can be calculated by the following formula:

$$\%\ cases\ with\ unnecessary\ shipments = \frac{Cases[TotalPartsUnused \geq 1]}{Cases[TotalPartsUnused \geq 0]}$$

Calculating this performance indicator using the numbers in the frequency table gives a value of 14.81%. So, there are unnecessary shipments in 14.81% of the cases where parts are involved.

5.3 Time analysis

This section analyzes two random variables related to time aspects in the maintenance policy. The first variable is the time between the arrival of an alert and the actual failure. We refer to this time as the Remaining Useful Life (RUL). It represents the expected remaining lifetime of the component when an alert is generated. The second variable is the time between the creation of a case and the time of the actual on-site maintenance activities. Both times are important to access the probability distribution of the downtime and the probabilities that maintenance is conducted before the customer experienced problems with the equipment. These probabilities are necessary to account for different costs associated for being on time with the maintenance activities or too late. The RUL after an alert is analyzed in Section 5.3.1 and the time to maintenance is analyzed in Section 5.3.2.

5.3.1 Remaining useful life

The predictive models typically warn for a possible failure in a certain time interval $[\tau_l, \tau_u]$ where τ_l represents the lower bound of the time interval and τ_u the upper bound of the time interval. The time between an alert arrival and a failure is called the Remaining Useful Life (RUL). However, according to RM, the mentioned time interval is not a reliable. They often do not expect that a failure will occur in that time interval. Therefore, we need to conduct an analysis on the RUL. In the remainder of this subsection, we aim to find a probability distribution for the RUL when an alert arises.

The time between a prediction and the predicted failure is not directly observable in the data. If a case is made, maintenance is often conducted before the equipment fails. The failure time is then not observable because the failure is prevented by conducting the maintenance. In order to find these failure times, cases can be selected 1) where the failure time is earlier than the time of the maintenance, or 2) where the alert is unfairly SNARed, or 3) where the case is SNARed due to contract or regional reasons. Each of these possible selections face some advantages and disadvantages. Disadvantages of the first two selections are that the size of the data is limited. Besides that, 1) has the difficulty that it is hard to identify in the data when the customer actually called. Only the date of maintenance can be accessed easily but is does not show if that is due to a customer call or that the maintenance was scheduled on that date.

2) faces difficulties in identifying whether or not an alert is unfairly SNARed. It is possible to check if a SNARed alert is followed by a case but is hard to find out if the case is related to the SNARed alert. If a case is made when the customer calls, it is not connected to a related alert in the database. This problem is also faced by 3). Another problem faced by 3) is that an alert is SNARed immediately when it is clear that the customer has no contract or the region is not monitored. This is the first thing that is checked so those alerts can also be induced by an FSE that is on-site. In addition, there is a chance that the alert is false. This makes it difficult to relate a case to an alert. An advantage of 3) is that alerts that are SNARed due to reasons related to the customer's contract or region can be identified easily. Whenever an RME SNARs an alert, he has to give the reason to SNAR it. When the RME SNARs an alert due the previously mentioned reasons, he has to select the reason that the customer has no contract or that the region is not monitored. Such alerts can be immediately identified in the data by looking at the data field 'SNAR reason'.

The decision is made to use alerts that are SNARed because that the region is not monitored. These are easy to identify, and the first corrective maintenance case after the arrival of the alert is used to determine the time between the alert and the case on a certain equipment ID. The assumption is made that this first case is earlier predicted by the alert. This is quite a strong assumption but it very hard to check if both are related. This should be done manually by an RME since they have the required knowledge to make such judgement. The query to extract the data from Vertica can be found in Appendix IV.

This data is copied to Excel. In Excel we make sure that we only use the first alert in a time interval of 50 days to prevent the use of successive alerts. We coupled this alert to the first case opened after that alert. In this way, we create alert-case combinations.

The Remaining Useful Lifetime (RUL) X_i for alert-case combination i is calculated by taking the difference between the alert arrival date and the call open date of the case. So:

$$X_i := CallOpenDate_i - AlertTime_i$$

To fit distributions for X, we need to estimate the parameters for such distributions. Several methods exist for such estimations. In this thesis, the moment estimation is used. Using the first moment M_1 and second moment M_2, parameters of different distributions can be estimated. The moments are defined as:

$$M_1 = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$M_2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2$$

The formulas of the parameter estimations for each distribution are given in Appendix V. The actual values of the parameter estimators are shown in Appendix VI.

To check which distribution fits the data best, the empirical distribution of X_i is plotted against the theoretical distributions with the estimated parameters. This plot can be found in Figure 15.

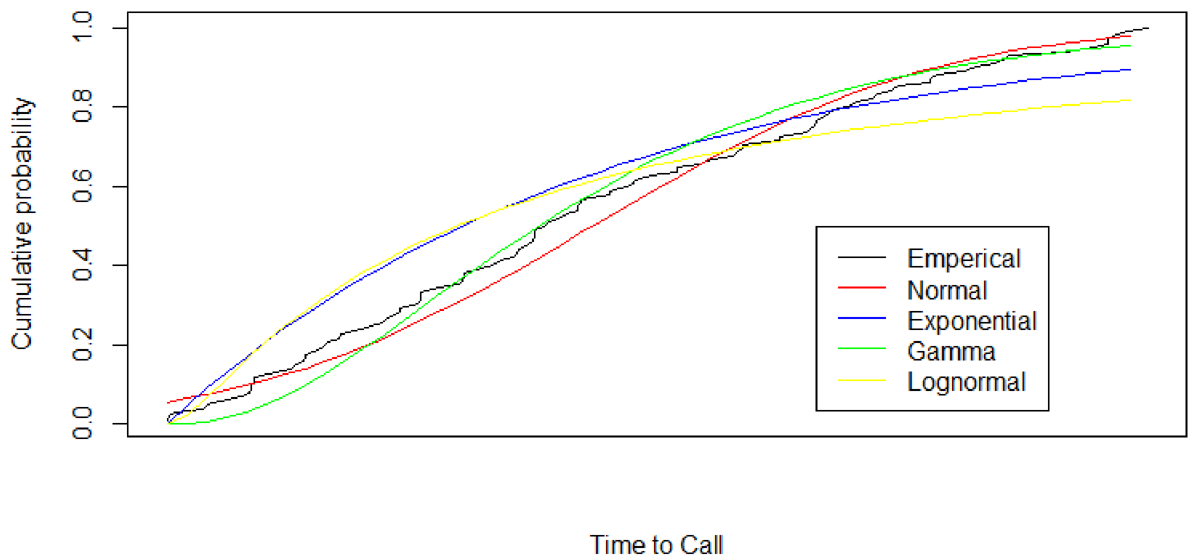


Figure 15: Plot of empirical distribution and theoretical distributions

On the first sight, it looks that the normal and gamma distributions provide a decent fit for X.

Kolmogorov-Smirnov (KS) tests for each distribution are conducted to compare the empirical distribution with the theoretical distributions with the aforementioned estimated parameters. These tests are performed in R. The results of these tests are shown in Table 7.

Table 7: KS tests for RUL distribution fitting

Distribution	p-value KS test
Normal	0.4607
Gamma	0.3615
Exponential	0.01652
Lognormal	0.00311

The N_0 hypothesis that the empirical distribution is the same as the theoretical distributions are rejected for the Exponential and the Lognormal distributions. The N_0 hypothesis cannot be rejected for Normal and the Gamma distributions, which is in line with the observations of Figure 15.

5.3.2 Time to on-site maintenance

The scheduling of maintenance activities is not part of the responsibilities of RMEs. When a case is made by a RME, the case is send to the LSO. The LSO then, is responsible to schedule the recommended maintenance activities. If the RME does not recommend to combine the maintenance with already scheduled maintenance activities, he recommend to schedule the maintenance as soon as possible. Therefore, this time to on-site maintenance can be modeled as a random variable denoted by T_{os}.

The analyze T_{os}, alerts are taken from the database that resulted in a case. Then T_{os} for alert-case combination i is determined by:

$$T_{os_i} \coloneqq OSWorkStart_i - AlertTime_i$$

The first and second moment of T_{os} are calculated by:

$$M_1 = \frac{1}{n}\sum_{i=1}^{n} T_{os_i}$$

$$M_2 = \frac{1}{n}\sum_{i=1}^{n} T_{os_i}^2$$

These moments can be used to estimate parameters to fit a distribution for T_{os}. The same parameter estimators are used as described in the previous section. The values of the parameter estimations can be found in Appendix VI.

Like in the previous section, the empirical distribution of the T_{os_i}'s is plotted against the theoretical distributions with the estimated parameters. This plot can be found in Figure 16.

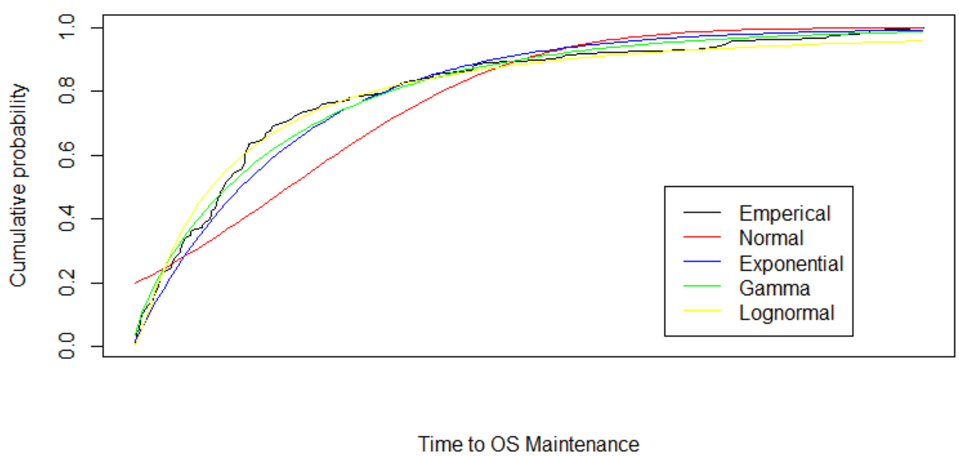


Figure 16: Plot of the empirical distribution and theoretical distributions of the Time to On-site maintenance

6 DECISION SUPPORT MODEL

"We cannot solve our problems with the same thinking we used when we created them." – Albert Einstein

The goal of this thesis was to create a model that supports RMEs in their proactive maintenance decision making. The model should support the RMEs in the decisions that follow on an alert raised by predictive models. Whenever the RME sees an alert, he needs to decide to create a case for the LSO or to reject the alert. This chapter aims to provide a mathematical model that supports the RME in making this decision. The model aims to minimize the expected costs and downtime until the alert is resolved. Section 6.1 discusses the assumptions made to create the model. Section 6.2 discusses the different parameters used in the model and Section 6.4 provides the actual mathematical model. We developed a newsvendor solution of the optimal probability threshold in Section 6.5. Section 6.6 includes a sensitivity analysis of the model on different model parameters.

6.1 MODELING ASSUMPTIONS

Several assumptions are made in order to create the decision support model. These assumptions are listed below and discussed afterwards.

1) The costs parameters are deterministic and considered as input values.
2) A customer call represents an equipment failure.
3) When a case is made, it is immediately observable by the LSO.
4) Diagnostic actions are only required when the LSO did not receive a case from RM.
5) The response time is implied by the contract of the customer and is considered to be deterministic.
6) The repair time and the diagnostic time are deterministic.
7) The failure can be fixed during the first visit.
8) After maintenance, the equipment is considered as as-good-as-new.
9) Maintenance is conducted during operating hours of the customer unless the customer is entitled for maintenance outside operating hours.
10) The probability that an alert is true is given.

Assumption 1) states that all relevant cost factors in the maintenance decision making by RMEs are known. The average values for these cost factors as calculated in Chapter 5 used initially. Later, the cost parameter values can be set such that they are customer- and predictive model-specific. Assumption 2) implies that a customer call is considered as an equipment failure. In reality, some customer may call the LSO of Philips when the performance of the equipment slightly decreased. Other may call when the system stopped working. For modeling purposes, the worst scenario is assumed which is that the system stopped working. Assumption 3) is made such that we do not have to take into account the probability that a case is made but the equipment fails before the LSO sees this case. Such situation can result in additional costs for diagnosis of the failure. This enables us to use assumption 4) which states that diagnostic actions only needs to be done when the LSO did not receive a case from RM.

Assumption 5) states that the maximum on-site response time after an equipment failure, as agreed in the contract, equals the actual response time. Assumption 6) mentions that the time required for repair of the equipment and the time to diagnose the problem, are considered as input values in the model. Assumption 7) states that any failure predicted by the predictive models can be solved in the first visit to the customer. This implies that downtime can only consists of response time after a customer call, time for diagnostics, and time for repair. Assumptions 8) is used to prevent that we should account for future failures after maintenance actions. This implies that maintenance actions are considered to be perfect.

Assumption 9) makes sure that we account for downtime because of scheduled maintenance activities. The customer cannot provide treatment to their patients during such maintenance activities because the FSE conducting maintenance on the equipment. If the customer is entitled for maintenance outside operating hours, this downtime is not incurred because customers do not treat patients during these hours. Assumption 10) assumes that the probability that an alert is true is known for each alert by the RMEs. Currently, RMEs receive no such probability value but they judge based on their experience and knowledge if an alert is true. Further research needs to be conducted to give a more objective indication of this probability to the RMEs.

6.2 MODEL PARAMETERS

This section defines the model parameters. Table 8 provide the different input parameters used in the model.

Table 8: Input parameters of the decision support model

Parameter	Notation
No FSE on site	$m \in \{0,1\}$
No open case	$o \in \{0,1\}$
Customer contract	$v \in \{0,1,\dots,6\}$
Outside working hours coverage	owh
Costs of downtime per unit time	c_d
Downtime compensation	$cc_{dt} \in \{0,1\}$
Customer's region monitored by RM	$r \in \{0,1\}$
Expected costs of PdM	c_{PdM}
Expected costs of PdM combined with PM	c'_{PdM}
Expected costs of CM	c_{cm}
Expected costs of diagnostics	$c_{diagnostics}$

SLA Response time for contract V	Tr_v
Estimated time for diagnostics	$t_{diagnostics}$
Estimated repair time	t_r
Time to next Planned Maintenance	t_{sm}
RUL	X
Time to on-site maintenance	T_{os}
Probability that the alert is true	P

6.2.1 Decision variables

The aim of the decision support model is to provide support for RM. RMEs should make the decision to either SNAR the alert, or to make a case and send it to the LSO. In addition, they have to make the decision to combine it with an already scheduled activity or not. Therefore, there are two decision variables in the model. a represents the decision to create a maintenance case and send it to the LSO and y represents the decision to combine the case with an already scheduled maintenance. The decision variables are shown in Table 9.

Table 9: Decision variables in the mathematical model

Decision variable	Notation
Initiate maintenance actions	$a \in \{0, 1\}$
Combine with next Maintenance activity	$y \in \{0, 1\}$

6.3 SITUATIONS WITH ASSOCIATED DOWNTIME AND COSTS

As mentioned in Section 5.3, the model has to deal with the two random variables, X and T_{os}. These variables measure the time from the alert arrival to the failure and on-site maintenance respectively. When the RME decides to create a case, the LSO schedules the maintenance activities to resolve the case. Since X is random, we have to distinct several scenarios when a case is created and send to the LSO.

A proactive case is scheduled on T_{os} when $a = 1, y = 0$, and $T_{os} < t_{sm}$. We assume that if the realization of T_{os} is greater than t_{sm}, the LSO makes the decision to combine the case with the already scheduled maintenance case on t_{sm}. It makes no sense to execute the case later than this moment because it will lead to an additional visit and costs. Therefore, a proactive case is scheduled on t_{sm} when $a = 1$ and $y = 1$ or when $a = 1, y = 0$, and $T_{os} > t_{sm}$.

The equipment can fail before the case is solved. This happens if $X < T_{os}$ when the case is scheduled on T_{os} or if $X < t_{sm}$ when the case is scheduled on t_{sm}. The downtime is equal to the response time plus the repair time when the equipment fails before the case is solved. Corrective maintenance costs are incurred.

The proactive maintenance case can also prevent a failure. This happens when $X > T_{os}$ or $X > t_{sm}$. The downtime is equal to the repair time and proactive maintenance costs are incurred. Costs of c'_{PdM} are incurred if the proactive case is combined with another case, and costs of c_{PdM} are incurred when the proactive case is not combined with another case. These scenarios are visualized in Figure 17.

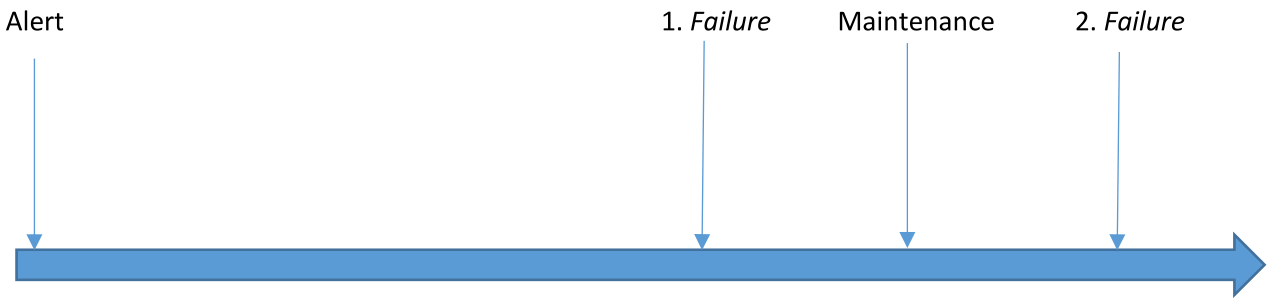


Figure 17: Timeline of alerts to maintenance or failure

When no case is made and the equipment fails, CM and diagnostic costs are incurred because the LSO did not receive a case. The problem needs to be diagnosed first because the problem is not known by the LSO, when the customer calls. In this situation, the downtime consists of response time, time for diagnostics and repair time. The costs, downtimes and probabilities for each scenario are shown in Table 10.

When the RME decides to create a case, it is always possible that the alert was false. The engineers in the LSO discover then that the alert was false. Costs of c_{FP} are incurred in such scenario.

Table 10: Characteristics of the different maintenance situations

Action	**Alert Realization**	**Probability on alert realization**	**Scenario realization**	**Probability on scenario realization**	**Costs incurred**	**Downtime incurred**
Case	True Positive	P	$X < T_{os}$, $T_{os} < t_{sm}$	$P \cdot \Pr(X < T_{os}, T_{os} < t_{sm})$	c_{cm}	$Tr_v + t_r$
			$X > T_{os}$, $T_{os} < t_{sm}$	$P \cdot \Pr(X > T_{os}, T_{os} < t_{sm})$	c_{PdM}	t_r
			$X < t_{sm}$, $T_{os} > t_{sm}$	$P \cdot \Pr(X < t_{sm}, T_{os} > t_{sm})$	c_{cm}	$Tr_v + t_r$
			$X > t_{sm}$, $T_{os} > t_{sm}$	$P \cdot \Pr(X > t_{sm}, T_{os} > t_{sm})$	c'_{PdM}	t_r
	False Positive	$1 - P$		$1 - P$	c_{FP}	0
Combine Case	True Positive	P	$X < t_{sm}$	$P \cdot \Pr(X < t_{sm})$	c_{cm}	$Tr_v + t_r$
			$X > t_{sm}$	$P \cdot \Pr(X > t_{sm})$	c'_{PdM}	t_r
	False Positive	$1 - P$		$1 - P$	c_{FP}	0
SNAR	True Positive	P		P	$c_{cm} + c_{diagnostics}$	$Tr_v + t_r + t_{diagnostics}$
	False Positive	$1 - P$		$1 - P$	0	0

6.3.1 Probability expressions

The probability expressions from Table 10 need to be defined in order to calculate the probabilities that different costs and downtime are incurred. Both X and T_{os} are random variables. The probability functions of several scenarios where downtime and costs are incurred need to be derived. First, we derive the different scenarios that can occur when the RME decides to create a case and not combine it with the next scheduled maintenance case.

The first scenario that can happen if this decision is made is when $X < T_{os}$ and $T_{os} < t_{sm}$. In this scenario, the case is scheduled on T_{os} and but the failure occurs earlier. The *Probability Distribution Function* (pdf) of X is represented by $g(x)$ and the pdf of T_{os} is represented by $f_{T_{os}}(t_{os})$. The probability that $X < T_{os}$ and $T_{os} < t_{sm}$, is represented by:

$$\Pr(X < T_{os}, T_{os} < t_{sm}) = \Pr(X < T_{os} < t_{sm}) = \int_{0}^{t_{sm}} \int_{x}^{t_{sm}} g(x) f_{T_{os}}(t_{os}) dt_{os} dx$$

The second scenario that can happen is that when $X > T_{os}$ and $T_{os} < t_{sm}$. In this scenario, the case is scheduled on T_{os} and the failure is prevented. We can use that $\Pr(X > T_{os}, T_{os} < t_{sm}) + \Pr(X < T_{os}, T_{os} < t_{sm}) = \Pr(T_{os} < t_{sm})$. Therefore, the probability that $X > T_{os}$ and $T_{os} < t_{sm}$ is be represented by:

$$\begin{aligned}\Pr(X > T_{os}, T_{os} < t_{sm}) &= \Pr(T_{os} < t_{sm}) - \Pr(X < T_{os}, T_{os} < t_{sm}) \\ &= \int_{0}^{t_{sm}} f_{T_{os}}(t_{os}) dt_{os} - \int_{0}^{t_{sm}} \int_{x}^{t_{sm}} g(x) f_{T_{os}}(t_{os}) dt_{os} dx\end{aligned}$$

The third scenario occurs when $X < t_{sm}$ and $T_{os} > t_{sm}$. Because the realization of T_{os} is greater than the time to the next scheduled maintenance, the LSO decides to combine the case on t_{sm}. In this scenario, the failure occurs before the maintenance case is executed. The probabilities that $X < t_{sm}$ and $T_{os} > t_{sm}$ are independent among each other. Furthermore, we can use that $\Pr(T_{os} > t_{sm}) = 1 - \Pr(T_{os} < t_{sm})$. Therefore, we can express the probability on this scenario realization by:

$$\Pr(X < t_{sm}, T_{os} > t_{sm}) = \Pr(X < t_{sm}) \cdot \Pr(T_{os} > t_{sm}) = \int_{0}^{t_{sm}} g(x) dx \cdot \left(1 - \int_{0}^{t_{sm}} f_{T_{os}}(t_{os}) dt_{os}\right)$$

The fourth scenario occurs when $X > t_{sm}$ and $T_{os} > t_{sm}$. The failure is successfully prevented on t_{sm}. Again, both realizations are independent among each other. Therefore, we can express the probability on this scenario realization by:

$$\begin{aligned}\Pr(X > t_{sm}, T_{os} > t_{sm}) &= \Pr(X > t_{sm}) \cdot \Pr(T_{os} > t_{sm}) = \int_{t_{sm}}^{\infty} g(x) dx \cdot \left(1 - \int_{0}^{t_{sm}} f_{T_{os}}(t_{os}) dt_{os}\right) \\ &= \left(1 - \int_{0}^{t_{sm}} g(x) dx\right)\left(1 - \int_{0}^{t_{sm}} f_{T_{os}}(t_{os}) dt_{os}\right)\end{aligned}$$

We have now defined the probabilities when the RME decides to create a case but not combine it with the next scheduled case. If we want to find the probabilities that these scenarios occur after the RME makes this decision, we need to multiply it with the probability P that the alert was true. Otherwise, these scenarios will not occur.

If the decision is made to combine the maintenance case with the next scheduled maintenance case, there are two possible scenarios. The first scenario is when the failure occurs before the case is executed on time t_{sm}. The probability on this scenario is represented by:

$$\Pr(X < t_{sm}) = \int_{0}^{t_{sm}} g(x)dx$$

The second scenario that can occur for a true alert when the combine decision is made is that the failure is successfully prevented by the maintenance. In such scenario, X is greater than t_{sm}. The probability that $X > t_{sm}$ is represented by:

$$\Pr(X > t_{sm}) = \int_{t_{sm}}^{\infty} g(x)dx = 1 - \int_{0}^{t_{sm}} g(x)dx$$

The pdf of X is given by:

$$g(x) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Since T_{os} follows a Gamma distribution, the pdf of T_{os} is given by:

$$f_{T_{os}}(t_{os}) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} {t_{os}}^{\alpha-1} e^{-\beta t_{os}}$$

With:

$$\Gamma(\alpha) = \int_{0}^{\infty} x^{z-1} e^{-x} dx$$

6.4 MATHEMATICAL MODEL FOR RM

The objective of the model will be to minimize the expected costs and downtime until the alert is resolved. It is a short-term and myopic optimization problem. After an alert arrival, the RME has to make the decision such that the expected costs and downtime are minimized. We define two objective functions. One to minimize the expected costs and one to minimize the expected downtime. The objective functions are summations of the expected costs and downtimes of the different actions. In addition, expected downtime costs are added to the expected costs function. The mathematical model is shown below and we refer to this model as Model 1:

Model 1

$$\text{Min } E[C] = a \cdot (1-y) \cdot E[C_{case}] + y \cdot E[C_{combine}] + (1-a) \cdot E[C_{SNAR}] + cc_{dt} \cdot E[D] \cdot c_d \quad 1)$$

$$\text{Min } E[D] = a \cdot (1-y) \cdot E[D_{case}] + y \cdot E[D_{combine}] + (1-a) \cdot E[D_{SNAR}] \quad 2)$$

s.t.

$$a \leq r, o, m, v \quad 3)$$

$$y \leq a \quad 4)$$

$$a, y \in \{0,1\} \quad 5)$$

1) represents the minimization objective of the expected costs as a result of the decisions made by the RME. $E[C_{case}]$ represents the expected costs of creating and sending a case to the LSO ($a = 1, y = 0$). $E[C_{combine}]$ represents the expected costs of creating a case and suggest to combine it with an already scheduled case ($a = 1, y = 1$). $E[C_{SNAR}]$ represents the expected costs of SNAR ($a = 0$). The expected downtime costs are represented by $cc_{dt} \cdot E[D] \cdot c_d$ and are only incurred when the customer is entitled for downtime compensation ($cc_{dt} = 1$).

We can calculate the expected costs of each action by summing the multiplications of scenario probabilities for that action with the associated costs. The scenario probabilities and costs can be found in Table 10 in Section 6.3. These expected costs expressions for the different actions are given below:

$$E[C_{case}] = P \cdot (c_{cm} \cdot (\Pr(X < T_{os}, T_{os} < t_{sm}) + \Pr(X < t_{sm}, T_{os} > t_{sm})) + c_{PdM} \cdot \Pr(X > T_{os}, T_{os} < t_{sm}) + c'_{PdM} \cdot \Pr(X > t_{sm}, T_{os} > t_{sm})) + (1-P) \cdot c_{FP}$$

$$E[C_{combine}] = P \cdot \big((Tr_v + t_r) \cdot \Pr(X < t_{sm}) + c'_{PdM} \cdot \Pr(X > t_{sm}) + (1-P) \cdot c_{FP}\big)$$

$$E[C_{SNAR}] = P \cdot \big(c_{cm} + c_{diagnostics}\big)$$

2) represents the minimization objective of the expected downtime. $E[D_{case}]$ represents the expected downtime of creating and sending a case to the LSO ($a = 1, y = 0$). $E[D_{combine}]$ represents the expected downtime of creating a case and suggest to combine it with an already scheduled case ($a = 1, y = 1$). $E[D_{SNAR}]$ represents the expected downtime of SNAR ($a = 0$). We can calculate the expected downtime of each action in the same way as the expected costs. The expected downtime expression for each action are given below:

$$E[D_{case}] = P \cdot \big((Tr_v + t_r) \cdot (\Pr(X < T_{os}, T_{os} < t_{sm}) + \Pr(X < t_{sm}, T_{os} > t_{sm}))\big) + t_r \cdot (\Pr(X > T_{os}, T_{os} < t_{sm}) + \Pr(X > t_{sm}, T_{os} > t_{sm}))$$

$$E[D_{combine}] = P \cdot (c_{cm} \cdot \Pr(X < t_{sm})) + t_r \cdot \Pr(X > t_{sm})$$

$E[D_{SNAR}] = P \cdot (Tr_v + t_r + t_{diagnostics})$

3) makes sure that an alert is SNARed if, the customer's region is not monitored ($r = 0$), the customer has no contract ($v = 0$), the FSE is already on-site ($m = 0$), or there is already a maintenance case opened for the system ($o = 0$). Alerts with such characteristics should be SNARed automatically.

4) enforces that a case can only be combined with an existing case when the RME decides to create a for the alert. 5) ensures that a and y can only take binary values.

6.4.1 Model output

We implemented Model 1 in R. The code can be found in Appendix XI. Since we have two objective functions, there is not always a single solution for the optimization problem. Lower costs can result in higher downtimes. The RME can take three different decisions. He can SNAR the alert ($a = 0$), create a case ($a = 1, y = 0$), or create a case and combine it with an already scheduled case ($a = 1, y = 0$). The R code aims to evaluate all three options in terms of expected costs and expected downtimes. The output of Model 1 consists of a summary of each option with the expected costs and downtime of each option. This gives the RME support in their decision making because they can account for the possible consequences of their decisions. The expected costs versus expected downtime are also visualized in a plot. An example of the model output is given in Figure 18. We used the default values for input parameters as given in Appendix VIII.

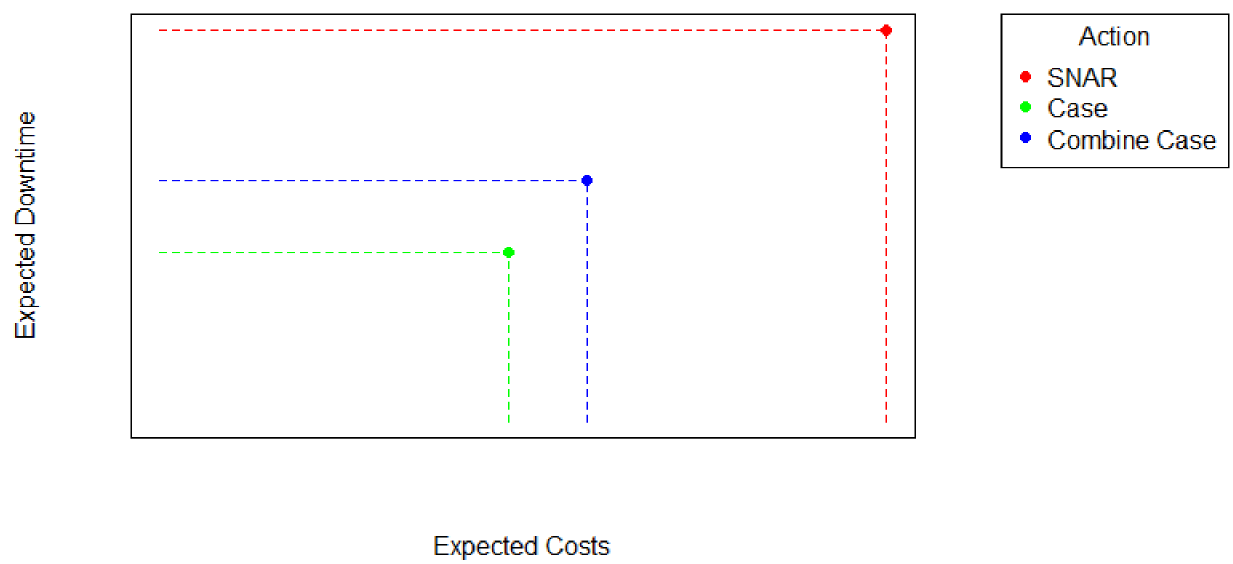


Figure 18: Example of model output

6.5 Towards a newsvendor solution

The expected costs function can also be written in a newsvendor form. An upcoming failure can be seen as the demand in a classical newsvendor problem. When there is a failure upcoming, there is demand for maintenance actions. Let us first assume that proactive maintenance activities are always conducted before the failure. In addition, we assume that there are no scheduled maintenance activities in the future yet. Then, we can rewrite the expected costs function to the following newsvendor form:

$$E[C] = c_{PdM} \cdot a + c_{cm} \cdot E[\max(K - a, 0)] + (c_{FP} - c_{PdM}) \cdot E[\max(a - K, 0)] \qquad 6)$$

c_{PdM} are the costs incurred if maintenance is done proactively based on a true alert. $K = 1$ represents that the predictive model that an alert generated by a predictive model is true. If $K = 0$ the alert was false. An alert can be either true or false. As mentioned before, the probability that an alert is true ($K = 1$) is P. We can say then that $K \sim Bernoulli(P)$. $E[\max(K - a, 0)]$ is the expected value of doing too less maintenance. $E[\max(a - K, 0)]$ is the expected value of acting on a FP alert. By using the optimality condition from the classical newsvendor solution, the criticality condition in our problem is:

$$\frac{c_{cm} + c_{diagnosics} - c_{PdM}}{c_{cm} + c_{diagnosics} - c_{PdM} + c_{FP}}$$

The optimality condition for the newsvendor problem becomes:

$$a^* = F_K^{-1}\left(\frac{c_{cm} + c_{diagnosics} - c_{PdM}}{c_{cm} + c_{diagnosics} - c_{PdM} + c_{FP}}\right)$$

This implies that:

$$F(a^*) = \Pr(K \leq a) = \frac{c_{cm} + c_{diagnosics} - c_{PdM}}{c_{cm} + c_{diagnosics} - c_{PdM} + c_{FP}} = c$$

The CDF of K is given by:

$$F(x) = \Pr(K \leq x) = \begin{cases} 0 & x < 0 \\ 1 - P & 0 \leq x < 1 \\ 1 & x \geq 1 \end{cases}$$

As shown before, the optimal action a^* can be found by using the inverse of the cumulative distribution function of K. The inverse of this function is for a Bernoulli distributed random variable, given by:

$$a^* = F_K^{-1}(c) = \begin{cases} 0, & 0 \leq c < 1 - P \\ 1, & 1 - P \leq c < 1 \end{cases}$$

The condition $0 \leq c < 1 - P$ can be rewritten to $P < 1 - c$ since P always takes values between 0 and 1. The optimal decision for the alert is to SNAR ($a^* = 0$) an alert if $P < 1 - c$.

The condition $1 - P \leq c < 0$ can be rewritten to $P \geq 1 - c$. In this condition, the optimal action is to create a case ($a^* = 1$). This means that $1 - \mathrm{c}$ serves as a probability threshold for creating a case. The optimal decision for an alert with $P < 1 - c$ is SNAR and for an alert with $P \geq 1 - c$ it is optimal to create a case.

6.5.1 Generalized Newsvendor solution

We built the previous newsvendor solution upon the assumption that proactive maintenance activities are always scheduled before the failure. If the failure is not critical, this assumption holds. Such failure leads to decreased functionality, but the customer can still use the equipment. No immediate maintenance actions are required. However, when the failure is critical, we cannot use the previous newsvendor solution because it does not account for extra costs if the failure arrives before the maintenance activities.

In order to account for such situations, we need to replace the parameter c_{PdM} with $E[C|a = 1, y = 0, P = 1]$. This represents the expected costs when a case is made based on a true alert. In addition, the case is not combined with an already scheduled maintenance case. Its value can be found by setting $a = 1$, $y = 0$, and $P = 1$ in Model 1 from Section 6.4. We set $P = 1$ to make sure that we only have the expected costs if the alert was true. If the alert is not true, c_{FP} is incurred. Assuming that $cc_{dt} = 0$, we can say that:

$$E[C|a = 1, y = 0, P = 1] = c_{PdM} \cdot \Pr(X > T_{os}) + c_{cm} \cdot \Pr(X < T_{os})$$

We can now generalize expression 6) such that Model 1 is rewritten to a newsvendor expression. We can rewrite Model 1 under the assumption that $cc_{dt} = 0$ as:

$$\begin{aligned} E[C] = E[C|a = 1, y = 0, P = 1] \cdot a + (c_{cm} + c_{diagnosics}) \cdot E[\max(K - a, 0)] \\ + (c_{FP} - E[C|a = 1, y = 0, P = 1]) \cdot E[\max(a - K, 0)] \end{aligned} \quad 7)$$

The criticality condition of the newsvendor problem becomes then:

$$\frac{c_{cm} + c_{diagnosics} - E[C|a = 1, y = 0, P = 1]}{c_{cm} + c_{diagnosics} + c_{FP} - E[C|a = 1, y = 0, P = 1]}$$

We can define the optimal probability threshold p^* as:

$$p^* = 1 - \frac{c_{cm} + c_{diagnosics} - E[C|a = 1, y = 0, P = 1]}{c_{cm} + c_{diagnosics} + c_{FP} - E[C|a = 1, y = 0, P = 1]}$$

We can also derive the optimal probability threshold p^*_{comb} for the decision to combine maintenance activities or SNAR the alert. We only need to change $E[C|a = 1, y = 0, P = 1]$ into $E[C|a = 1, y = 1, P = 1]$ such that:

$$p^*_{comb} = 1 - \frac{c_{cm} + c_{diagnosics} - E[C|a = 1, y = 1, P = 1]}{c_{cm} + c_{diagnosics} + c_{FP} - E[C|a = 1, y = 1, P = 1]}$$

The newsvendor solution is tested simultaneously with the sensitivity analysis of Model 1 on P.

6.6 SENSITIVITY OF INPUT PARAMETERS

The mathematical model from the previous section in implemented in R. The R program evaluates the expected costs and expected downtime for each of the three possible actions. This helps RMEs to see the impact of their decisions on the expected costs and downtime related to an alert. This section evaluates how sensitive Model 1 is to different values of input parameters.

6.6.1 Influence of P

Varying the input parameter P can give some valuable insights in how credible an alert should be to initiate maintenance actions. The values for the various input parameters can be found in Appendix VIII.

We vary P from 0 to 1 with a step size of 0.01. The influence of P on the expected costs can be found in Figure 19. We also plotted the newsvendor solutions p^* (the green dotted line) and p^*_{comb} (the blue dotted line).

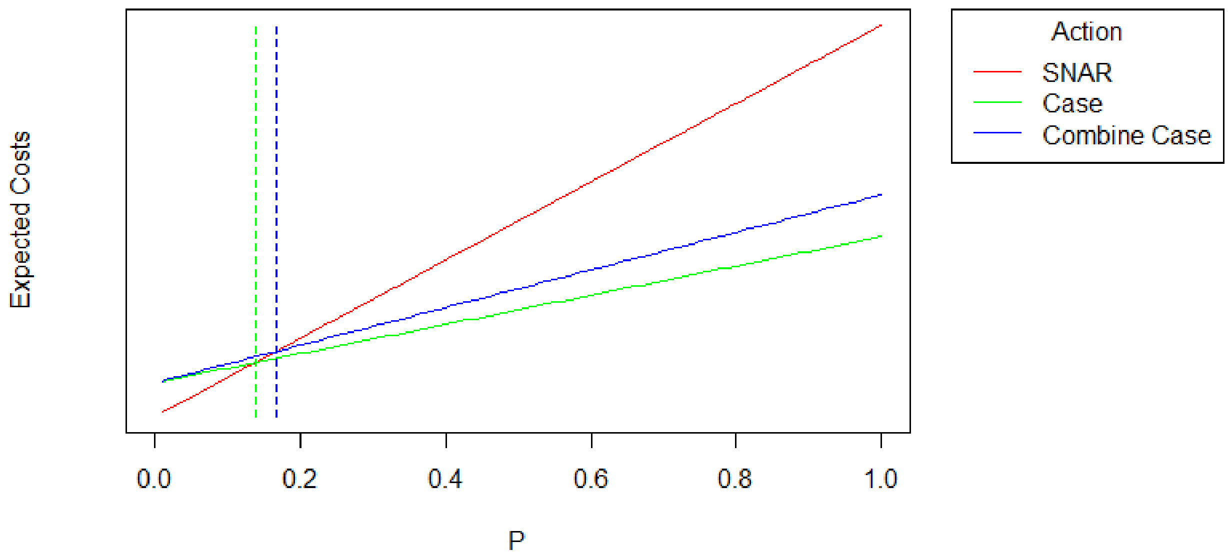


Figure 19: Influence of P on expected costs

In Figure 19, we observe that the expected costs are increasing in P for every decision. Until a certain value of P, SNAR is the best decision for an alert. For higher values of P, creating a case is the best decision in terms of costs. Figure 19 shows that this turning point of P coincides exactly with the newsvendor solution p^*. The newsvendor solution p^*_{comb} coincides exactly with the intersection of expected costs of combining the case and SNAR. Therefore, we can conclude from this figure that the newsvendor solutions provide an optimal probability threshold for creating a case versus SNAR. For the given input parameters, it is not beneficial in terms of costs to combine the case with an existing case.

Figure 20 shows the influence of P on the expected downtime under the different decisions.

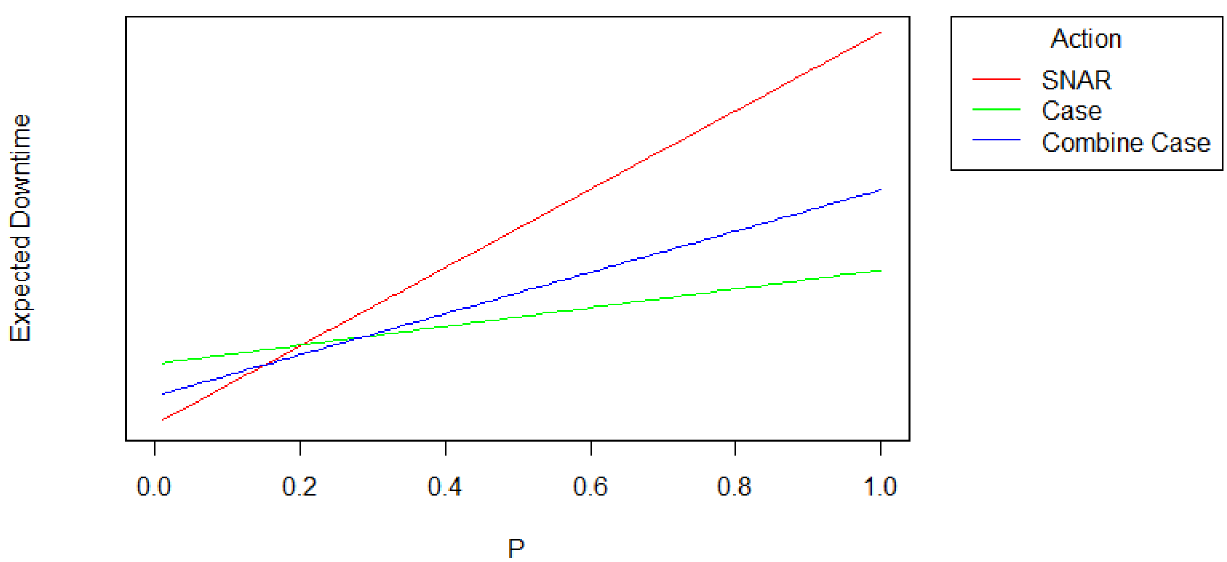


Figure 20: Influence of P on expected downtime

We can observe that the expected downtime is increasing in P for all actions. Now there are two turning points of the optimal solution. For the lowest values of P, SNAR is the most beneficial for the expected downtime. Then there is an interval for P where combining the case is the most valuable option. For higher values of P it is not beneficial anymore to combine the case.

6.6.2 Influence of t_{sm}

If RMEs think an alert is credible enough to make a case, they have to decide if the LSO should combine it with an already scheduled maintenance activity. When the RME receives an alert, he can see when the next activity is scheduled on the equipment. t_{sm} represents the time to this activity. For RMEs, it is valuable to know the maximum value of t_{sm} to combine the case with that activity. Figure 21 shows the influence of t_{sm} on the expected costs of each action. There is an intersection between the green and blue line. For values of t_{sm} lower than this intersection, it is beneficial in terms of costs to combine the case with the next scheduled case.

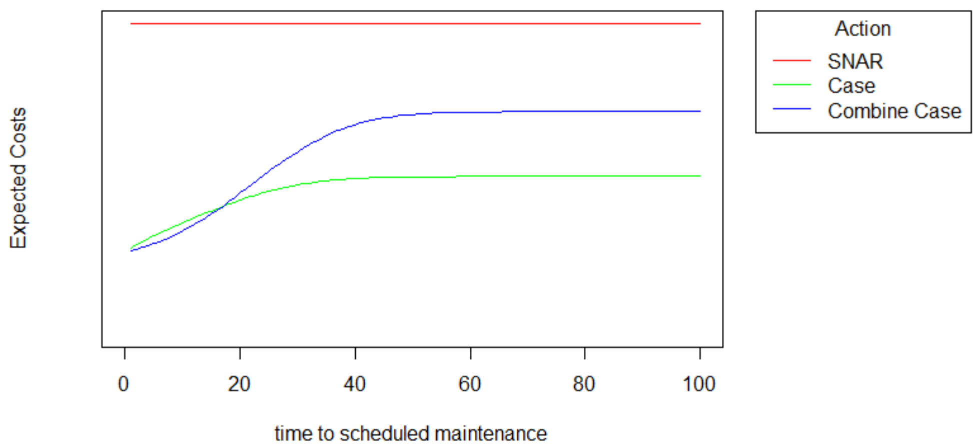


Figure 21: Influence of t_{sm} on expected costs with $P = 0.8$

Figure 22 shows the influence of t_{sm} on the expected downtime. We can observe that, if we use the default values of parameters, it is not beneficial to combine the maintenance case with the next scheduled maintenance activity.

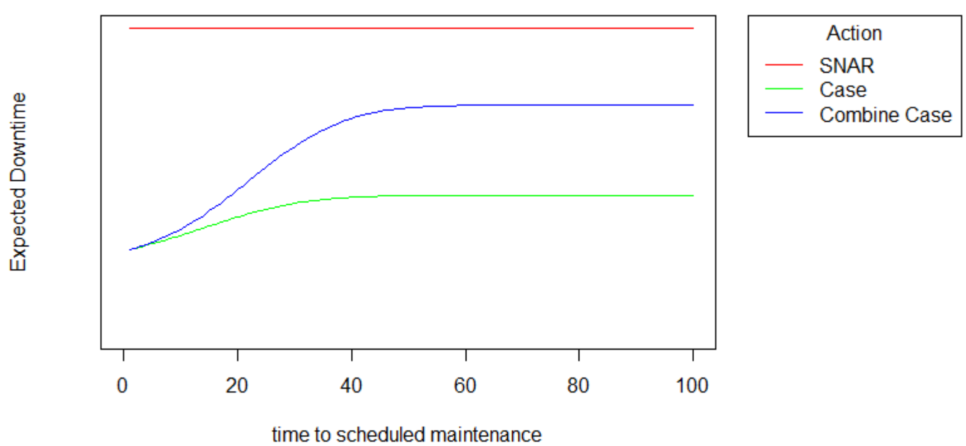


Figure 22: Influence of t_{sm} on expected downtime with $P = 0.8$

7 CASE STUDY: FLAT DETECTOR

"Science is curiosity, testing and experimenting." – Venkatraman Ramakrishnan

In the previous sections, analysis is conducted on aggregated values of parameters. However, every component in the equipment has its own characteristics with component specific failure modes and associated costs. Therefore, it is valuable to conduct the analysis for components separately by a case study. The case selected in this research is the *Flat Detector*. We only use data related to this component in this section. The Flat Detector is a critical component in iXR equipment. The predictive model 'IXR_PRED_FDXD' monitors this component and it generates the following alert: "Possible failure of the FDXD within the next 20 days". The analysis of spare part decisions is not done for this case study. The useful data for that analysis was too limited to create valuable insights in such decisions.

7.1 PARAMETER SETTING

This section aims to find the values for the input parameters in the decisions support model described in Chapter 6. This helps us to create output for the specific case of the flat detector. First, we analyzed the false negative costs of the predictive model related to the flat detector. After that, we analyzed the random variables as described in Chapter 5 specifically for the flat detector. The SQL queries can be found in Appendix IV.

7.1.1 FN costs

We use the same method as used in Section 5.1.3.2 to identify FN cases related to the predictive model 'IXR_PRED_FDXD'. We were able to identify 107 cases that we can classify as FN. However, the same errors as in Section 5.1.3.2 may be present in this data. In order to access the TP costs of the model, we found 15 cases resulting from an alert generated by the predicted model for the flat detector. Figure 23 show the comparison of the TP costs and the FN costs. The average costs per case for each costs factor is shown along with the total average costs per cases. It also contains the 95% CI of the average numbers.

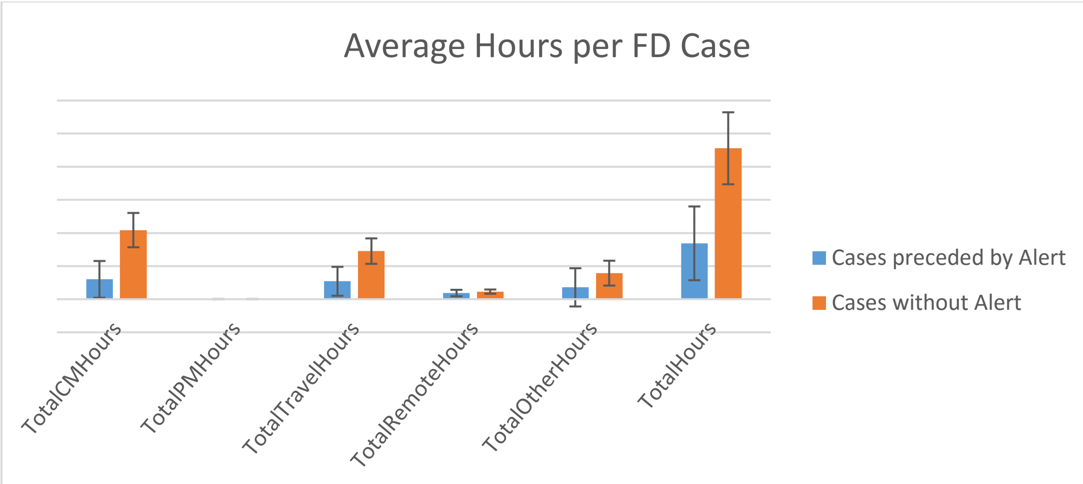


Figure 23: Average hours spend by flat detector cases on each cost factor

Figure 23 reveals that the average costs of missing an alert are higher than acting on an alert. This applies for all individual cost factors. However, the difference are not statistically significant for TotalRemoteHours and TotalOtherHours. The small difference in TotalRemoteHours can be explained by the fact that after a customer call, a FSE will go to the customer to diagnose the problem rather than the try to solve the problem remotely first. We can explain the differences in TotalCMHours by the prevention of diagnostic actions after an alert. Alerts contain already information about the failure model of the component. In other words, the problem to prevent is already diagnosed. This can also explain that the TotalTravelHours is lower after an alert. Multiple visits might be necessary for diagnostic and actual repair purposes.

7.1.2 Remaining useful life

This subsection aims to find a distribution for the RUL after an alert arrival for the flat detector. This is necessary because each predictive model has its own characteristics related to the prediction interval. We apply the method as described in Section 5.3.1 but use only data on the flat detector. First, the parameters for different distributions are estimated and second, the theoretical distribution is compared with the empirical distribution. Figure 24 show the comparison of the empirical distribution with various theoretical distributions.

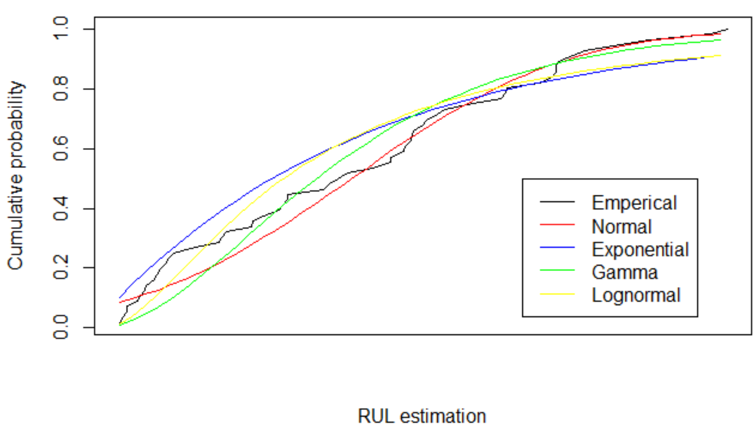


Figure 24: Distribution fitting for the RUL for FD alerts

The results of the KS tests for the different distributions are shown in Table 11.

Table 11: KS test results for distribution fitting for the RUL of FDXD alerts

Theoretical Distribution	p-value KS test
Normal	0.5539
Gamma	0.1558
Exponential	0.158
Lognormal	0.1108

The KS-tests provide significant results for all tested theoretical distributions so we cannot reject any of these distributions. The Normal distribution provide the highest value for the KS test and it looks visually to have a decent fit. This is in line with the findings in Section 5.3.1 where the normal distribution also provides the best fit for the RUL. Therefore, we do not need to adapt the probability distribution for X in the decision support model to use it for the flat detector. The values of the estimated parameters can be found in Appendix VII.

7.1.3 Time to on-site Maintenance

We now need to estimate the time between the arrival of an alert, to the time the actual maintenance starts. Because this time is random from an RM point of view, we try fit theoretical distributions on these times. We use the same method as described in Section 5.3.2 to fit these distributions on the empirical distribution. We found 16 alerts that resulted into a case with a known start time of the on-site maintenance. Figure 25 shows the plot of the empirical distribution with the theoretical distributions with estimated parameters.

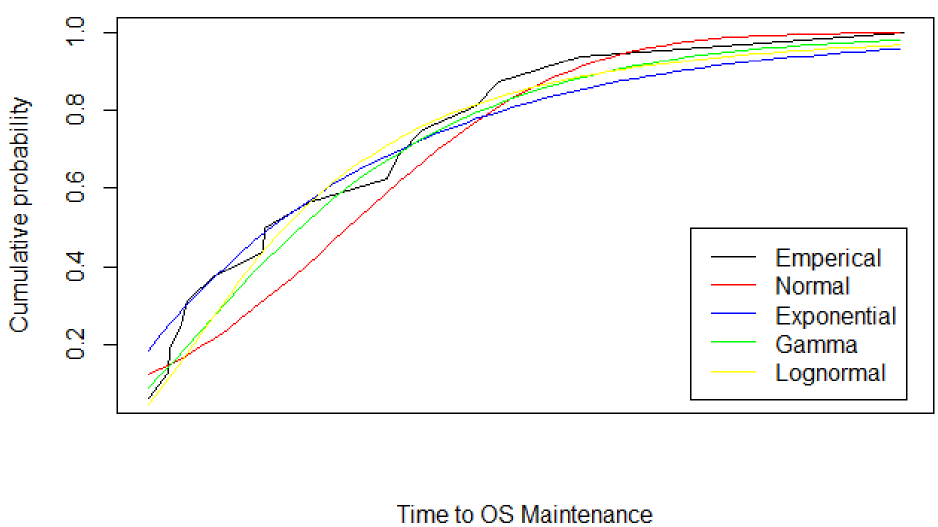


Figure 25: Distribution fitting for the Time to OS maintenance for FDXD alerts

The results of the KS tests for the different distributions can be found in Table 12 on the next page.

In the decision support model described in Chapter 6, we used the Gamma distribution to model the time between an alert arrival and the start of the actual maintenance. Figure 25 and Table 12 indicate that this distribution also provides the best fit in this case study. Therefore, we do not need to adapt the model for the flat detector. The values of the estimated parameters can be found in Appendix VII.

Table 12: KS test results for distribution fitting for the Time to OS maintenance for FDXD cases

Theoretical Distribution	p-value KS test
Normal	0.7423
Gamma	0.9643
Exponential	0.5859
Lognormal	0.8168

7.2 Model Output

Now we modeled the random variables for the specific case of the Flat Detector, we can use it in the model. In this way, we can create predictive model specific output. Using the default values as given in Appendix IX, we retrieve the plot in Figure 26.

With the Flat Detector specific default values, creating a case is the optimal decision to make by the RMEs. It outperforms the other actions in both expected costs and expected downtime.

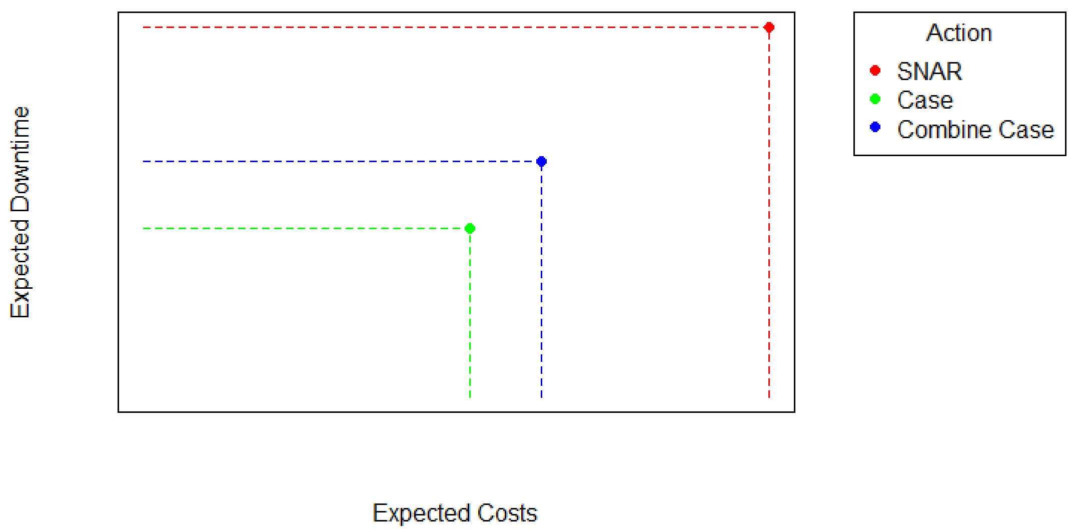


Figure 26: Model output default values for Flat Detector with $P = 0.8$

The previous plot is made for a credible alert with $P = 0.8$. If we use the same input values but set P to 0.15, we receive the plot in Figure 27. It can be seen that there is no optimal decision to make. No action outperforms all others in terms of both expected downtime and expected costs. Create a case is the best option in terms of costs while combining the case is the best option in terms of downtime.

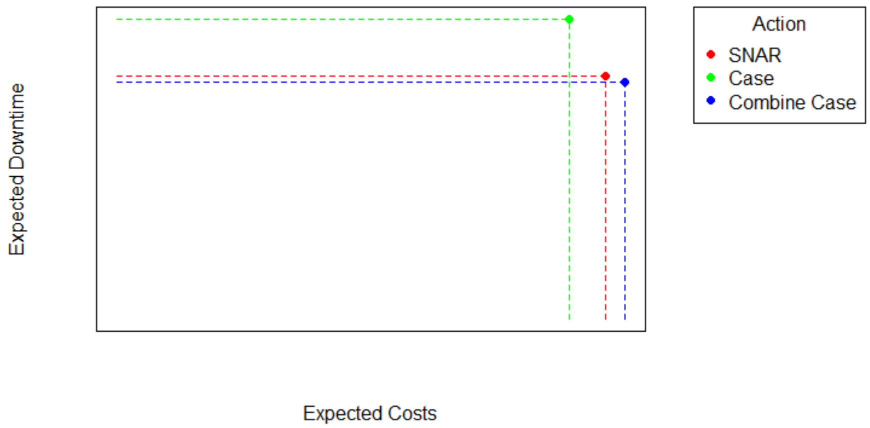


Figure 27: Model output default values for Flat Detector with P=0.15

If we vary P from 0 to 1 with steps of 0.01, we receive the plots in Figure 28 and Figure 29. We can see that the newsvendor solutions provide a probability threshold for creating a case. This is in line with the results in Chapter 6.

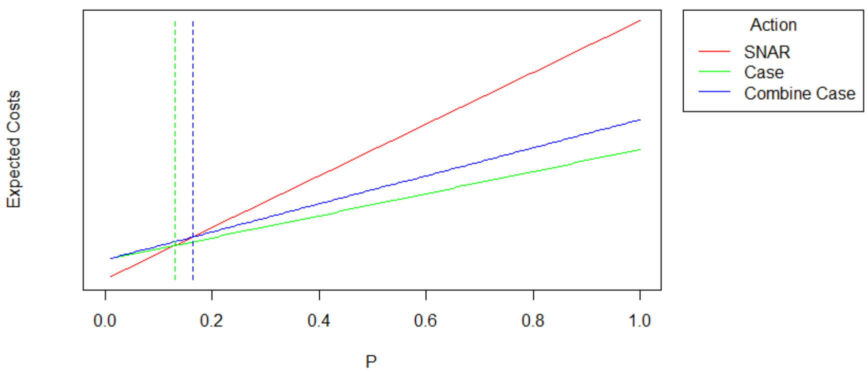


Figure 28: Influence of P on expected costs in Flat Detector case

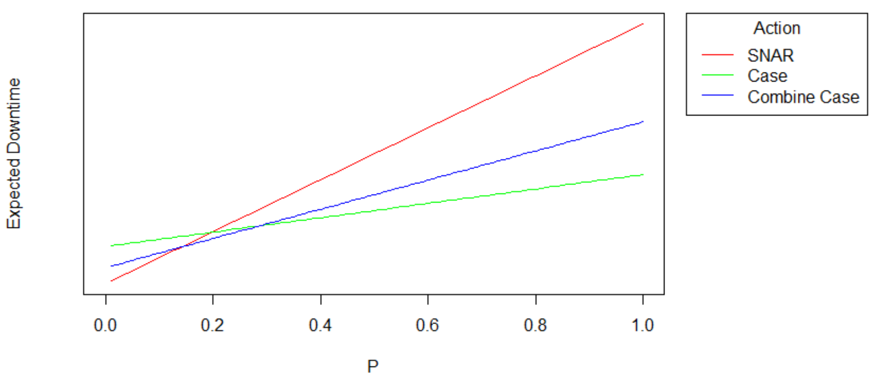


Figure 29: Influence of P on expected downtime in Flat Detector case

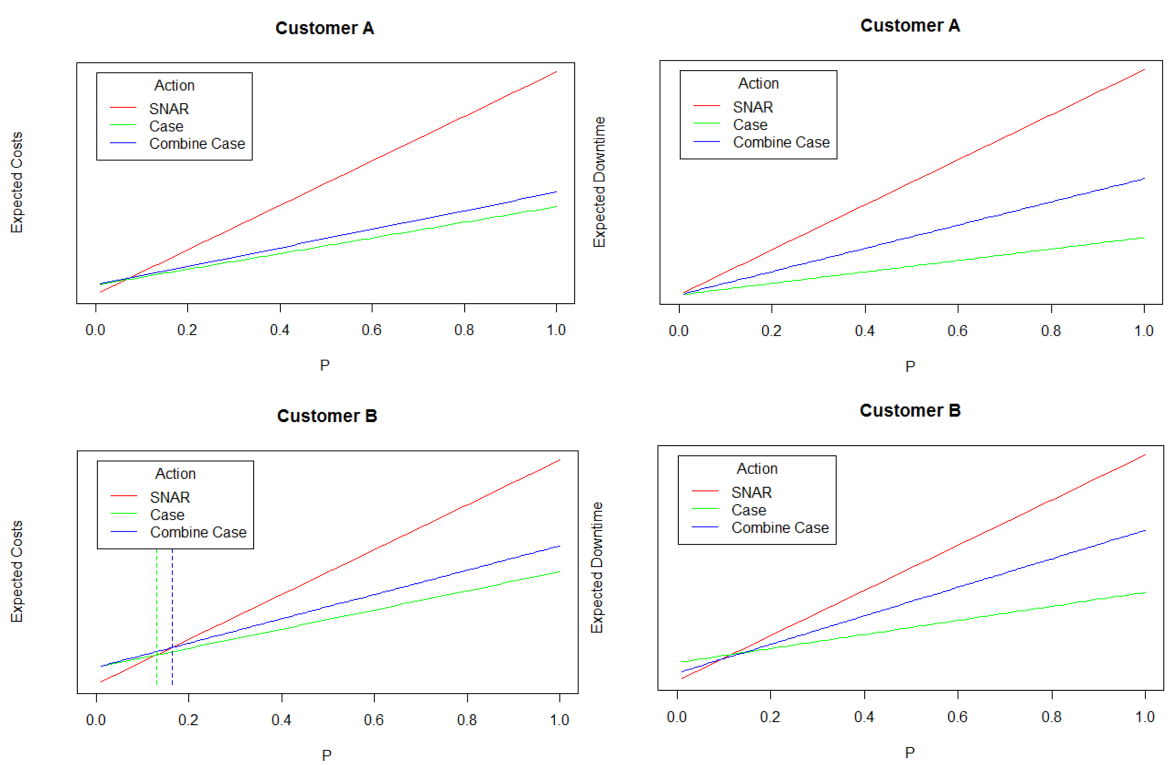
7.3 Influence of the service contract

Chapter 3 discusses the different service contracts that customers can sign for their equipment. This section evaluates the influence of the service contracts on the optimal decisions according to the model. We create three fictitious customers, which we refer to as Customer A, Customer B and Customer C. Customer A has a RightFit Uptime contract with the most extensive entitlements. Customer B has a RightFit Select contract with no coverage options. Customer C has the most basic service contract, RightFit Assists. The customer-specific parameters are shown in Table 13. Data from Appendix I is used to set these input parameters. The input values for the decision support model can be found in Appendix X.

Table 13: Customer specific parameters

Parameter	Notation
Customer contract	$v \in \{0, 1, \ldots, 6\}$
Outside working hours coverage	owh
Downtime compensation	$cc_{dt} \in \{0, 1\}$
SLA Response time for contract V	Tr_v

We vary P to find out if different probability thresholds exist for different types for customer. Note that we cannot use the newsvendor threshold for customer A. The newsvendor solution does not take into account the downtime compensation. Figure 30 shows the influence of P on the expected costs and downtime for the different customers.



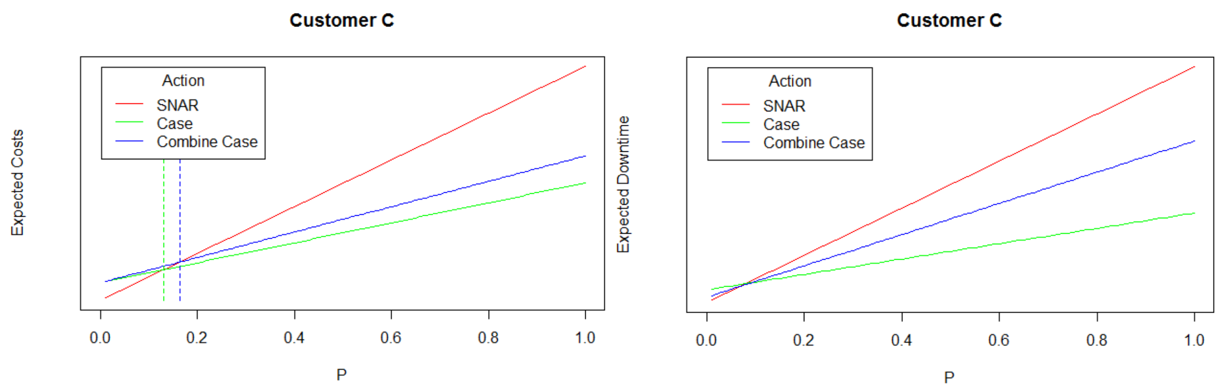


Figure 30: Influence of P on expected costs and downtime for different customers

We observe that the expected costs incurred for Customer A are the highest. This is due to the compensation of downtime he receives. The expected downtime of all actions is the lowest for Customer A. The reason behind this is that shorter on-site response times are offered to customers with higher contracts. After a customer call, the FSE is faster on-site to conduct maintenance on the failed equipment. In addition, the customer is entitled for maintenance outside operating hours. For Customer B and C, the expected costs of each action are equal. cc_{dt} is the only customer-specific parameters that influences the expected costs.

8 VALUE OF INFORMATION

"Information is not knowledge." – Albert Einstein

This chapter describes how the value of prognostic information generated by predictive models can be accessed. Valuating such information creates managerial insights in how much Philips can invest in the development or improvement of predictive models. In addition, the value of information provides show customers the benefits of RM. From the other hand, the value gives Philips information in how they can price the extra service to the customer. We can access the value prognostic information by comparing costs incurred by using this information with the costs incurred when this information is not used.

The value of information in terms of costs is noted by V_C. Then V_C is the difference between the costs incurred if prognostic information is available and if it is not available. If the RME decides to SNAR an alert, no information is send to the LSO. There is no difference between a SNARed alert and no alert at all because the LSO has no information in both situations. Therefore, we can say that the costs of having no information available, is equal to the expected costs of rejecting of an alert. If the expected costs of creating a case (either combined or separate), are higher than the expected costs of SNAR, the value of information is zero. We can define V_C as:

$$V_C = \max\big(E[C|a=0] - \min\big[E[C|a=1,y=0], E[C|a=1,y=1]\big], 0\big)$$

We use Model 1 from Chapter 6 and fix the values for a and y to calculate $E[C|a=0]$, $E[C|a=1,y=0]$, and $E[C|a=1,y=1]$. V_C can be calculated under different values of P allowing us access the value of information under different levels of imperfectness. Note that V_C is the value of an individual alert generated by the predictive model.

We can also access the value of information in terms of downtime noted by V_D. V_D is the difference in downtime if prognostic information is available and if it is not. We can define V_D as:

$$V_D = \max\big(E[D|a=0] - \min\big[E[D|a=1,y=0], E[D|a=1,y=1]\big], 0\big)$$

The value of downtime says how much downtime can be prevented by having prognostic information available. We can use Model 1 again to find the required values for this equation.

8.1 Value of information for different customers

In this section, we access the value of information for Customer A, B and C. We use the values for input parameters as given in Appendix IX and X. These values are valid for the flat detector. Figure 31 shows the value of information in terms of costs and Figure 32 shows the value of information in terms of downtime for the different customers.

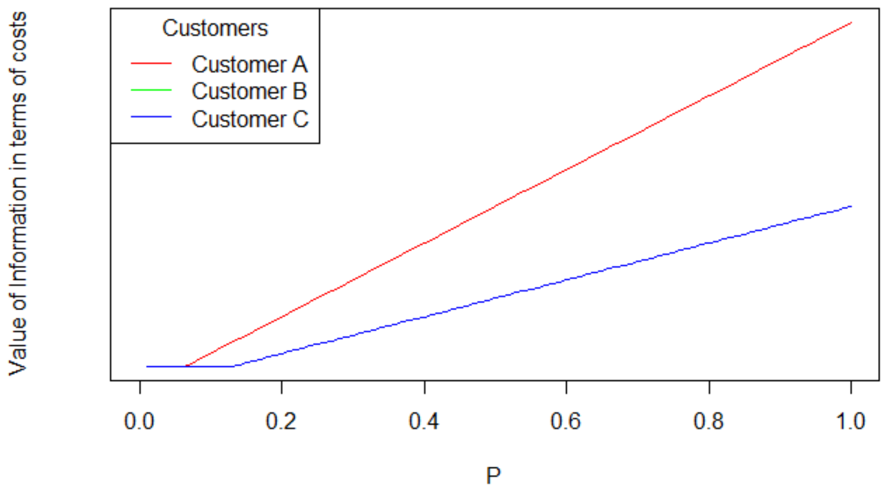


Figure 31: Value of information in terms of costs

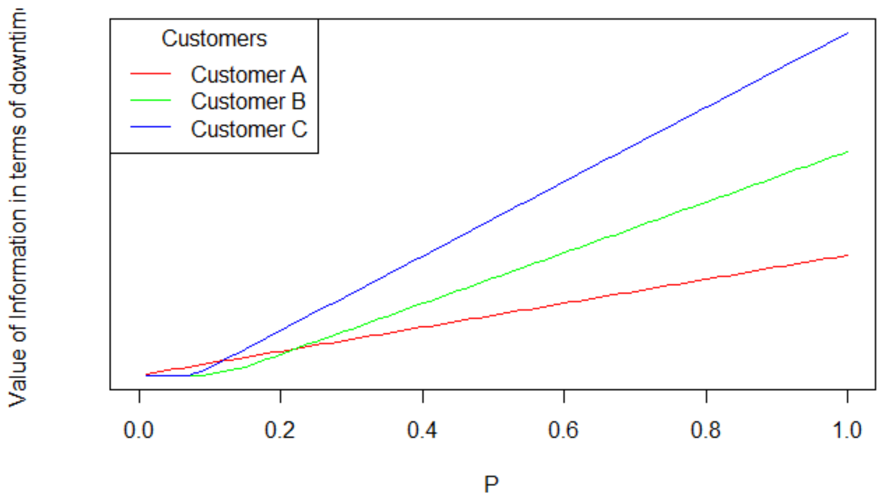


Figure 32: Value of information in terms of downtime

We can observe from both figures that predictions that provide more certainty on a future failure is more valuable both in terms of costs and in downtime. The value of information in terms of costs is higher for customer A than customers with lower service contracts. In terms of downtime, more accurate information is more valuable for customer with lower service contracts. For such customer, more downtime can be prevented by using prognostic information despite its imperfectness.

9 CONCLUSIONS AND RECOMMENDATIONS

"The more I learn, the more I realize I don't know." –Albert Einstein

During the research, we followed the regulative cycle as given by van Aken et al. (2007). We defined our research problem in Chapter 2. We analyzed and diagnosed the problem in Chapter 3 to Chapter 5. In Chapter 6, we developed a design for the research problem which we implemented by conducting a case study in Chapter 7. This chapter concludes the regulative cycle by answering the research questions and by providing recommendations for Philips.

9.1 RESEARCH QUESTIONS

In Chapter 2, we defined the research questions that guided us in this research. We motivated this research by the observation that proactive maintenance decision-making was done subjectively by RM. They receive no decision-support in judging the alert and how they should make customer-specific decisions on alerts generated by predictive models. To solve this business problem, we formulated the following main research question:

How can maintenance decision-making be optimized accounting for the imperfectness of information on machine conditions?

We defined several sub questions that all covers an important component in answering the research question. We focused the research on business driven decisions that does not require technical knowledge. Such knowledge is hard to capture in a model.

1. **How should the RMEs account for the service level agreements?**

We analyzes the service contracts used in Philips in Chapter 3. This gave us some understanding of the service requirements for customers with different contracts. We identified several entitlements and coverage options that should be taken into account in the decision-making by RMEs. The most important characteristics of service contract are the downtime compensation, the service window, and the uptime guarantee. The downtime compensation has influence on the expected costs of the decisions made by RMEs. The service window in the contract might provide customers services outside their operating hours, which leads to less expected downtime. The most valuable contracts include an uptime guarantee. If the customer has such contract, the decision-making should be aimed more towards downtime prevention.

2. How can RMEs account for imperfectness in predictions?

In Chapter 4, we aimed to answer this question. We first provided some information on the development of predictive models to create understanding on why models generate imperfect information. We found several factors that cause this imperfectness in the model. Important factors are the limitations of the service data that is used, different practices in different markets, and ambiguous error messages. Measurements for this imperfectness are the confusion matrix of the predictive model and the confidence level of the alerts. We were not able to develop a method to translate these measurements to a probability that a prediction is true. We only provide some information on how such method can be developed. In this thesis, we take the probability that a prediction is true as a given input. We eventually showed how optimal decisions could vary under different values of this probability.

3. What are the relevant cost factors in the maintenance decision-making by RMEs?

We identified the different cost factors relevant for maintenance decision-making in the first sections of Chapter 5. We compared these cost factors for proactive maintenance cases with reactive maintenance cases. Proactive maintenance cases are created by RM based on an alert. We referred to reactive maintenance cases as False Negative cases. We showed that in reactive maintenance, the costs associated with the case are on average more than twice as high. In addition, we analyzed the costs of cases that were initiated based on a false alert. We show that these costs are significantly lower. These costs are all relevant in the decision-making by RMEs. The analysis resulted in input values for parameters in the model of Chapter 6.

In the remaining part of Chapter 5, we analyzed other relevant characteristics of the current proactive maintenance policy. We modeled the remaining useful life and the time to on-site maintenance after an alert as random variables. We estimated the parameters and fitted a distribution such that we were able to model these times.

How can maintenance decision-making be optimized accounting for the imperfectness of information on machine conditions?

In order to answer this main research questions, we used input from the previous sub question. We created a mathematical model that evaluates different decisions that RMEs can make. It evaluates each action in terms of expected costs and expected downtimes. The model is implemented in R to visualize the evaluation of each action.

We found that there exist a probability threshold for alerts to create a case. For an alert with a higher probability that it is true, RM should create a case. We were able to rewrite the mathematical model to a Newsvendor problem. By using the standard newsvendor solution to this problem, we were able to find the optimal probability threshold for creating a case. We verified this optimal solution with a simulation. However, this Newsvendor solution only account for the different cost factors and does not take into account the expected downtimes. We can argue that for customers with a contract that does not include an uptime guarantee, the only objective is to minimize the expected costs. For such customers, RMEs can use the Newsvendor solution to make the optimal decisions for alerts.

9.2 FUTURE RESEARCH DIRECTIONS

This section provides some directions for future research. First, we discuss some directions specifically Philips specifically. These are motivated by limitations of this research and we discuss how to overcome these limitations.

9.2.1 Further research in Philips

This thesis developed a decision support model for RMEs but it faces some limitations. This section states those limitations and define further research direction to overcome these limitations.

Long-term optimization

Our model only supports the decisions made by RMEs how they should act on an alert such that the expected costs and/or downtime for that alert are minimized. These alerts cover one aspect of the imperfectness of predictive models. Another aspect of the imperfectness of these models is that they can miss failures. The decision support model created in this research does not account for such failures because those are handled directly by the LSO instead of RM. However, we showed that the impact of missing a failure could be very high in terms of costs and downtime.

In order to account for missing failures, the objective function of the mathematical model should include a time aspect. The current model is only focused on short-term optimization and excludes long-term effects of decisions. Both the arrivals of alerts and the arrivals of failure should be modelled such that the long run expected costs and/or downtime can be minimized. If the objective is to minimize the expected long-run costs, the objective function can look like:

Min $E[CR] = \frac{E[CC]}{E[CL]}$

$E[CR]$ represents the expected costs rate. $E[CC]$ the expected cycle costs and $E[CL]$ the expected cycle length. Renewal theory is required to define renewal events that end a renewal cycle. Renewal events are events that makes the system as-good-as-new. These events include proactive and reactive maintenance activities. By changing parameters related to the predictive model, the value of better predictive models can be accessed. In order to find a long-run probability threshold for P, more knowledge has to be gained on the distribution of alerts. The value of his threshold depends on how many alerts are generated for every value of P. It is possible to calculate long-term costs for each value of P, when this information is not available. Before we can find a distribution of alerts with different values of P, a method should be developed that determines the value of P for an alert.

Accessing the credibility of an alert

In this research, we took the credibility of an alert as an input for the model and noted it by P. We discussed what is required to access the value of P for an alert based on data. However, we were not able to develop a method that estimates the value of P based data on the alert. It would be valuable for Philips if a method is developed that estimates this value of P to provide direct and better support for the decisions for RM. We gave some suggestions in Section 4.2.3. Platt (1999) provides a method that translates SVM output to probabilistic output. The probabilistic output in that paper represents the probability that the predicted class is true. This probabilistic output would be the value P for an alert. If that method can be embedded in the predictive model, P can immediately be estimated by the predictive model.

Predicting spare part demand

Another interesting research area for Philips is the prediction of a spare part demand. The predictive models currently employed in Philips contain a prediction of a failure. However, in many cases that results from an alert, a spare part is not required. Other cases can require different part replacements to prevent a problem at the customer. The predictive models do not provide predictions on if or which part is required to prevent a failure. This results in lower First Visit Fix percentages for cases that require a part replacement (see section 5.2.2). In addition, we observed that there were unnecessary spare part shipments in almost 15% of the maintenance cases that required a part replacement.

More research needs to be conducted on how the demand for a certain spare part can be predicted for a predicted failure. The predictive model should be equipped with deeper diagnostic features to predict such demand. Whenever this demand can be predicted more accurately, the predictions can be used in the planning of spare part shipments. If Philips is able to predict a spare part replacement, it can be allocated closer to the customer before the maintenance is scheduled. This can possibly be done with cheaper and slower transport such that costs can be saved. This can also contribute to academic research because this area is explored very limited.

BIBLIOGRAPHY

Bloch, H. P., & Geitner, F. K. (1983). *Machinery failure analysis and troubleshooting.* Houston, TX: Gulf.

Bousdekis, A., Magoutas, B., Apostolou, D., & Mentzas, G. (2015). A proactive decision making framework for condition-based maintenance. *Industrial Management & Data Systems, 115*(7), 1225-1250.

Candea, G., Kawamoto, S., Fujiki, Y., Firedman, G., & Fox, A. (2004). Microreboot - a technique for cheap recovery. *Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, (pp. 31-44).

Djurdjanovic, D., Lee, J., & Ni, J. (2003). Watchdog Agent - an infotronics-based prognostics approach for product performance degradation assessment and prediction. *Advanced Engineering Informatics, 17*(3), 109-125.

Frazzon, E. M., Israel, E., Albrecht, A., Pereira, C. E., & Hellingrath, B. (2014). Spare parts supply chains' operational planning using technical condition information from intelligent maintenance systems. *Annual Reviews in Control, 38*(1), 147-154.

Gupta, A., & Lawsirirat, C. (2006). Strategically optimum maintenance of monitoring-enabled multi-component systems using continuous-time jump deterioration models. *Journal of Quality in Maintenance Engineering, 12*(3), 306-329.

Heng, A., Zhang, S., Tan, A. C., & Mathew, J. (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing, 23*(3), 724–739.

Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 1483-1510.

Lewandowski, M., & Oelker, S. (2014). Towards autonomous control in maintenance and spare part logistics - challenges and opportunities for preacting maintenance concepts. *Procedia Technology, 15*, 333-340.

Platt, J. C. (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Advances in large margin classifiers, 10*(3), 61-74.

Salfner, F., & Malek, M. (2007). Using hidden semi-markov models for effective online failure prediction. *Proceedings of the IEEE 26th International Symposium on Reliable Distributed Systems.*

Salfner, F., Lenk, M., & Malek, M. (2010). A Survey of Online Failure Prediction Methods. *ACM Computing Surveys, 42*, 1-42.

Sharma, A., Yadava, G. S., & Deshmukh, G. S. (2011). A literature review and future perspectives on maintenance optimization. *Journal of Quality in Maintenance Engineering, 17*(1), 5-25.

Siborska, J. Z., Hodkiewicz, M., & Ma, L. (2011). Prognostic modelling options for remaining useful life estimation. *Mechanical Systems and Signal Processing*, 1803–1836.

Sipos, R., Fradkin, D., Moerchen, F., & Wang, Z. (2014). Log-based Predictive Maintenance. *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1867-1876). ACM.

van Aken, J., Berends, H., & van der Bij, H. (2007). *Problem Solving in Organizations: A Methodological Handbook for Business Students.* Cambridge: Cambridge University Press.

van Rijsbergen, C. J. (1979). *Information Retrieval* (2nd ed.). Londen: Butterworth.

APPENDICES

APPENDIX I: RIGHTFIT PORTFOLIO DEFAULT AGREEMENTS

- Due to confidentiality reasons, this Appendix is not disclosed.

Appendix II: Interview Summaries

Interview 1: LSO

Within Philips, the Remote Monitoring program is enrolled for 1 to 2 years now. Initially, Remote Service engineers received alerts from the predictive models directly. However, it turned out that many of them where not valid. “There was a lot of rubbish”. There was a need for a filter between the predictive models and Remote/Field service engineers such that only valid alerts reach the engineers. That is why Remote Monitoring was introduced. They serve as the filter between the predictive models and the service engineers and they aim to reject invalid alerts. The RSEs and FSEs are in the Local Service Organizations (LSO). The LSO serve one or a few countries while RM serves globally.

A customer call can be received by a Customer Service (CS) employee or an RSE. If the customer that calls is not very technical, it is more difficult for the RSE to access the problem. The first suggestion made by the RSE to the customer is to reset the device. If that is not successful, he tries to diagnose the problem remotely. This is done by looking into the log files send by the system. He tries to access which component causes the problems. Ton estimates the success rate of the remote diagnosis with 70%. If the remote diagnosis is not successful, the RSE consults the T2 helpdesk. Engineers in the T2 helpdesk have more technical knowledge and thus a deeper understanding about the system. If this they are not able to diagnose the problem, the FSE has to go to the customer and conduct on-site diagnosis.

The planner or the RSE makes a case in OneEMS when the customer calls. In the Netherlands, remote service/diagnostic activities are labeled as TECU in the database system.

Some customers have in-house technicians that can execute some maintenance activities. If such technicians can conduct the required maintenance, the required parts are send to the customer. No RSE or FSE is dispatched to the customer in this case. Ton estimates that 10% of the problems can be solved by in-house technicians.

RSEs can also receive cases that are made by Remote Monitoring (RM). RM acts on alerts received from predictive models. They make a case when they consider the alert to be valid. This case is then send to the relevant LSO such that the RSE receives the case. A recommendation is written by the RME in the case that indicates how RSE should solve the problem. According to Ton, the RSEs in the Netherlands follow the recommendation always up.

Sometimes spare parts are required to solve the problem. If the problem is not completely clear, the RSE can decide to send multiple spare parts to ensure that the problem can be solved. Otherwise, it is possible that the wrong part is sent and the right part should be send with another shipment. The consequences of this scenario are more delays and more costs. That’s why RSE sometimes send more spare parts to the customer. If a part is not used, it should be returned to the warehouse. This is also a relevant costs factor. However, this decision is now done without using the information about possible costs. The decision is only based on the feeling and experience of the RSE. No costs are taken into account in the decisions.

Another factor that is not taken into account is the service level of the customer. RSEs have limited knowledge of the service contract of customers. In addition, the information on the service contracts is quite difficult to access. The RSE considers it desirable to have some support in the cost/service trade-off.

Interview 2: Predictive Model Development

There are different reasons that there is imperfectness in the predictions of the predictive models. These are due to uncertainties of different aspects related to the creation of the models by data scientists. Dimitrios mentioned the following uncertainties that affect the reliability of the models:

- The quality of the data
- Error messages are ambiguous
- Human decisions
- Sample size
- Different customers act differently. Some chose to 'live' with the problem. Dimitrios made the suggestion to investigate differences in behavior in the different markets by splitting the data based on market.

Model making

Predictive models try to classify data points into one of the two different classifications. The first classification is *close to failure.* The other classification is *working good.* The data scientists collect data of right before a failure and put this data in the bad data pool. Data of a period that is not close to a failure is collected and put into the good data pool. One problem with this method is that the failures that are used here, are only the failures where the component is replaced. For some failures, it is possible to do some calibration of the component that will fix the problem. These failures are not taken into account in making the model. It is even possible that this calibration is done in a period of which the data is put in the good pool. This can make the model less reliable.

In the development of predictive models, the goal is the maximize its precision while keeping the false positive rate under a certain limit. The limit depends on the model and system but often a limit of 1% is used. The confusion matrix is created by cross validation on historic data. They split the test data in a prior failure bin and a good bin. The assumption is made that error messages do not depend on the component or system.

The data set they use to construct the confusion matrix contains rows that represent one day of one machine. Most of the columns represent error messages but some are combinations of such messages. Each data field represents the number of times that an error message is received on that day on that machine. As mentioned, the data set is split into prior failure and good state. Data on the prior failure case contains information on how many days before the call open date is measured.

Based on this call open date, the data scientists can estimate how much time before a failure, a failure can be predicted.

The confidence of an alert is derived by calculating how far the prediction is from the hyperplane that classifies data points in one of the two categories. Support vector machines are used to construct this hyperplane. A prediction close to the hyperplane has a lower confidence than a prediction far from the hyperplane. The confidence of an alert is a number between 0 and 1. However, it is not fully understood what is says about the likelihood of a failure.

Interview 3: Remote Monitoring

When an alert arrives at RM, the RME checks for reasons to SNAR the alert immediately. These reasons include:

- The customer already called
- The engineer is already on-site
- The alert is a one-time occurrence

If the engineer is already on-site, the alert should be SNARed automatically by an AutoSNAR workflow program. However, this filter does not work 100% yet so the RME has to check it manually. Prediction models can be triggered by actions that are executed by FSEs. Now, only if the model is triggered during the visit of the FSE at the customer, the alert is AutoSNARed. However, it is better to AutoSNAR all alerts generated on a day that the FSE is on-site. Currently, those alerts are always SNARed.

The decision to make a case is now based on the experience of the RME. However, there is a need for a more structured way of making decisions. The decisions should take various parameters into account. According to Jan, these should include at least the customer contract and the culture. The culture is important because customers in different countries have different expectations of the service. For example, in Germany and Japan the customers want their equipment running with full performance while in countries like France or Spain, the customer can live with a decreased performance. A contract and country specific threshold for alerts is desired.

Some models generate to many alerts resulting in many false positives. An example is the hard disk model. Jan indicated that this model is too sensitive to minor changes in performance. Such alerts are considered as False Positives by the RMEs. Also collateral alerts can be considered as FP because they are often the result of a miss function of another component in the system. FP alerts can be identified by looking into the log files. The RMEs can identify if a problem exists and what the problem is. A thorough understanding of the log files is very important to make the right maintenance decisions.

Jan showed an example what can happen if a FSE without understanding of a log file makes these decisions. In that example, the FSE replaced the wrong parts in six subsequent visits. Jan showed that this could be prevented if someone with understanding of log files made the decisions. The RMEs have this understanding.

In Radar 2.0 (ISDA) RME wants the alerts together with the CAT patterns and the single lines in one single view.

Reasons that an alert are AutoSNARed are:

- Case still open
    - This happens when the RME chooses for the option 'SNAR until *date*' or 'SNAR until case status closed'

- Outdated alert
- Country not monitored
- FSE is on-site

When the RME chooses for the option ‘SNAR until *date*’ or ‘SNAR until case status closed’, alerts are also AutoSNARed.

At RM, they are now working on the connection between big data models and recommendations. Now it is still done manually but this will be automated in the future.

Differences in hours made on case

From data, it can be identified that for some cases, only remote work is done to close a case. On the question how this is possible, Jan answered that this can be due to several reasons. For example, it is possible that the problem is not actually fixed. The engineer can just think that on-site maintenance is not necessary. Another reason could be that the on-site maintenance actions are done in another case. This could be the case when the actions are executed during a planned maintenance case. It can also be that the hospital has in-house biomed engineers that can execute the on-site maintenance actions.

Valid Alerts

When a case based on an alert is executed by the FSE, the cases is rated by the RSE. He checks if the case and thus the alert was valid. This rating can be found in the data.

Jan estimates that 60-80% of the customer calls cannot be predicted with the log files. These are often problems like that the customer lost something or mechanical problems like a crack.

Finding FNs

It is very difficult to find what happens in case of a missed failure or a SNARed alert that was TP alert. In such cases, the customer calls directly to the LSO which creates a case often in the market’s language. RM is not notified if the customer calls with a problem that should have been predicted with a predictive model.

A solution could be to do a post mortem analysis. This means to not act on alerts and wait until the component fails. In this way, the time between an alert and the actual failure can be analyzed as well as the cost of ignoring/missing an alert.

Spare parts

The decision to order a part for a case is made by the FSE. He can take the part along during his first visit to the customer or he can order it when he is at the customer. Taking the spare part in the first visit is beneficial when the part is needed. However, there is some uncertainty in if the part is actually required to fix the problem. So the consequence of this action can be that the part is not used and returned to the warehouse. Unnecessary shipments costs are incurred in this case. Also, the FSE might just replace the part even if it is not necessary. In this case unnecessary shipment and part costs are incurred.

When the part is not taken along during the first visit, the engineer can find out that he actually need a part to fix the problem. He has to order the part during his first visit and has to return to the customer another date to replace the component. During this time, the customer can experience limited functionality or even a breakdown of the system. So there is a trade-off between the costs of bringing parts and the implied service to the customer. RM thinks that the spare part decision should be contract and market specific.

Interview 4: Data Science

Last time you mentioned that only part replacements were used as service data.

- **Why is this the case?**

In the model development, there has been experimented using richer service data. However, this made the model development way more complicated because the data was ambiguous. Sometimes prior a maintenance actions, some characteristics could be found in the log files indicating that something is wrong. But often nothing could have been found in the logfiles which make the feature (error message) selection in the model difficult. Using richer service data made the models more 'trigger happy' resulting in high FP rates. The model developers experienced that, at the moment, only using component replacements is working better.

- **Why are the recommended actions and the actual maintenance actions resulting from alerts not always part replacements then?**

More information on that can be asked to RM. Reasons are for example that FSEs were more likely to replace a component when the customer experienced problems in cases used in model development. It could have been that the current actions to fix the problem (as in the recommendations given by RM) were not executed before replacing a component. That's one reason why the actions on alerts are not always component replacements.

- **How many data points (component replacements) are used in the model development?**

This differs among the models. But generally about hundreds to thousands of replacements.

In our last meeting you mentioned that models are developed according to the following optimization problem:

$$Max\ Precision$$

s.t.

$$FP < 0.1$$

But this problem does not take into account the amount of failures that are not predicted by the model, the so called FN rate. In my opinion, such missing failures are most costly and have the most impact for the customer.

- **Why are these not taken into account?**

The FN rate is not taken into account because it is not understood very well. A big part of the FN cases in trainings data does not relate to initial goal to find the appropriate number of features used in the model.

- **How much is known about the costs and impact of missing a failure?**

Nothing is known about the costs and impact of missing a failure. This makes it difficult to use in the development of predictive models. If you want to account for missing failures as well in the predictive model development, the costs but also the type of error should be taken into account.

APPENDIX III: R CODE FOR FINDING ANSWER ON SAFETY QUESTION 3

```
Data1 <- read.csv("10000iXRcatmasterlist records 2016.csv", header = TRUE)
attach(Data1)
Data <- data.frame(Data1$CaseID, Data1$CustomerComplaint)
for(i in 1:length(Data[,1])) {
  AlarmYN <-  sub(".*the issue was discovered? *(.*?) *4. Was this an out of box failure.*", "\\1", Data[i,2])
  AlarmYN <- gsub("?\n* ","",AlarmYN, fixed = TRUE)
  AlarmYN <- gsub("\n*","",AlarmYN, fixed = TRUE)
  AlarmYN <- gsub("\n","",AlarmYN, fixed = TRUE)
  AlarmYN <- gsub("?","",AlarmYN, fixed = TRUE)
  Data[i,3] <- AlarmYN
  Data[i,4] <- 1
}
write.csv(Data, file = "AlarmYN all cases.csv", na="")
```

APPENDIX IV: SQL QUERIES

FN cases

```
Select distinct  a.SiteID, a.ModelUID, a.Confidence, a.AlertAction, a.SnarReason, a.AlertTime, c.CallOpenDate, DATEDIFF(day, a.AlertTime, c.CallOpenDate) as DiffDate, c.Notification, c.CaseID, c.ConfigId, c.CallType, c.TotalCMHours, c.TotalPMHours, c.TotalTravelHours, c.TotalRemoteHours, c.TotalOtherHours, c.TotalCMCosts, c.TotalPMCosts, c.TotalTravelCosts, c.TotalRemoteCosts, c.TotalOtherCosts, c.Type, c.CustomerComplaint
From Development.ISDA_model_output_alert as a, Development.iXR_fdvsv_catmasterlist as c
Where (a.SnarReason like '%Late %' OR a.SnarReason like '%Later %' OR a.SnarReason like '%late %')
AND a.SiteID =  c.ConfigId
and c.CallType = 'CM'
and c.CallOpenDate < a.AlertTime
and DATEDIFF(day, a.AlertTime, c.CallOpenDate) > '-30'
and DATEDIFF(day, a.AlertTime, c.CallOpenDate) < '-1'
```

and NOT EXISTS(select a.CaseId from Development.ISDA_model_output_alert as a where right(a.CaseId,9) = c.CaseID)

and c.TotalCMHours is not null

and c.CaseID is not null

FP Cases

Select distinct a.*, c.PcqlSortDescription, c.CaseNumber, c.Priority, c.ServiceType, cat.CallOpenDate, cat.TotalCMHours, cat.TotalPMHours, cat.TotalTravelHours, cat.TotalRemoteHours, cat.TotalOtherHours, cat.TotalCMCosts, cat.TotalPMCosts, cat.TotalTravelCosts, cat.TotalRemoteCosts, cat.TotalOtherCosts, cat.CustomerComplaint, cat.ExternalEngineerText, cat.InternalEngineerText

From Development.Teradata_oneems_case as c, Development.iXR_fdvsv_catmasterlist as cat, Development.ISDA_model_output_alert as a

Where right(c.CaseNumber,9) = cat.CaseID

and c.PcqlSortDescription = 'False alert'

and a.CaseId = c.CaseNumber

General RUL Estimation

Select distinct a.SiteID, a.AlertDescription, a.Confidence, a.AlertTime, c.CallType, DATEDIFF(second, a.AlertTime, c.CallOpenDate)/86400.0000 as DiffDate, c.CallOpenDate, c.CustomerComplaint

From Development.ISDA_model_output_alert as a, Development.iXR_fdvsv_catmasterlist as c

Where SnarReason like '%Region not monitored%'

and a.SiteID = c.ConfigId

and DATEDIFF(day, a.AlertTime, c.CallOpenDate) < '50'

and DATEDIFF(day, a.AlertTime, c.CallOpenDate) > '1'

AND c.CallOpenDate > a.AlertTime

AND c.CallType = 'CM'

AND c.CaseID is not null

General Time to OS maintenance

Select distinct a.AlertTime, c.CallOpenDate, c.OSWorkStart, DATEDIFF(second, a.AlertTime, c.OSWorkStart)/86400.0000 as DiffDate

From Development.ISDA_model_output_alert as a, Development.iXR_fdvsv_catmasterlist as c

Where right(a.CaseId,9) = c.CaseID

and c.OSWorkStart is not null

and DATEDIFF(second, a.AlertTime, c.OSWorkStart) > '0'

and a.CaseId like '010%'

FN Costs of the FDXD model

Select distinct a.SiteID, a.ModelUID, a.Confidence, a.AlertAction, a.SnarReason, a.AlertTime, c.CallOpenDate, DATEDIFF(day, a.AlertTime, c.CallOpenDate) as DiffDate, c.Notification, c.CaseID, c.ConfigId, c.CallType, c.TotalCMHours, c.TotalPMHours, c.TotalTravelHours, c.TotalRemoteHours, c.TotalOtherHours, c.TotalCMCosts, c.TotalPMCosts, c.TotalTravelCosts, c.TotalRemoteCosts, c.TotalOtherCosts, c.Type, c.CustomerComplaint

From Development.ISDA_model_output_alert as a, Development.iXR_fdvsv_catmasterlist as c

Where (a.SnarReason like '%Late %' OR a.SnarReason like '%Later %' OR a.SnarReason like '%late %')

AND a.SiteID = c.ConfigId

and c.CallType = 'CM'

and c.CallOpenDate < a.AlertTime

and DATEDIFF(day, a.AlertTime, c.CallOpenDate) > '-30'

and DATEDIFF(day, a.AlertTime, c.CallOpenDate) < '-1'

and NOT EXISTS(select a.CaseId from Development.ISDA_model_output_alert as a where right(a.CaseId,9) = c.CaseID)

and c.TotalCMHours is not null

and c.CaseID is not null

and a.ModelUID = 'IXR-PRED-FDXD'

RUL estimation of the FDXD model

Select distinct a.SiteID, a.AlertDescription, a.Confidence, a.AlertTime, c.CallType, DATEDIFF(second, a.AlertTime, c.CallOpenDate)/86400.0000 as DiffDate, c.CallOpenDate, c.CustomerComplaint

From Development.ISDA_model_output_alert as a, Development.iXR_fdvsv_catmasterlist as c

Where SnarReason like '%Region not monitored%'

and a.ModelUID = 'IXR-PRED-FDXD'

and a.SiteID = c.ConfigId

and DATEDIFF(day, a.AlertTime, c.CallOpenDate) < '50'

and DATEDIFF(day, a.AlertTime, c.CallOpenDate) > '1'

AND c.CallOpenDate > a.AlertTime

AND c.CallType = 'CM'

AND c.CaseID is not null

Time to OS maintenance for the FDXD model

Select distinct a.AlertTime, c.CallOpenDate, c.OSWorkStart, DATEDIFF(second, a.AlertTime, c.OSWorkStart)/86400.0000 as DiffDate

From Development.ISDA_model_output_alert as a, Development.iXR_fdvsv_catmasterlist as c

Where right(a.CaseId,9) = c.CaseID

and a.MODELUID = 'IXR-PRED-FDXD'

and c.OSWorkStart is not null

and DATEDIFF(second, a.AlertTime, c.OSWorkStart) > '0'

and a.CaseId like '010%'

APPENDIX V: FORMULAS FOR PARAMETER ESTIMATION

The following parameter estimators are used to test the empirical distribution of any random variable Y on different theoretical distributions.

Normal distribution

$$\hat{\mu} = M_1$$

$$\hat{\sigma}^2 = M_2 - M_1^2$$

Gamma distribution

$$\hat{\alpha} = \frac{M_1^2}{M_2 - M_1^2}$$

$$\hat{\beta} = \frac{M_1}{M_2 - M_1^2}$$

Exponential

$$\hat{\lambda} = \frac{1}{M_1}$$

Lognormal

$$M_1' = \frac{1}{n} M_1' \sum_{i=1}^{n} log(Y_i)$$

$$M_2' = \frac{1}{n} \sum_{i=1}^{n} log(Y_i)^2$$

$$\hat{\mu}_l = M_1'$$

$$\hat{\sigma}_l^2 = M_2' - {M_1'}^2$$

APPENDIX VI: GENERAL MOMENT ESTIMATIONS RANDOM VARIABLES

- Due to confidentiality reasons, this Appendix is not disclosed.

APPENDIX VII: MOMENT ESTIMATIONS CASE STUDY

- Due to confidentiality reasons, this Appendix is not disclosed.

APPENDIX VIII: DEFAULT VALUES FOR INPUT PARAMETERS

- Due to confidentiality reasons, this Appendix is not disclosed.

APPENDIX IX: DEFAULT VALUES FOR INPUT PARAMETERS IN CASE STUDY

- Due to confidentiality reasons, this Appendix is not disclosed.

APPENDIX X: INPUT VALUES FOR DIFFERENT CUSTOMERS

- Due to confidentiality reasons, this Appendix is not disclosed.

APPENDIX XI: IMPLEMENTATION OF DECISION SUPPORT MODEL IN R

```
#Failure prediction parameters
P <- 0.8

sigma <-CONFIDENTIAL
mu <- CONFIDENTIAL


#Time to On-site Maintenance parameters
alpha <- CONFIDENTIAL
beta <- CONFIDENTIAL

#Costs parameters
c_pdm1 <- CONFIDENTIAL
c_pdm2 <- CONFIDENTIAL
c_cm   <- CONFIDENTIAL
c_diagnostics <- CONFIDENTIAL
c_FP <- CONFIDENTIAL
c_d <- CONFIDENTIAL

#Customer parameters
cc_dt <- 0
tsm <- 30
Tr <- CONFIDENTIAL
owh <- 0

#Maintenance parameters
trepair <- CONFIDENTIAL
tdiagnostics <- CONFIDENTIAL

#Random Variables
g <- function(x) {dnorm(x, mu, sigma)}        #RUL
f <- function(Tos) {dgamma(Tos, alpha, beta)}   #Time to On-site Maintenance

#Probabilities
P_Tos_lss_tsm <- integrate(f,0, tsm)$value #Tos less than tsm
P_x_lss_tsm <- integrate(g, 0, tsm)$value  #X less than tsm

#Double integral X<Tos<tsm
g1<- function(x){integrate(function(Tos) dnorm(x, mu, sigma)*dgamma(Tos, alpha, beta), x, tsm)$value}
g2 <-function(x) {sapply(x,g1)}

#####Probability expressions for the different scenarios#####
#Case
C_P_Scenario1 <- integrate(g2,0,tsm)$value
C_P_Scenario2 <- P_Tos_lss_tsm-C_P_Scenario1
```

```
C_P_Scenario3 <- P_x_lss_tsm*(1-P_Tos_lss_tsm)
C_P_Scenario4 <- (1-P_x_lss_tsm)*(1-P_Tos_lss_tsm)

#Combine case
Comb_P_Scenario1 <- P_x_lss_tsm
Comb_P_Scenario2 <- 1-P_x_lss_tsm


#####Expected Costs and downtime calculations for each action#####
#SNAR
ED_SNAR <- P*(Tr+trepair+tdiagnostics)
EC_SNAR <- P*(c_cm +c_diagnostics) +c_d*cc_dt*ED_SNAR

#case not combine
ED_case <- P*(Tr+trepair)*(C_P_Scenario1+C_P_Scenario3)+(1-
owh)*trepair*(C_P_Scenario2+C_P_Scenario4)
EC_case <-
P*(c_cm*(C_P_Scenario1+C_P_Scenario3)+c_pdm1*C_P_Scenario2+c_pdm2*C_P_Scenario4)
+c_d*cc_dt*ED_case+(1-P)*c_FP

#case  combine
ED_comb_case <- P*(Tr+trepair)*Comb_P_Scenario1+(1-owh)*(trepair*Comb_P_Scenario2) #+(1-
P)*trepair
EC_comb_case <- P*(c_pdm2*Comb_P_Scenario2 +c_cm*Comb_P_Scenario1) +c_d*cc_dt*ED_case +(1-
P)*c_FP


#Value of information
ValueofInfoC <- max(EC_SNAR-min(EC_case,EC_comb_case), 0) #In terms of expected costs
ValueofInfoD <- max(ED_SNAR-min(ED_case,ED_comb_case), 0) #In terms of expected downtime

Max_C <- max(EC_SNAR,EC_comb_case,EC_case)
Max_D <- max(ED_SNAR,ED_comb_case,ED_case)
par(mar=c(5.1, 4.1, 4.1, 11), xpd=T)
plot(c(), c(), xlim = c(0, Max_C), ylim =c(0, Max_D), xlab="Expected Costs", ylab="Expected Downtime")
points(EC_SNAR, ED_SNAR, col = 'red',pch=19)
points(EC_case, ED_case, col = 'green',pch=19)
points(EC_comb_case, ED_comb_case, col = 'blue',pch=19)
segments(0,ED_SNAR,EC_SNAR,ED_SNAR, col='red', lty = 2,lwd = 1)
segments(EC_SNAR,0,EC_SNAR,ED_SNAR, col='red', lty = 2,lwd = 1)
segments(0,ED_case,EC_case,ED_case, col='green', lty = 2,lwd = 1)
segments(EC_case,0,EC_case,ED_case, col='green', lty = 2,lwd = 1)
segments(0,ED_comb_case,EC_comb_case,ED_comb_case, col='blue', lty = 2,lwd = 1)
segments(EC_comb_case,0,EC_comb_case,ED_comb_case, col='blue', lty = 2,lwd = 1)
legend("topright", inset=c(-0.4,0), c("SNAR","Case","Combine Case"), pch=19, col=c("red","green",
"blue"), title="Action")

ED_SNAR
```

EC_SNAR

ED_case
EC_case

ED_comb_case
EC_comb_case

ValueofInfoC
ValueofInfoD