MASTER

Condition based maintenance with event and usage data at Canon Production Printing

Hoogendoorn, D.D.

Award date:
2020

Department of Industrial Engineering & Innovation Sciences
Operations, Planning, Accounting and Control Group

Condition Based Maintenance with Event and Usage Data at Canon Production Printing

Master Thesis

D.D. Hoogendoorn

BSc Industrial Engineering – Avans University of Applied Sciences 2017

Student Identity Number – 1275429

Supervisors

dr. ir. R.J.I. Basten, TU/e, OPAC

ir. dr. S.D.P. Flapper, TU/e, OPAC

dr. Ö. Tüncel, TU/e, OPAC

MSc L.A. Teeuwsen, Canon Production Printing

April, 2020

TU/e School of Industrial Engineering

Series Master Theses Operation Management and Logistics

Subject headings: condition based maintenance, condition monitoring, maintenance policy selection, machine learning, Random Forest

It should be noted that all numbers and costs mentioned in this thesis are fictitious due to confidentiality reasons

Abstract

Condition Based Maintenance (CBM) is a maintenance program that determines when maintenance should be performed based on the condition of a part. The condition of the system is determined by monitoring continuous degradation signals. This research aims to determine how a CBM program can be created based on event data (error messages and warning messages) and usage data (usage counters and machine configuration parameters). This thesis uses a Random Forest machine learning model to find patterns in the event and usage data that can predict failures. Random oversampling is used to increase the performance of the Random Forest model. Three case studies have been conducted, and it is concluded that for these parts each failure has a unique pattern of event and usage data that cannot be used to predict other failures. The research did yield a Python program that can automatically create a CBM program and is easily adapted for different parts and even different machines.

Management Summary

This research presented in this thesis is conducted at Canon Production Printing (CPP) with the subject of Condition Based Maintenance (CBM).

Problem context

CPP has a wide range of black & white cutsheet printers which that are currently maintained via usage-based maintenance and corrective maintenance (CM). CPP aims to implement a CBM program for black & white cutsheet printers. This research focusses on the VP6000 as it is responsible for the majority of maintenance cost of the black & white cutsheet printers. The VP6000 does not continuously register sensor data. Instead it registers event data that is based on sensor data and usage data. Event data is defined as error and warnings messages, while usage data is defined as production counters and machine configuration parameters. The errors and warnings are based on sensor data or on system faults. The main research question therefore is:

"How can event and usage data be used to monitor the condition of the system and how can maintenance decisions be made based on this data?"

Research approach

This research uses two methods to answer the main research question:

- A selection procedure to time-efficiently find suitable candidates for this research.
- A procedure to create a CBM program. A CBM program consists of three steps: data acquisition, data processing and maintenance decision making (Jardine, Lin, & Banjevic, 2006)

To answer the main research question first a selection of parts is made that might be suitable for a CBM. The aim is to select three parts for this research. These parts are ideally selected based on the highest average yearly response time. However, the response time is not defined for the parts and has to be determined by manually examining visit logs. This is too time intensive for all parts. Therefore, first a selection of parts is made based on the maintenance funnel by Tiddens et al. (2018). The maintenance funnel first efficiently selects a number of parts that benefit from CBM based on two criteria. The criteria high part cost and low failure frequency as proposed by van Elderen (2016) are used. CPP values these criteria the most in selecting parts for CBM, however these criteria do not guarantee that CBM is a feasible program for the selected parts. The maintenance funnel is therefore used as it has two more steps that determine if CBM is feasible for a part. Tiddens et al. (2018) first propose several criteria to filter out more parts for which CBM is not feasible. Then, the potential cost savings are determined based on a sensitivity analysis to conclude if it is economically feasible to implement CBM for the part.

A CBM program consists of three steps. Firstly, the event and usage data that are relevant for predicting failures are collected. Secondly, the data is processed. There are many types of event and usage data per selected part. Any combination of these data types might be useful to predict failures. A machine learning model is therefore used as it can autonomously find patterns in for these large sets of data. Finally, maintenance decisions

are made based on the costs of CM visit and CBM. A method is proposed to determine how many CM and CBM visits there based on how often the machine learning model makes correct predictions, false predictions or fails to predict a failure. Based on the number of visits and the cost of each visit it is determined how much cost can be saved if the CBM program is implemented.

Results

The first step of this research resulted in three parts that are used. These parts are the: Preheat 1,2 unit, Preheat 3 unit and the Printhead Seneca. It is concluded that there are large differences in the potential cost savings for each part. This is due to selecting the parts based on part cost and failure frequency instead of response time and failure frequency.

For the selected parts it is concluded that a CBM program is not possible. It is concluded that each failure has its own unique pattern of event and usage data that cannot be used to predict other failures.

This research built a Python program that can automatically process event and usage data to create a machine learning model that can predict failures for any part. Furthermore, a method is proposed that can be used to make maintenance decisions based on the performance of the machine learning model.

Recommendations

This thesis makes several recommendations for CPP. The most important recommendations are the following:

Implementation plan

An implementation plan is recommended so that the Python program can be applied to other parts. This implementation plan can be applied to any part. First, the available event and usage data has to be collected per part. Then the Python program should be applied. The results can then be used for maintenance decision making.

Part selection

A CBM program is not possible for the selected parts of this research. It is therefore recommended to use a different method for part selection and use the Python program to create a CBM program. Firstly, it is recommended that a selection of parts is made that are expected to have the highest average yearly response times. The expert knowledge of the field service technicians and the service product specialists should be used to find these parts. Secondly, it should be determined for which parts CBM based on event and usage data is possible. The event and usage data are used by the field service technician to determine which parts might be the cause of the customer's issue and replace them. It is therefore expected that the field service technicians can give a good estimation of which parts can be used for CBM with event and usage data.

Additional recommendations

- It is recommended that CPP investigates why the event and usage data the patterns of event and usage data cannot be used to predict failures. The triggers that cause the errors and warnings might be wrongly defined.
- For optimal part selection the response time is needed per part. It is therefore recommended that CPP starts to collect the response time of parts.
- The data is collected in seven-day intervals, however it is expected that the most significant errors occur closely to the failure. It is therefore recommended to collect the data continuously.

Preface

This report is the result of my master thesis project conducted at Canon Production Printing in collaboration with Eindhoven University of Technology (TU/e). It would not have been possible to complete this thesis without the support other people.

First of all, I would like to thank Özge Tüncel who guided me during my project and provided me with feedback on my work. Secondly, I would like to thank Rob Basten, my first supervisor, your feedback has greatly improved the quality of my report. Furthermore, I also really appreciate the time and effort you spend on me. Thirdly, I want to thank my second supervisor, Simme Douwe Flapper, for your extensive feedback and critical questions. I really appreciate your flexibility and helping me make my report consistent.

I also want to express my gratitude to all the employees at Canon Production Printing. Especially, my company supervisor: Lonneke Teeuwsen. Thank you for all the time you spend in me and for all the discussions we had. I also want to express my sincerest thanks to Mark ten Have for providing me with helpful tips. Finally, I want to thank all the employees at CPP for making time for me and always being open to questions and discussions.

Finally, I want to thank everyone who supported me during my student life. First of all, I would like to thank my parents for their continued support. Then, I would like to thank all my friends at the TU/e with whom I spend many days in MetaForum and many nights in the Villa. Finally, I want to thank all my friends from my hometown. Once more to everyone, thank you! It has been a great journey!

Table of contents

List of tables

List of figures

List of definitions

Term	Definition
Age Based Maintenance	A form of preventive maintenance in which a part is replaced after it has reached a predefined age.
Buffer window	A window between the prediction point and prediction window that allows CPP time to replace the part before it is "likely to fail" within the prediction window.
Classification Problem	A machine learning task in which the model tries to predict a class label.
Click	A standardized form that is used to measure production. A click represents one single-side A4 sized print.
Clustering	A method to divide the population into a number of clusters. The population within a cluster is similar to each other while the population of two clusters are dissimilar.
Condition Based Maintenance	A maintenance policy that performs maintenance before a part fails based on the current condition of a component.
Consumable	Parts that are consumed during printing, e.g. ink.
Corrective Maintenance	A maintenance policy that performs maintenance after a component fails.
Count of usage Maintenance	A form of usage based maintenance in which a part is replaced after it has reached predefined level of production. At CPP this level is defined in clicks.
Counter	Counts the number of clicks a certain system has printed.
Criticality Classification	The first filter in the maintenance funnel by Tiddens et al. (2018) used to reduce the number of parts to the candidates that would benefit the most from CBM.
Data Processing Parameters	The parameters of the data processing method proposed by Wang et al (2017).
Diagnostic Condition Based Maintenance	A form of condition based maintenance that performs maintenance when faults are detected in the system.
Downtime	The time between a failure and the moment the printer is functioning again. Between these moments the printer is not operational.
Error	An error is a message registered by the machine when an unexpected condition happens. The printing process is stopped and the error is visible for the customer and registered for analysis.
Event Data	A dataset that contains all errors, warnings and replacement dates of components.
Expendable	A part CPP wants to operate to failure.
F1-score	Harmonic mean of the precision and recall scores. It is used to evaluate the precision and recall scores.
Failure	A part breakdown. This always results in a replacement.
Failure frequency	The average number of failures per printer per year for a part.
False Negative	A failure that the ML model did not predict.
False Positive	An incorrect prediction made by the ML model, i.e. the ML model predicted a failure when none occurred.

Feature	Features describe the instance for the ML model.
Field Service Technician	The term used to describe the service technicians that maintain the printers.
Focused Feasibility Study	The last filter in the maintenance funnel. Determines if it is technically possible and economically feasible to implement CBM.
High Dimensional Data	The dataset contains ten or more features (Han, Kamber, & Pei, 2012).
Instance	A learning moment for the ML model. An instance is considered positive if it has class label "1", it is negative if it has class label "0".
Jaccard Index	Statistic that describes the similarity between two datasets.
Labor costs	The hourly wage of the Field Service Technician.
Machine Learning	A computer program that learns autonomously and can make decision autonomously.
Maintenance Categorization Matrix	A version of the classification diagram specifically created for Océ. It uses the criteria: failure frequency and part costs.
Maintenance Funnel	A maintenance effectiveness analysis used to determine which parts benefit the most from a CBM policy and whether it is possible for these parts.
Modificative Maintenance	A maintenance policy that is performed to make a modification to the machine. This is done once per machine.
Observation Window	The window before each prediction point in which the event data is collected that is used to make the prediction.
Parameter	An adjustable value that is used to configure parts. An example is the light value of the Printhead, which determines how bright the light is.
Part	The level at which a field service technician makes a replacement on-site. Also, the level at which the replacements can be ordered by the FST.
Precision	Ratio of correct prediction by the ML model to all predictions made by the ML model.
Prediction Point	Point in time when the ML model will make a prediction.
Prediction Window	A window after the prediction point in which the ML model will indicate if a part is "Likely to fail" or "Not likely to fail".
Preventive Maintenance	The maintenance strategies that are performed before the part actually fails.
Prognostic Condition Based Maintenance	A type of condition based maintenance that predicts when the faults will occur in the system.
Recall	Percentage of the actual failures that the ML model is able predict.
Repair time	Time an FST is repairing a part.
Response time	The time a printer is down before the repair is started.
Showstopper	A criterion that if true, means that a part is not fit for CBM.

Showstopper Analysis	The second filter in the maintenance funnel. Used to determine if CBM is feasible for the selected part.
Stratified cross validation	A method to evaluate a ML model. It divides the data into groups and iterates over these groups to train and test the ML model. The stratification ensures there is an equal number of positive instances in each group. The instances can be either randomly assigned to each group or the dataset can be split into equal sized groups.
True Positive	Correct prediction made by the ML model.
Usage Based Maintenance	A maintenance policy that performs maintenance after a component has reached a predefined level of usage.
Visit Costs	The costs made to travel to the customer (includes the hourly wage of the service mechanic).
Visit Log	A log in which the FST describes what has been done during a maintenance visit
Warning	A warning is a message registered by the machine when an unexpected condition happens. The process is not stopped, and the warning is not visible for the customer. It is visible for the FST and CPP.

List of abbreviations

Abbreviation	Definition
CBM	Condition Based Maintenance
CM	Corrective Maintenance
CPP	Canon Production Printing
FN	False Negative
FP	False Positive
FST	Field Service Technician
ML	Machine Learning
OW	Observation Window
PW	Prediction Window
RF	Random Forest
SCV	Stratified Cross Validation
SMOTE+ENN	Synthetic Minority Over-Sampling Technique with Edited Nearest Neighbor under-sampling
SW	Sub-window
TN	True Negative
TP	True Positive
UBM	Usage Based Maintenance
VP6000	VarioPrint 6000 printer

List of variables

Variable	Definition
c_{ij}	The number of errors per type (i) in sub-window (j)
co_i	The value of counter type i at the end of the observation window
B	The cost savings of a CBM visit compared to a CM visit
C_{cbm}	Cost of CBM visit
C_{cm}	Cost of CM visit
C_{csq}	Consequence of failure cost per failure
C_{fst}	Cost of FST per hour
C_{loss}	Cost due to loss of income per hour
C_{part}	Cost of part
C_{pen}	Penalty cost per visit
C_{S}	Setup costs per visit
$C_{totalCBM}$	The total maintenance cost based on a CBM policy with machine learning
$C_{totalCM}$	The total maintenance cost based on a CM policy
M	The size of the sub-window in the observation window in clicks
N	Number of clicks prediction point is moved forward to create instances
OW	Observation window size (X*M)
p_i	1 if pattern i is found in the OW otherwise 0
rep	The number of failures
T	The number of error and warning types
T_{repair}	Visit repair time
T_{response}	Visit response times
TN	The number of true negatives
TP	The number of true positives
w_i	Value of parameter type i at the end of the OW
X	The number of sub-windows in the observation window
Y	The size of the prediction window in clicks
Z	The size of the buffer window in clicks

1. Introduction

This research is the result of a graduation project conducted at Canon Production Printing (CPP) B.V. in order to be awarded a master's degree in Operations Management & Logistics at Eindhoven University of Technology. CPP is considered a world-wide leader in printing. CPP is an innovative company focusing on: "accelerating digital imaging technologies and developing high-tech printing products and services" (Canon Production Printing, 2019). The main subject of this research is condition based maintenance (CBM). This research is part of the pro-active service logistics for capital goods (ProSeLoNext) project.

The remainder of this chapter is structured as follows. First, the research background is described in Section 1.1. This is followed by the research design in Section 1.2. Finally, Section 1.3 describes the outline of the remainder of this thesis.

1.1 Research background

In this section the research background is described. The section deals with three topics: the company background, an introduction to maintenance and the problem context.

1.1.1 Company background

Canon Production Printing is formerly known as Océ. Océ was acquired by Canon in 2010 and in January 2020 the name was officially changed in Canon Production Printing CPP). CPP's headquarters are in Venlo, the Netherlands. The company currently employs around 3000 people worldwide.

CPP currently has three product categories, namely continuous feed, cut sheet and large format printers. Continuous feed printers are continuously fed via a roll of paper. Cut sheet printers are fed sheets of paper from a stack. Large format printers can handle large size paper inputs. These printers can be continuously fed or sheet fed. All three product categories have printers that can only print in black and white, or color.

All these product categories have a service & support department that is responsible for the installation and maintenance of all the printers. This research focusses on the black & white group which is responsible for the service and support of the black & white cut sheet printers.

This research is part of the pro-active service logistics for capital goods project. In this project a consortium of three universities and seven companies work together on several topics in after sales services. The three main topics are predictive maintenance and service logistics, service business models, and service control towers. This research falls within the predictive maintenance topic.

1.1.2 Introduction to maintenance

In general maintenance can be divided into three categories, namely modificative maintenance, preventive maintenance and corrective maintenance (Arts, 2017). Figure 1 shows the different kinds of maintenance.

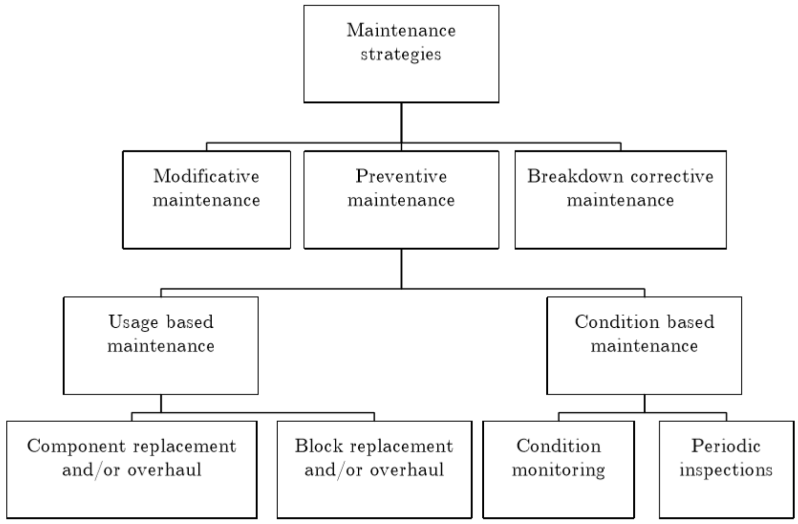


Figure 1 - Maintenance strategies (Arts, 2017)

Corrective maintenance is aimed at repairing systems that have broken down. Preventive maintenance in comparison, is aimed at performing maintenance before the system breaks down. As can be seen in Figure 1 preventive maintenance is divided into two categories, namely usage based maintenance (UBM) and condition based maintenance (CBM). UBM performs maintenance actions after a certain threshold of usage has been reached; an example of UBM is replacing the brake discs on a car after driving 100,000 kilometers. CBM measures the current condition of a part and performs maintenance when the current condition passes a threshold level. An example of CBM is replacing a break disc of a car after the thickness of the disc has been reduced to 80% of its original size. The measurements can be either continuous or periodical.

Jardine et al. (2006) propose three steps to create a CBM program (Figure 2). Firstly, the data that might represent the health of the system is collected. In the second step the data is first cleaned and then processes. The data is processed so that the deterioration of the system can be analyzed. Finally, maintenance decisions rules are suggested to create an efficient CBM program. The authors differentiate between two types of maintenance decision making, namely diagnostic and prognostic. The first focusses on detecting faults in the system while the latter focusses on predicting when faults will occur in the system.

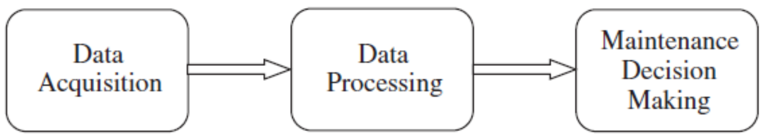


Figure 2 - Three steps of a CBM program (Jardine, Lin, & Banjevic, 2006)

1.1.2 Problem context

This research focusses on the service and support department of the B&W group at CPP. CPP aims to reduce the service costs and improve the uptime for the printers within this group. To achieve this goal, CPP has defined a proactive maintenance concept (Figure 3).

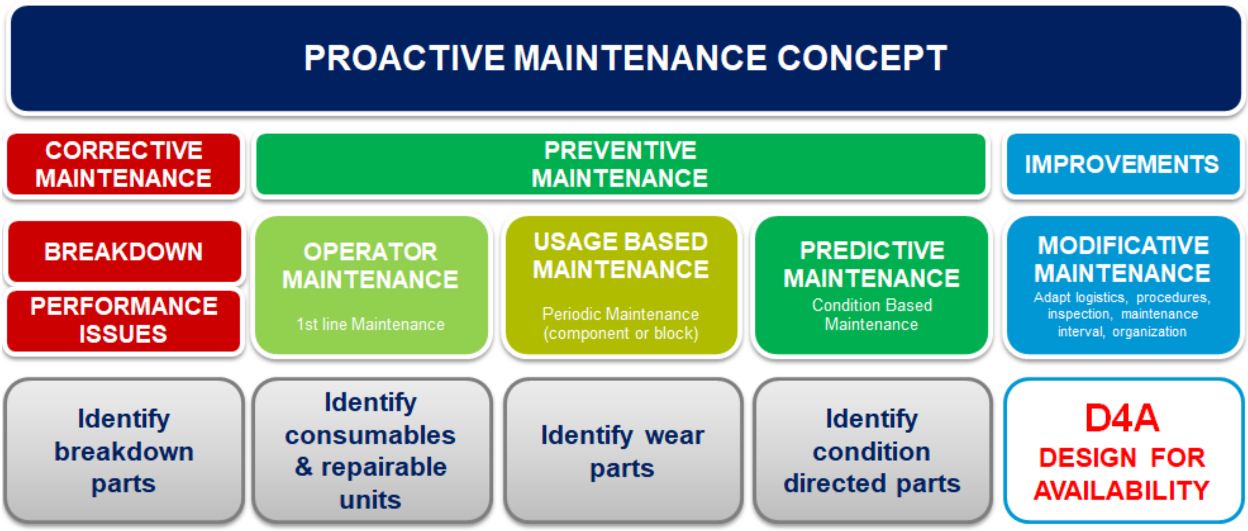


Figure 3 – Proactive maintenance concept (ten Have, 2019)

As can be seen from Figure 3, CPP has defined three pillars within the proactive maintenance concepts, namely corrective maintenance (red), preventive maintenance (green) and improvements (blue). The final goal is to identify for every printer within the B&W group, which parts should receive corrective maintenance, operator maintenance, usage based maintenance, predictive maintenance or modificative maintenance. Currently, none of the printers have parts that are maintained based on their condition.

CPP has defined scheduled service visits as uptime. Implementing CBM for a part might therefore result in higher uptime as these visits are scheduled with the customer. Furthermore, once a customer experiences a part failure the field service technician (FST) has to get to the customer's location and diagnose the problem. During this time the printer is down. A CBM policy replaces the part before it fails and the FST knows which part to replace beforehand. As a result, the downtime is reduced and the uptime increases.

Additionally, implementing CBM for parts might reduce maintenance costs as CPP loses income when the customer is unable to print as they do not use consumables during that time.

CBM will therefore contribute to the goals of increasing uptime and reducing maintenance costs. However, it is currently unknown how to determine the condition of a part based on the available data. The printers register errors, warnings, usage counters and parameter values. CPP aims to implement CBM for parts based on this information.

1.1.3 Problem statement

CPP aims to implement CBM for the black & white cut sheet printers to improve uptime and reduce maintenance costs. Within the B&W group, the VP6000 is responsible for 70% of the total maintenance costs. It is therefore expected that reducing the maintenance costs of the VP6000 will lead to the largest overall savings for the B&W group. Furthermore, the same is expected for the uptime.

The printers are maintained by CPP at customer locations. A maintenance visit needs to be planned with the customer and the FST requires time to get to the customer location.

It is therefore necessary to know that a part is going to fail in advance. A prognostic CBM policy is better than a diagnostic CBM policy.

These printers only register errors, warnings, usage counters and parameter values. Errors and warnings are defined as event data, while usage counters and parameter values are defined as usage data.

Based on the above information the following problem statement is defined:

"Currently it is unknown how event and usage data can be used to create CBM strategies for parts of the VP6000"

1.2 Research design

The research design is described in this section. First, the research questions are described in Section 1.2.1. Then the scope of the research is described in Section 1.2.2. The deliverables for CPP are described in Section 1.2.3 and the academic deliverables are described in Section 1.2.4.

1.2.1 Research questions

Considering the problem statement the following main research question is formulated:

"How can event and usage data be used to monitor the condition of the system and how can maintenance decisions be made based on this data?"

To answer the main research question, five research questions are defined. The first is aimed at finding a method to select suitable parts for this research. The second applies this method to reduce the number of parts to a selection that can be used for this research. The remaining three research questions are based on the steps to create a CBM program as proposed by Jardine et al. (2006). This section describes the general approach per research question. The detailed descriptions are provided in each chapter.

1. How to determine which parts are suitable candidates for this research?

First, it is determined how the parts should be selected for this research. For this the situation at CPP is discussed and the literature is reviewed for a best approach.

2. Which parts of the VP6000 are suitable candidates for this research?

A part is considered suitable for the research if it financially benefits from a CBM policy and if a CBM policy is a feasible solution. First a selection of parts is made based on high part costs and low failure frequency. Then a showstopper analysis is conducted to filter out parts for which CBM is not feasible. Finally, an economic feasibility study is conducted for the selected parts to see if a CBM policy would result in cost savings.

The remainder of the research questions are based on the steps to create a CBM program as suggested by Jardine et al. (2006).

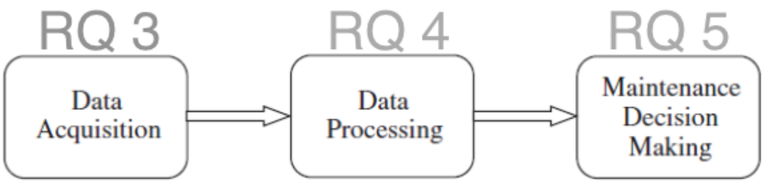


Figure 4 - Three steps of a CBM program (Jardine, Lin, & Banjevic, 2006)

3. What event and usage data should be monitored to determine the condition of the selected parts?

The second research question is based on the data acquisition step. To answer this research question all the available data is collected per part.

4. How can the selected event and usage data per part be processed so that it can be used for maintenance decision making?

This research question is based on the data processing step. The first step in this research question will be to clean the available data. This is followed by using literature to find a method to process the event and usage data so that it can be used to create a CBM program. A machine learning model is used as it can handle datasets with many different errors, warnings, usage counters and parameter values.

5. How can maintenance decisions be made based on the processed data?

The fourth research question is based on the final step in the methodology as proposed by Jardine et al (2006). In these case studies the optimal data processing parameters are determined per selected part to create a CBM program that yields the largest cost reduction compared to only a CM.

1.2.2 Scope

This section describes the scope of this research.

- This research focusses only on the VP6000 as it is currently responsible for the majority (70%) of the maintenance costs of the B&W cut sheet printers. Improvements to the maintenance concept of this printer might therefore lead to the largest overall cost reductions.
- Due to the time constraints of this research three parts are selected for RQ2, 3 and 4. This decision has been made in consultation with the service product manager.
- Expendables are excluded from the research. Expendables are parts CPP operates to failure.
- This research is focused on predicting failures, as there is only data available about when a failure occurs.

1.2.3 Deliverables for Canon Production Printing

This research will have the following deliverables:

- An internal report. This is the report for CPP which will contain the answers to the research questions and meet the criteria from the TU/e. This report will contain sensitive information.

- An external report. This report will not contain sensitive information.
- A Python program that will process the data and train a machine learning model to predict the replacement of any part of the VP6000.
- A guide on how to optimize the machine learning model so that the largest cost reduction can be achieved.

1.2.4 Academic Deliverables

- A case study in which the maintenance funnel for selection of suitable candidates for CBM is applied.
- A case study in which the steps to create a CBM program by Jardine et al. (2006) is applied.
- A case study in which the methodology for processing event data proposed by Wang et al. (2017) is applied.
- An addition of usage data to the methodology for processing event data by Wang et al. (2017).
- An addition of the methodology by Wang et al. (2017) for imbalanced data.
- An addition to the methodology by Wang et al. (2017) to optimize the method parameters based on potential cost savings for maintenance decision making.

1.3 Thesis Outline

The remainder of this thesis is outlined as follows. Chapter 2 answers Research Question 1 by selecting a methodology for parts selection. In Chapter 3 Research Question 2 is answered by applying this methodology to the parts at CPP. In Chapter 4 Research Question 3 is answered by collecting all event and usage data per selected part that might be used to create a CBM program. In Chapter 5 a methodology to process the data into a classification problem for machine learning is applied to answers Research Question 4. Chapter 6 answers Research Question 5 by optimizing the parameters of the data processing method to achieve the highest potential cost savings. Chapter 7 contains the conclusion and recommendations.

2. Parts selection methodology

The VP6000 is a complex machine consisting of a large number of parts. It is therefore necessary to find a selection of parts that are suitable for this research. This chapter will therefore answer Research Question 1:

> "How to determine which parts are suitable candidates for this research?"

In Section 2.1 the optimal maintenance policy selection criteria for CPP are described. Section 2.2 reviews the literature to find a method for parts selection for this research. Finally, in Section 2.3 a chapter conclusion is given.

2.1 Maintenance policy selection criteria

In an integrated maintenance approach, effectiveness analysis (which parts to improve?) should always be performed before efficiency analysis (how to improve these parts?) (Lee et al., 2014; Seecharan, Labib, & Jardine, 2018). CPP has defined the proactive maintenance concept to reduce maintenance costs and improve uptime. The effectiveness analysis should be conducted to select parts that are likely to contribute the most to these goals. After data cleaning there are 1031 parts that have been registered for the VP6000 (appendix A shows the data cleaning process). It is expected that most of these are not suitable candidates, so an efficient method is needed to find the suitable candidates.

One of the major benefits of CBM over CM is the reduction of downtime. Each CBM visit is conducted before a part fails, as a result the printer is not down while it waits for the part to be repaired. Downtime is often used to select parts for a CBM policy (Labib, 1998; Scarf, 2007; Seecharan, Labib, & Jardine, 2018; Tiddens, Braaksma, & Tinga, 2018). The time the printer is down while waiting for repair is defined as the response time. CPP generates revenue when the printer is operational as the customer uses consumables sold by CPP, e.g. ink. When the printer is down CPP therefore loses income. Reducing the response time therefore contributes to both goals of CPP, as the uptime is increased, and the loss of income is reduced. Ideally, the parts for this research would be selected based on the highest average yearly response time, i.e. response time multiplied by the average number of failures per year. However, CPP has not defined the response time per part.

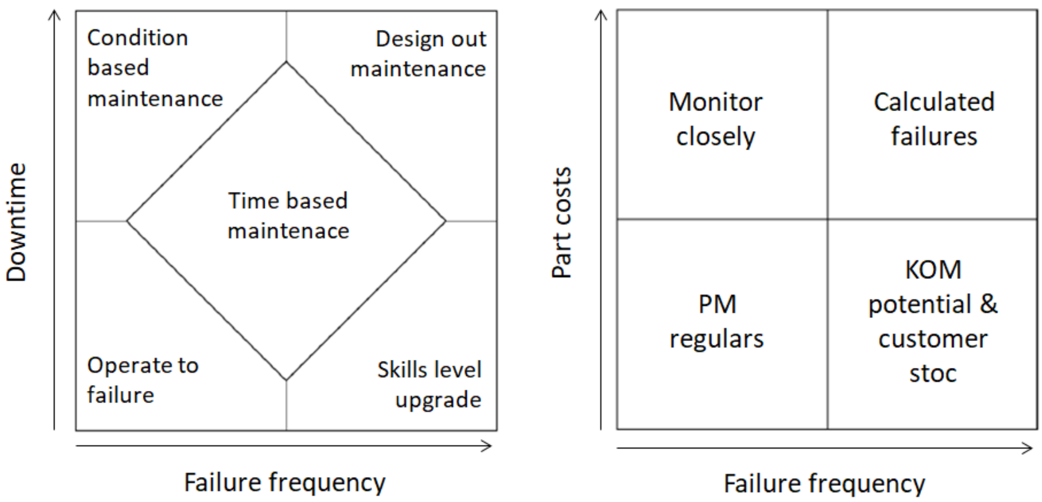
For this research, determining the response time has to be done by manually examining visit logs. As the reason for the response time of a service visit is noted in the visit logs by the FST. The service product specialist of the VP6000 has to estimate the response time of a part by finding several visits for which it is certain that the response time is caused by that part. This is too time consuming for all parts. The next section therefore reviews the literature to find a different method for parts selection. The response times are later determined for the selection of parts that are used for this research.

2.2 Maintenance policy selection in literature

One method to select parts to improve is by determining the critical parts of the system. The criticality of parts is commonly determined based on a dependability analysis (Brahimi et al., 2016). In this analysis the criticality is determined based on their availability, reliability, maintainability, safety and integrity (Avizienis et al., 2004). An example of a dependability analysis is the failure mode, effects and criticality analysis (FMECA). Dependability analyses determine a score per part or per failure mode based on expert knowledge. This process is therefore considered too time consuming to select parts of the VP6000 for this research.

Another popular approach for maintenance policy selection is multi-criteria decision making (MCDM) (Ding & Kamaruddin, 2015). The benefit of this method is that it can include multiple objectives. It can also include feasibility objectives to determine if the maintenance policy is feasible for the selected parts. The most widely used MCDM method is the analytical hierarchy process. For MCDM it is necessary to first determine the criteria, their relative weights and then determine the scores per part. This method is therefore considered too time consuming for this research.

A more efficient method to determine the optimal maintenance policy per part is the classification diagram. In this diagram all the parts are plotted based on their failure frequency and downtime (Figure 5a). It can therefore be used for all 1031 parts of the VP6000. Parts with low failure frequency and high downtime are optimal candidates for CBM. Parts with high downtime and high failure frequency should be designed out of the machine. As stated in the previous section, downtime information (or response time) is not easily available at CPP. However, different factors can be used (Scarf, 2007; Tiddens, Braaksma, & Tinga, 2018). Van Elderen (2016) concludes based on interviews, that CPP finds failure frequency and part costs the most important factors for maintenance policy selection (Figure 5b). The author suggests the maintenance categorization matrix (Figure 5b) in which the best parts CBM are in the monitor closely quadrant. These parts have high part cost and low failure frequency. Van Elderen determined these criteria based on what the stakeholders of his research valued the most. The research by van Elderen and this research share several stakeholders. The benefit of the maintenance categorization matrix is therefore that the selected parts are important to the stakeholders. The downside of the classification diagram is that it has no criteria to determine if CBM is feasible for a part.



a) Downtime and failure frequency (Scarf, 2007) b) Part costs and failure frequency (van Elderen, 2016)

Figure 5 - Classification diagrams

Tiddens, Braaksma & Tinga (2018) add two more steps to the classification diagram to find parts for which CBM is feasible. Their approach is shown in Figure 6. The authors first use the classification diagram, with downtime and failure frequency, to find the most critical parts for CBM in the criticality classification. The authors then define eleven criteria, named showstoppers, to determine if CBM is a feasible policy for the remaining parts. If one or more of the eleven showstoppers are true then CBM is not feasible for the part. If after the showstopper analysis it is still uncertain whether CBM is feasible for a selected part, a focused feasibility study is conducted. It contains a technical and economic feasibility study to see if CBM is a feasible option. The technical feasibility study is similar to the steps to create a CBM program by Jardine et al. (2006). The economic feasibility determines if a CBM policy could result in a cost reduction. The benefit of the maintenance funnel is that it combines the efficiency of the classification diagram with a feasibility check for the parts.

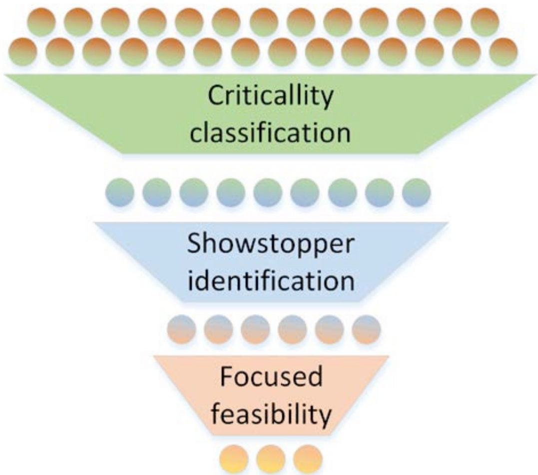


Figure 6 - Maintenance funnel (Tiddens, Braaksma, & Tinga, 2018)

In conclusion, the maintenance funnel proposed by Tiddens et al. (2018) combines the efficiency of the classification diagram with two more steps to determine if a CBM policy is feasible for the selected parts. The maintenance funnel is therefore used in this research. Furthermore, the maintenance categorization matrix as proposed by van Elderen (2016) is used in the criticality classification. It is included as it contains the criteria valued the most by CPP, namely part cost and failure frequency. It is not expected that by using these criteria the parts with the largest potential cost savings are selected, as for this the response time is needed (Section 2.1). However, the economic feasibility study filters out the parts for which a CBM policy is not likely to result in cost savings. This ensures that even though not the parts with the high potential cost savings are selected, at least all selected parts will potentially result in cost savings. The technical feasibility study is not included as these questions are similar to the steps to create a CBM program which are answered in RQ2, RQ3 and RQ4.

2.3 Conclusion

The ideal parts for this research should have be selected based on the highest average yearly response times. As reducing the response time results in cost savings and higher uptime. However, the response time has to be estimated per part by manually examining visit logs. As the reason for the response time is noted in the visit logs. This process is too time consuming for all parts.

As an alternative the maintenance funnel approach as proposed by Tiddens et al. (2018) is used. This is an efficient method for part selection with steps to ensure CBM is a feasible policy for the selected parts. The first step of the maintenance funnel is the criticality classification based on downtime and failure frequency. As downtime is not available, the maintenance categorization matrix by van Elderen (2016) is used. This matrix is designed for CPP and selects the parts CPP values the most for CBM. The parts with high part cost and low failure frequency are selected for CBM. It is not expected that by using these criteria the parts with the largest potential cost savings are selected, as for this the response time is needed. However, the addition of the economic feasibility study in the maintenance funnel ensures that only parts are selected that potentially result in cost savings. In this last step of the maintenance funnel, the response times are determined per selected part as they are needed for the economic feasibility study. Furthermore, it is expected that only a few parts are left for the economic feasibility study. As a result, it is no longer too time consuming in this step.

Chapter 3 Part Selection for the VP6000

In this chapter the maintenance funnel approach as proposed by Tiddens et al. (2018) is applied to answer Research Question 2:

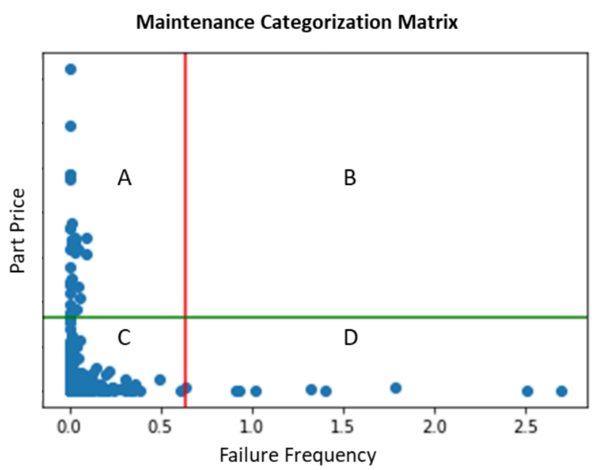
"Which parts of the VP6000 are suitable candidates for this research?"

Only the European event data is uploaded by the service mechanics and are accessible for CPP. For this reason, it has been decided that for the part usage data the focus will also be on Europe. The decision to use three years of data has been made after discussion with the service data analyst. Going farther back the data will contain failures of parts for which solutions have been implemented. A few modifications might be made for the parts in this period. The selected parts are therefore checked to see if these received modifications.

The first step of the maintenance funnel as proposed by Tiddens et al. (2018) is the criticality classification. This step is conducted in Section 3.1. The second step in the maintenance funnel is the showstopper analysis (Section 3.2). For the last step of the maintenance funnel an economic feasibility study is conducted (Section 3.3). Finally, a chapter conclusion is given in Section 3.4.

3.1 Criticality classification

The first step in the maintenance funnel by Tiddens et al. (2018) is the criticality filter. The authors apply the classification diagram to identify the most promising candidates for CBM. The authors determine these candidates based on downtime and failure frequency. Failures at CPP are defined as part breakdowns that result in replacements. As discussed in Section 2.1, retrieving the downtime for each part is too time intensive. Instead, the maintenance categorization matrix as proposed by van Elderen (2016) is used. This matrix is based on the criteria: part cost and failure frequency. Van Elderen (2016) determined that CPP finds these criteria the most important for maintenance policy selection. Figure 7 shows the parts of the VP6000 plotted in the maintenance categorization matrix.



A : Monitor Closely

B : Calculated Failures

C : PM Regulars

D :KOM Potential & Customer Stock

Figure 7 - Maintenance categorization matrix (van Elderen, 2016) for the VP6000

Each variant of the classification diagram determines the boundaries in a different manner. Lee et al. (2014) state that the boundaries should be based on the user's maintenance or production needs, while Scarf (2007) and van Elderen (2016) use the Pareto rule to determine the boundaries. Seecharan, Labib & Jardine (2018) state that if the parts are spread homogenously, i.e. there are no large discrepancies between the part locations, the boundaries can be placed at half of the highest value in each range. However, if there are large discrepancies in the range a clustering method might be a better approach to determine the boundaries. As can be seen from Figure 7, the parts are mostly centered in the bottom left corner of the matrix. The parts are therefore not spread homogenously in the matrix. Furthermore, there are large discrepancies between the values of the failure frequency. As can be seen from Figure 7, most failure frequencies are between 0 and 0.5. However, there are a some that are higher with two outliers around 2.5. The same is true for part price as there are a few outliers. A clustering method is therefore the best approach.

Clustering methods divide the population into several subsets or clusters. The population within a cluster is similar to each other, while the populations of different clusters are dissimilar (Han, Kamber, & Pei, 2012). The Jenks natural breaks optimization method as proposed by Jenks (1977) creates clusters with a similar in-cluster population by minimizing the in-cluster variance. This is the variance between the values in the cluster and the cluster mean. Variance is a measure of dissimilarity. It is therefore a good criterion for creating clusters. In addition, the Jenks natural breaks optimization maximizes the dissimilarity between clusters by maximizing the between-cluster variance. This is the variance of the values of one cluster with the mean of the other cluster.

Using the Jenks natural breaks optimization, the boundary for the failure frequency is determined to be 0.63449 and the boundary for the part price is determined to be €1104.90. The Jenks natural breaks optimization is implemented in Microsoft Excel by using the Real Statistics Resource Pack (Real Statistics, 2019).

All the parts that are in the monitor closely quadrant of the maintenance categorization matrix are in Appendix B. Table 1 shows the selection of parts in the monitor closely quadrant that constitute 79% of the total yearly part costs. The number of parts to consider further is reduces, as identifying the potential showstoppers is time consuming for the service product specialists.

Table 1 - Parts in the monitor closely quadrant

Part name	Failure Frequency	Part cost	Avg. yearly part costs	Cum. perc. total yearly part costs
Printhead Seneca	0.088	€ 2,292.67	€ 315,623.87	16%
Powerunit	0.095	€ 2,057.99	€ 303,210.66	32%
Preheat 1,2 unit	0.049	€ 2,136.80	€ 173,459.58	41%
Industrial Controller 1	0.032	€ 2,282.27	€ 155,039.07	49%
Preheat 3 unit	0.059	€ 1,380.08	€ 144,489.35	56%
Printhead Cicero	0.029	€ 2,307.57	€ 104,609.99	62%
Main Node	0.047	€ 1,562.49	€ 86,237.35	66%
Develop Unit 1	0.026	€ 2,187.38	€ 84,578.35	70%
Develop Unit 2	0.026	€ 2,072.62	€ 82,940.77	75%
User Interface	0.038	€ 1,216.10	€ 73,371.69	79%
		Total	**€ 1,523,560.68**	**100%**

3.2 Potential showstopper identification

The second filter in the maintenance funnel is the showstopper identification. The showstopper identification for the parts is shown in Table 2. The goal of the showstopper identification is to filter out parts for which CBM is not possible. Table 3 shows the potential showstoppers (PSs).

There are four categories of showstoppers, namely: clustering, technical feasibility, economic feasibility and organizational feasibility. If one of these showstoppers is rated "yes" for a part, then CBM is not a suitable option. If one or more showstoppers are rated "maybe" for a part, then a focused feasibility study is conducted. If all of the showstoppers are rated "no" then the part is immediately considered suitable for CBM. The showstoppers identification is conducted with the service product specialists of the VP6000.

Table 2 - Showstopper identification

Part name	c1	c2	t1	t2	e1	e2	o1	o2	o3	o4	o5	Result
Printhead Seneca	**M**	N	**M**	**M**	N	N	N	N	N	N	N	**M**
Powerunit	**M**	N	**Y**	**Y**	N	N	N	N	N	N	N	**Y**
Preheat 1,2 unit	**M**	N	**M**	**M**	N	N	N	N	N	N	N	**M**
Industrial Controller 1	**M**	N	**Y**	**Y**	N	N	N	N	N	N	N	**Y**
Preheat 3 unit	**M**	N	**M**	**M**	N	N	N	N	N	N	N	**M**
Printhead Cicero	**M**	N	**M**	**M**	N	**Y**	N	N	N	N	N	**Y**
Main Node	**M**	N	**Y**	**Y**	N	N	N	N	N	N	N	**Y**
Develop Unit 1	**M**	N	**Y**	**Y**	N	N	N	N	N	N	N	**Y**
Develop Unit 2	**M**	N	**Y**	**Y**	N	N	N	N	N	N	N	**Y**
User Interface	**M**	N	**Y**	**Y**	N	N	N	N	N	N	N	**Y**

Y : Yes

M : Maybe

N : No

Table 3 - Potential showstoppers based on (Tiddens, Braaksma, & Tinga, 2018)

Id	Potential Showstoppers (PS)
c	**Clustering**
c1	CBM activity does not fit in production planning of the customer
c2	CBM activity is part of a cluster of activities that are conducted together. The CBM activity does not influence the time at which this cluster of activities is conducted
t	**Technical feasibility**
t1	Degradation of the part cannot be detected with existing technology within the company
t2	Degradation of the part cannot be modeled with additional research
e	**Economic feasibility**
e1	Insufficient financial resources to cover possible investment costs in setting up a CBM system
e2	Part does not fail often enough for a positive business case
o	**Organizational feasibility**
o1	Maintenance personnel will not trust the CBM system
o2	Maintenance personnel is not willing to adopt CBM
o3	CBM does not fit with the operational mission
o4	CBM does not fit with the internal and external relations of the company
o5	CBM does not fit with the availability of spare parts

3.2.1 Clustering showstoppers

This section discusses the clustering showstoppers for the selected parts.

c1: CBM Activity does not fit in production planning of the customer

All the parts score “Maybe” on the PS-c1 (Table 2). Not all customers might be able to fit CBM activities into their production schedules. However, the assumption is made that most customers will be able to fit it into their production schedules and will prefer it to a CM visit.

c2: CBM activity is part of a cluster of activities that are conducted together

For the VP6000 there are no predetermined clusters of maintenance activities. The FST only goes to the customer on CM visits. During this visit the FST will only do CM activities and a few preventive maintenance activities. The preventive maintenance activities each have their own trigger and are not conducted in predetermined clusters. In conclusion, no parts are in predetermined clusters that would be impacted by implementing a CBM policy. The parts are therefore rated “No” on PS-c2 (Table 2).

3.2.2 Technical feasibility showstoppers

This section discusses the technical feasibility showstoppers for the selected parts.

t1: Failure of the part cannot be predicted with existing technology within the company

The Printhead Seneca has a sensor that measures the temperature and a light intensity setting. The errors and warnings from the sensor and the setting might be used to predict a failure. The Printhead Seneca therefore receives a “Maybe” on PS-t1. The Printhead Cicero is comparable to the Printhead Seneca in sensors, thus also receives a “Maybe” on PS-t1. The Preheat 1,2 unit has sensors that measure the temperature of the preheat plates and sensors that measure the motor speed of the belts that transport the paper through the Preheat 1,2 unit. The errors and warnings registered by these sensors might be used to predict a failure. The Preheat 1,2 unit therefore receives a “Maybe” on PS-t1. The Preheat 3 unit is comparable in sensors to Preheat 1,2 unit. It therefore also receives a “Maybe” on t1. These sensors might register errors and warnings that would be useful in predicting a failure. The Powerunit, Industrial Controller 1 and the Main Node are industrial computers. There are no sensors that measure degradation signals for these parts, and it is thus not possible to predict a failure. These three parts therefore receive a “Yes” on PS-t1, thus are filtered out of the research. The User Interface is a computer screen that is as the interface of the printer. There are no sensors related to the interface, thus this part receives a “Yes” and is filtered out. Develop Unit 1 and Develop Unit 2 have sensors that register when a motor or roller has stopped running. This might be an indicator of degradation, however in practice both the develop units are hardly replaced on error or warning data. They are replaced, because the customer has issues with the print quality and the FST does not know what causes it. As a result, many of the replacements were not actual failures. Therefore, both the develop units receive a “Yes” on PS-t1.

t2: Failure of the part cannot be predicted with additional research

For every part additional research is conducted to find out whether this is possible. Event data is used to predict failures in literature (Sato, Morimoto, & Takata, 2017) (Wang, Li, Han, Sarkar, & Zhou, 2017). It is therefore concluded that it might be possible to predict failures based on event data. However, additional research is needed to determine if this is possible for the VP6000. All the parts that measure degradation signals therefore receive a “maybe” on PS-t2.

3.2.3 Economic feasibility showstoppers

This section discusses the economic feasibility showstoppers for the selected parts

e1: Insufficient financial resources to cover possible investments

PS-e1 is “No” for all parts (Table 2) as there are few additional investments required. The CBM monitoring is done with existing sensors in the printer and there is already an environment for maintenance planning. The CBM activities can be implemented in this environment. This implementation has to be done by a software engineer and this will incur some costs. There are sufficient financial resources to cover these costs and therefore all the parts receive “No”.

e2: Part does not fail often enough for a positive business case

The second showstopper in the economic feasibility category is PS-e2. All the parts in the monitor closely quadrant are expected to result in a positive business case. To ensure this a lower bound has been implemented for the failure frequency in the maintenance categorization matrix (van Elderen, 2016). However, the Printhead Cicero is only used in the VP6000 Classic, the oldest model. CPP is actively trying to phase out the Classic by selling Titans to customers that currently own a Classic. The support for the Classic stops in 2023. It is therefore expected that the failures of the Printhead Cicero will reduce and the business case will not be positive. For this reason, the Printhead Cicero is rated "Yes" on PS-e2 (Table 2).

3.2.4 Organizational feasibility showstoppers

This section discusses the organizational feasibility showstoppers for the selected parts.

o1: Maintenance personnel will not trust the CBM system

CBM has already been successfully implemented for another product at CPP, the VPi300. The maintenance personnel are therefore familiar with the system and they have trust in the system. All the parts are thus rated "No" on PS-o1 (Table 2).

o2: Maintenance personnel is not willing to adopt CBM

The successful implementation of CBM for the VPi300 has already shown that the maintenance personnel is willing to adopt it. All parts are therefore rated "No" on PS-o2 (Table 2).

o3: CBM does not fit with the operational mission

As CPP has defined the proactive maintenance concept it is part of their operational mission to implement CBM. PS-o3 is therefore not a showstopper for any of the parts.

o4: CBM does not fit with the internal and external relations of the company

Maintenance is never outsourced for the VP6000. Therefore, there are no external relations. The internal relations are with the regions that employ the FSTs worldwide. These regions are willing to work with CBM as the implementation of CBM for the VPi300 showed. It therefore does fit with the internal relations of CPP. PS-04 is thus not a showstopper for any of the parts.

o5: CBM does not fit with the availability of spare parts.

The spare part management can be adapted to fit the new situation after the implementation of CBM for any of the parts. PS-o5 is therefore also not a showstopper.

In conclusion, the Preheat 1,2 unit, Preheat 3 unit and the Printhead Seneca are selected for the economic feasibility study.

3.3 Economic feasibility

Tiddens et al. (2018) state that it is difficult to accurately determine the financial benefits of implementing CBM for a part. The authors suggest that a detailed cost benefit analysis can only be conducted if a similar type of CBM has already been implemented. However, this is not the case in this research. It is unknown how well the model will be able to predict failures of the selected parts and there is no comparative case to use as a reference.

As an alternative the authors propose a balanced scorecard approach in this case with the areas: innovation and growth, maintenance, production, customer, society, financial. However, this approach does not really determine the economic feasibility of the selected parts.

A sensitivity analysis to determine the potential cost savings is therefore proposed. The potential cost savings are based on the number of CBM visits and the cost of a CBM visit. This research applies a machine learning (ML) model to predict if a part will fail or not. Every time the ML model predicts a failure a CBM visit occurs. It is therefore necessary to determine the total number of predictions made by the ML model in different scenarios. Furthermore, the cost of a CM visit and a CBM visit are needed to determine the potential cost savings.

First Section 3.3.1 describes the approach to determine the number of CBM visits in different scenarios. Section 3.3.2 describes the cost functions used. The cost factors are described in Section 3.3.3. The sensitivity analysis for Preheat 1,2 unit, Preheat 3 unit and Printhead Seneca are conducted in Section 3.3.4, Section 3.3.5 and Section 3.3.6 respectively.

3.3.1 Number of CBM visits

This research uses a ML model to predict if a part will fail or not. This section describes how to determine the number of CBM visits for different scenarios based on a ML model.

When the ML model predicts a failure, a CBM visit is conducted. The ML model either correctly predicts a failure (true positive (*TP*)) or predicts a failure when there is none (false positive (*FP*)). The total number of CBM visits is therefore equal to all true and false positives combined. Furthermore, the ML model might not predict a failure when there is one (false negative (*FN*)), this would result in a CM visit. In addition, the ML model might correctly predict that there is no failure (true negative). There are no visits related to these and they are therefore not used for the economic feasibility study. For the sensitivity analysis it is assumed that the ML model is able to predict all failures, i.e. no false negatives, therefore the number of true positives is equal to the number of actual failures. As a result, there are only CBM visits. This assumption is made to show the maximum potential savings per part in the sensitivity analysis.

In the ideal situation the ML model only makes correct predictions (*TP*) and no false predictions (*FP*). However, in reality this is likely not the case. ML models are therefore commonly evaluated based on the ratio of correct predictions to all predictions made. This ratio is called the precision (Han, Kamber, & Pei, 2012) and is shown in Equation 3.1.

$$precision = \frac{TP}{TP + FP} \tag{3.1}$$

Given the assumption that the ML model can predict every failure, *TP* is equal to the number of failures. Then it is possible to determine the total number of CBM visits (*TP+FP*) given different precision scores by rewriting Equation 3.1 and the number of failures. The different scenarios in the sensitivity analysis therefore use different levels of precision.

3.3.2 Cost functions per visit

Van Elderen (2016) defined a cost function for a CM visit (Equation 3.2) and for a CBM visit (Equation 3.3) at CPP. Equation 3.4 shows the cost savings of a CBM visit over a CM visit. These cost functions always assume a replacement. The data at CPP only shows when replacements are made, these are therefore a good fit for this research. However, there might also be minimal repairs. These are conducted to get the printer up again, while the part is replaced at a later time. These visits can be prevented by a CBM policy as minimal repairs are conducted after the part has failed. As a result, the cost savings might be higher in reality. The author also defined loss of income due to required slack in the production planning for CM visits; however, after discussion with the Service Product Specialists this cost it is decided to not include this cost factor. The reason is that they could not provide a reasonable estimation of this number.

C_{cbm} : The cost of one CBM visit

C_{cm} : The cost of one CM visit

B : The cost savings of one CBM visit compared to one CM visit.

C_{csq} : The consequence of failure cost per failure

- A failure can potentially cause more damage to the printer than just to the part that failed. The further damage is defined as the consequence of failure cost.

C_{fst} : Cost of FST per hour

C_{loss} : Cost due to loss of income per hour

- CPP generates revenue per click by selling the customer consumables, e.g. ink.

C_{part} : Cost of part

C_{pen} : Penalty cost per visit

- A penalty for customer dissatisfaction for an unscheduled visit (CM visit), as customers prefer a scheduled visit (CBM visit).

C_{S} : Setup costs per visit

- The costs made in preparation of a service visit, i.e. preparation time of the FST and travel expenses (including FST salary).

T_{repair} : Repair time

- Time in hours needed to repair the system.

T_{response} : Visit response time

- The time between the moment when a system goes down and the actual repair starts.

$$C_{\text{cm}} = C_{\text{S}} + C_{\text{pen}} + (C_{\text{loss}} + C_{\text{fst}}) * T_{\text{repair}} + C_{\text{loss}} * T_{\text{response}} + C_{\text{part}} + C_{\text{csq}} \tag{3.2}$$

$$C_{\text{cbm}} = C_{\text{S}} + (C_{\text{loss}} + C_{\text{fst}}) * T_{\text{repair}} + C_{\text{part}} \tag{3.3}$$

$$B = C_{\text{cm}} - C_{\text{cbm}} \tag{3.4}$$

The total costs in case of a CM policy are shown in Equation 3.5. The total costs in case of a CBM policy are defined in Equation 3.6. Every time the ML model makes a prediction (*TP+FP*) a CBM visit occurs. If the ML model fails to predict a failure (*FN*) then a CM visit occurs. Note that for the sensitivity analysis it is assumed that there are no FN.

C_{totalCM} : Total costs in case of a CM policy
C_{totalCBM} : Total costs in case of a CBM policy

(3.5) $$C_{\text{totalCM}} = \text{failures} * C_{\text{cm}}$$

(3.6) $$C_{\text{totalCBM}} = (TP + FP) * C_{\text{cbm}} + FN * C_{\text{cm}}$$

Finally, the cost savings of a CBM policy over a CM policy are described in Equation 3.7. Each time the ML model makes a correct prediction (*TP*) *B* is saved (Equation 3.4), while every time the ML model makes an incorrect prediction (*FP*) then an unnecessary CBM visit occurs (C_{cbm}).

(3.7) $$Total\ cost\ savings = TP * B - FP * C_{\text{cbm}}$$

As each *TP* saves *B* and each *FP* costs C_{cbm} it is concluded that the ratio of *TP* to *FP* determine if there are positive or negative savings, given that *B* and C_{cbm} remain the same. The precision shows the ratio of *TP* to *TP+FP* which can be used to determine the ratio of *TP* to *FP*. For example, if the precision (*TP*:(*TP+FP*)) is 4:5, then the ratio of *TP:FP* is 4:1. It is therefore concluded that the precision needs to be a certain level before costs are saved given that *B* and C_{cbm} remain the same. Furthermore, it is concluded that as the precision increases the cost savings also increase.

3.3.3 Cost factors

Table 4 shows the cost factors per selected part. Equation 3.2 is used to determine the cost of a CM visit (C_{cm}), while Equation 3.3 is used to determine the cost of a CBM visit (C_{cbm}). Every failure always results in a part replacement as the dataset only contains information about parts consumption.

Table 4 - Cost factors per selected part

	Preheat 3	Preheat 1,2	Printhead Seneca
Ccsq	€ -	€ -	€ -
Cfst	€ 87.10	€ 87.10	€ 87.10
Closs	€ 88.35	€ 88.35	€ 88.35
Cpart	€ 1,380.08	€ 2,136.80	€ 2,292.67
Cpen	€ 134.00	€ 134.00	€ 134.00
CS	€ 87.10	€ 87.10	€ 87.10
Trepair (hours)	2.20	2.82	2.86
Tresponse (hours)	8.60	5.15	0.90
Ccm	€ 2,746.94	€ 3,307.65	€ 3,095.06
CCbm	€ 1,853.16	€ 2,718.66	€ 2,881.55
B	€ 893.78	€ 588.98	€ 213.51

B	: Difference between each CM and CBM visit (Equation 3.3)
C_{cbm}	: The cost of one CBM visit based on Equation 3.2
C_{cm}	: The cost of one CM visit based on Equation 3.1
C_{csq}	: None of selected parts have consequence of failure (C_{csq}) costs.
C_{fst}	: The cost of the FST is €87.10 per hour in Europe.
C_{loss}	: The loss of income cost is €88.35 per hour. Defined as the capacity of the printer multiplied by revenue per click.
C_{part}	: Cost of part
C_{pen}	: The penalty for customer dissatisfaction is determined by the Service Product Manager as €134.00
C_S	: Average setup cost per visit. This estimated by the Service Product Specialists as €87.10.
T_{repair}	: The repair time is determined by the Service Product Specialists for each of the selected parts based on expert knowledge and the analysis of visit logs
$T_{response}$	: The response time is also determined by the Service Product Specialists for each of the selected parts based on expert knowledge and the analysis of visit logs

Table 5 shows the costs for the parts based on only CM visits (Equation 3.5).

Table 5 - Failures and maintenance costs 01-01-16 to 01-07-19

Part	Failures	CM costs
Preheat 1,2	204	€ 674,759.66
Preheat 3	237	€ 651,024.39
Printhead Seneca	192	€ 594,251.66

3.3.4 Preheat 1,2 unit economic feasibility

Table 6 shows the sensitivity analysis for Preheat 1,2 unit in relation to the precision and the corresponding cost savings. As can be seen from Table 6, as the level of precision increases the savings also increase. This is expected as increasing the precision reduces the number of false positives (*FP*) and every false positive results in an unnecessary CBM visit with cost C_{cbm} (Equation 3.7). Table 6 shows that if the ML model has a precision of 0.85 or higher then costs can be saved in relation to the current situation (CM visits). It is therefore concluded that for this part a CBM policy is economically feasible.

Table 6 - Sensitivity analysis Preheat 1,2 unit (savings over period 01-01-16 to 01-07-19)

Precision	TP	FP	Total CBM visits	Total CBM costs	Savings
0.60	204	136	340	€ 924,345.24	€ -249,585.58
0.65	204	110	314	€ 853,660.02	€ -178,900.36
0.70	204	88	292	€ 793,849.45	€ -119,089.78
0.75	204	68	272	€ 739,476.20	€ -64,716.53
0.80	204	51	255	€ 693,258.93	€ -18,499.27
0.85	204	36	240	€ 652,479.00	€ 22,280.67
0.90	204	23	227	€ 617,136.38	€ 57,623.28
0.95	204	11	215	€ 584,512.43	€ 90,247.23
1.00	204	0	204	€ 554,607.15	€ 120,152.52

3.3.5 Preheat 3 unit economic feasibility

As can be seen from Table 7, the required precision before Preheat 3 unit saves cost is 0.70. This is lower than for the Preheat 1,2 unit. This is expected as B is higher for Preheat 3 unit than for the Preheat 1,2 unit. As a result, every true positive of the Preheat 3 unit saves more costs compared to a true positive of the Preheat 1,2 unit. Furthermore, the C_{cbm} is lower for the Preheat 3 unit compared to the Preheat 1,2 unit. This means that every false positive cost less in comparison. If the benefits of a correct predictions are higher, while the costs of a wrong prediction are lower it is expected that the required level of precision before costs are saved is also lower. The Preheat 3 unit has the highest potential cost savings of the three parts. Furthermore, it requires the lowest precision level to save costs. It is therefore concluded that CBM is economically feasible for this part.

Table 7 - Sensitivity analysis Preheat 3 unit (savings over period 01-01-16 to 01-07-19)

Precision	TP	FP	Total CBM visits	Total CBM costs	Savings
0.60	237	158	395	€ 731,998.61	€ -80,974.22
0.65	237	128	365	€ 676,403.78	€ -25,379.39
0.70	237	102	339	€ 628,221.59	€ 22,802.80
0.75	237	79	316	€ 585,598.89	€ 65,425.50
0.80	237	60	297	€ 550,388.83	€ 100,635.56
0.85	237	42	279	€ 517,031.93	€ 133,992.46
0.90	237	27	264	€ 489,234.51	€ 161,789.88
0.95	237	13	250	€ 463,290.26	€ 187,734.13
1.00	237	0	237	€ 439,199.17	€ 211,825.22

3.3.6 Printhead Seneca economic feasibility

Table 8 shows the sensitivity analysis for the Printhead Seneca. This part requires the highest level of precision before it achieves cost savings when CBM is implemented, namely 0.95. This is expected as the Printhead Seneca has the lowest B and the highest C_{cbm}. This means that every true positive results in a small benefit compared to the other parts, while every false positive costs more compared to the other parts. As a result, a higher precision score is needed before CBM is economically feasible for the Printhead Seneca. It is therefore the least likely part to result in cost savings out of the selected parts. However, it might still reduce maintenance costs and it is therefore included in the research.

Table 8 - Sensitivity analysis Printhead Seneca (savings over period 01-01-16 to 01-07-19)

Precision	TP	FP	Total CBM visits	Total CBM costs	Savings
0.60	192	128	320	€ 922,095.72	€ -327,844.07
0.65	192	104	296	€ 852,938.54	€ -258,686.89
0.70	192	83	275	€ 792,426.01	€ -198,174.35
0.75	192	64	256	€ 737,676.58	€ -143,424.92
0.80	192	48	240	€ 691,571.79	€ -97,320.13
0.85	192	34	226	€ 651,230.10	€ -56,978.45
0.90	192	22	214	€ 616,651.51	€ -22,399.86
0.95	192	11	203	€ 584,954.47	€ 9,297.18
1.00	192	0	192	€ 553,257.43	€ 40,994.22

3.4 Conclusion

The first step in maintenance funnel is the criticality classification. In this step the maintenance categorization matrix is applied to find the parts CPP that CPP values the most for CBM. This results in 29 potential candidates for CBM. This selection is further reduced for the showstopper analysis as this is a time consuming process for the service product specialists. The parts are reduced based on a Pareto analysis of the average yearly part costs. This results in a selection of ten parts that are responsible for 79% of the average yearly part costs.

In the second step of the maintenance funnel the showstopper analysis is conducted. In this step seven more parts are filtered out.

In the last step of the maintenance funnel an economic feasibility study is conducted. For this step the response times are determined with the service product specialist based on the visit logs. It is concluded that CBM might result in cost savings for all three parts. However, there are large differences in the potential cost savings per part and the required precision to become economically feasible. The Preheat 1,2 unit, Preheat 3 unit, and Printhead Seneca require a precision of 0.85, 0.7 and 0.95 respectively to reduce costs. As expected, selecting parts based on high part price does not necessarily result in selecting the parts with large potential cost savings. This is a major limitation of selecting parts based on high part price and low failure frequency. However, it is concluded that the proposed sensitivity analysis for the economic feasibility study helps to determine if CBM.

4. Data acquisition for the CBM program

The parts selection resulted in three parts that might be candidates for CBM. In this chapter all the data is collected that might be used to predict a failure of the selected part. This chapter answers Research Question 3:

“What event and usage data should be monitored to determine the condition of the selected parts?”

Section 4.1 describes the available event and usage data types. In Section 4.2 the parts and their functions are described. In addition, the available event and usage data per part is described. Finally, Section 4.3 provides a chapter conclusion.

4.1 Available data types

The VP6000 collects event and usage data which might be used to predict a failure. The event data consists of error and warning messages. The usage data contains the usage counters and parameter values.

The warnings are not visible to the customer and never result in downtime. The warnings contain information that is useful to the diagnostic process of the FST that is on-site with the customer. CPP. In contrast to the warnings, the error messages are visible to the customer. Error messages are shown when the printer has stopped working and explain the cause. Some error messages require a restart of the printer to solve it, while others require an action by the operator or the FST. This data is also used by the FST to diagnose problems with the printer. Furthermore, errors and warnings can be used by CPP the analyze the printer population to find possible issues. Lodewijks (2016) developed an early issue detection system based on the error data at CPP. Errors and warning messages are used in literature to predict when a part is going to fail (Wang, Li, Han, Sarkar, & Zhou, 2017) (Sato, Morimoto, & Takata, 2017) (Alcorta, 2017). These might therefore be used as degradation signals.

Furthermore, the VP6000 has counters and parameters. Counters keep count of how many clicks have been produced. There are counters on the entire printer level, however also on a part level. The part counters are reset after a part is replaced. Finally, the parameters are adjustable settings of the printer. An example is the print light value of the printhead. This value describes the light intensity of the LEDs in the printhead. Parameters and counters describe how the printer is used. This is defined as the usage data. The usage data has a major impact on the degradation of a part (Deloux, Fouladirad, & Berenguer, 2016). The counters and parameters of the parts might therefore be used as degradation signals.

All data described in the previous two paragraphs is collected by the FST during a service visit. An FST will connect a laptop to the printer during a service visit, which will then download all the data. Furthermore, the data is automatically uploaded every seven days to CPP for the printers that are connected to the internet (20% of the population in Europe). The event and usage data are currently only available for Europe.

In Chapter 3 the Printhead Seneca, Preheat 1,2 unit, and Preheat 3 unit have been selected for the research. In the following section the function of each of these parts will be explained. Furthermore, all the error types, warnings types, counters and parameters that are available per part are collected.

4.2 Available data per part

This section will describe the available data per part. A summary of the available data can be seen in Table 9.

Table 9 – Summary of available data types per part

Part name	Error types	Warning types	Parameter types	Counter types	Total
Preheat 1,2 Unit	20	4	1	5	30
Preheat 3 Unit	22	9	1	3	35
Printhead Seneca	12	2	3	2	19

4.2.2 Preheat 1,2 unit

The Preheat 1, 2 unit consists of two identical preheating units that share a single motor and control. They are always replaced as a whole unit. The function of the Preheat 1,2 unit is twofold. Firstly, it transports the paper to the Preheat 3 unit. Secondly, it heats the paper to the optimal temperature for printing.

For the Preheat 1, 2 unit there are 24 error types available. In addition, there is one parameter and there are five counters. In total there are 30 possible degradation signals. In appendix D the function of the unit is explained in more detail as are the error types, parameters and counters.

4.2.2 Preheat 3 unit

The Preheat 3 unit is placed after the Preheat 1, 2 unit. It is separate from the Preheat 1,2 since it has another function in addition to paper transportation and preheating. The Preheat 3 unit is responsible for the handoff of the paper to the TTF units. These transfer the image to the paper. This has to be timed precisely right to ensure that the image is printed on the paper in the right place. If the paper is sent to the TTF unit too early, then the image will be printed too low on the paper. If it is sent too late then the image will be printed too high on the paper.

A total of 31 error types, one parameter and three counters are defined for the Preheat 3 unit. In total there are 35 possible degradation signals to measure. In appendix E a detailed explanation of the Preheat 3 unit function is given. In addition, a more detailed overview of the error types, parameters and counters is given in this appendix.

4.2.3 Printhead Seneca

The Printhead Seneca is present in both the primary and secondary printing process of the VP6000. The printheads are identical for these processes. The function of the printhead is to form the image that will be printed.

For each printhead the same errors types, counters and parameters are defined. The only difference is that they are registered for either the primary or secondary printhead. Table 9 shows the total number of errors types, warning types, parameters and counters. There are six error types related to each printhead. Furthermore, one parameter and one counter are available for each printhead. Finally, there is an overall counter value. In total there are nine possible degradation signals that can be measured per printhead. In appendix C, a detailed explanation of the printhead function and the error types, parameters and counters can be found.

4.3 Conclusion

The VP6000 registers errors, warnings, counters and parameters. These might all be used to measure the degradation of a part. All the relevant errors, warnings, counters and parameters have therefore been determined per selected part. It is concluded that a large number of event and usage data types are available per part. For the Preheat 1,2 unit, Preheat 3 unit and the Printhead Seneca there are a total of 30, 35 and 19 event and usage data types respectively.

5. Data processing for the CBM program

In this chapter the Research Question 4 is answered:

"How can the selected event and usage data per part be processed so that it can be used for maintenance decision making?"

This chapter first describes how the data is cleaned in Section 5.1. In Section 5.2 methods to process the for CBM based on event data are discussed. Section 5.3 describes the steps to data processing that are required to prepare the data for machine learning. In Section 5.4 the data is processed into instances and features for machine learning. In Section 5.5 the dataset that resulted from the data processing is described. Feature selection methods from literature are discussed and selected in Section 5.6. This is followed by instance selection methods in 5.7. Section 5.8 describes different machine learning models and a selection is made for this research. How to evaluate a machine learning model's performance is discussed in Section 5.9. In Section 5.10 an experimental setup is described in which several combination of machine learning models with instance and feature selection methods are tested to find the best combination for this research. The results are discussed in Section 5.11. Finally, a chapter conclusion is given in Section 5.12.

5.1 Data cleaning

Figure 8 describes the data cleaning process.

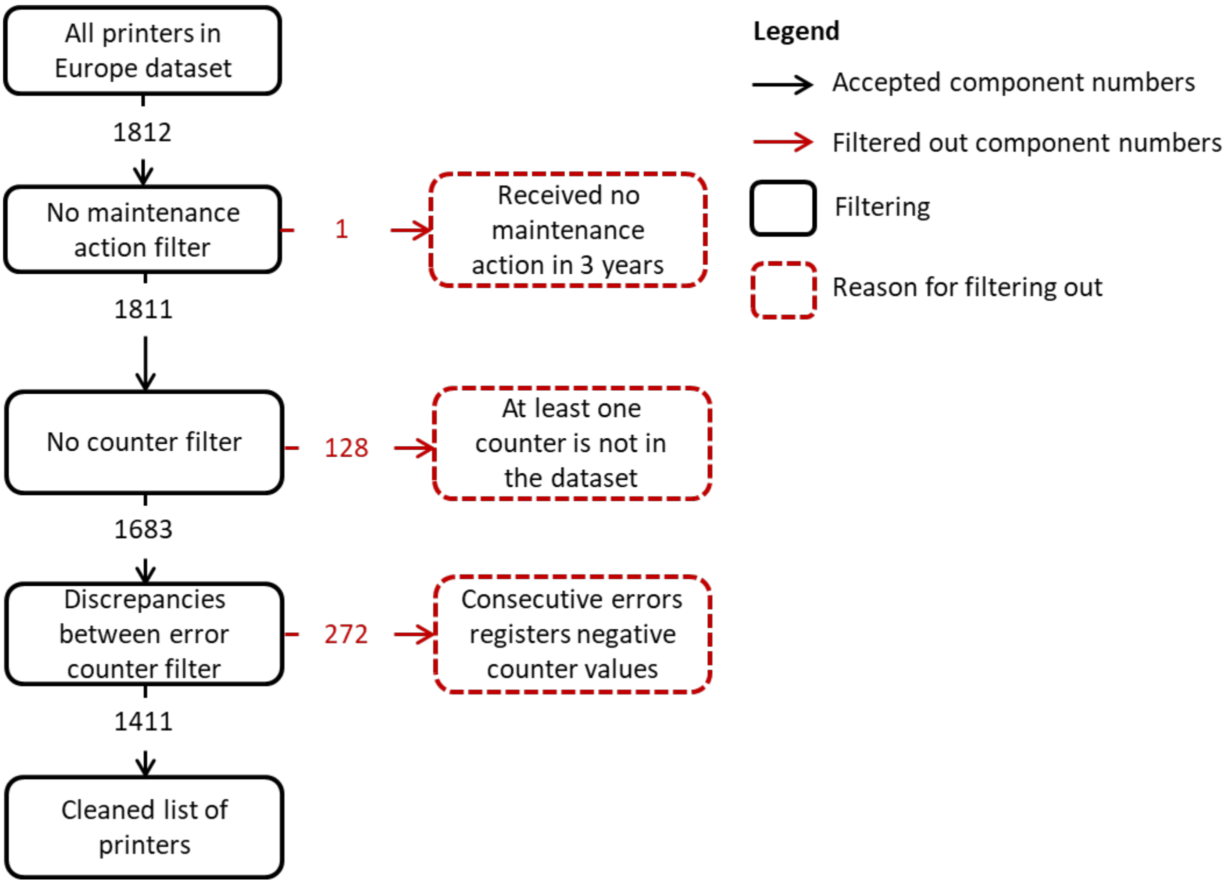


Figure 8 - Data cleaning before processing

- Maintenance action filter: These printers did not receive maintenance in the dataset containing all maintenance actions. This dataset also contains all the consumables. This means that these printers either did not print anything or their maintenance actions are never registered.
- No counter filter: At least one of the counters selected in the previous chapter never appears in the dataset. These should be registered every service visit.
- Discrepancies between error counters filter: Every time an error occurs it registers the level of the overall counter. If the error there is a negative difference between two consecutive errors, the printer is removed from the dataset. It is suspected that this is caused by installing a backup which causes the counter values to return to the level at the time of the backup. However, this is not clear and these printers are therefore removed.

This section uses the event and usage data from the start of 2016 to the end of 2018. This period is expected to representative of the dataset. In addition, it is time consuming for the service data analyst to retrieve this data from the system. Therefore, it is decided to only use three years. Furthermore, only the event and usage data from Europe are available at CPP. This research therefore only uses data from Europe.

5.2 CBM based on event logs in literature

Alcorta (2017) uses a statistical approach for CBM based on errors and warnings. The author uses expert knowledge to determine which errors might indicate the failure of a certain part. A statistical process control (SPC) chart is created for the selected error types. The author determines one rule to indicate a CBM visit, namely when both SPC charts indicate that the errors are out of bounds then a CBM action is advised. Lodewijks (2016) also uses an SPC chart for issue detection for the VP6000 at CPP, however the author's research focusses on finding faults in the whole population, e.g. due to a software update. The benefit of the SPC chart is that it is simple to implement. However, an SPC chart has to be created for each event and usage data type. Furthermore, the number of rules increase exponentially as the number of SPC charts increase. Jardine et al. (2006) warn for this limitation of rule-based reasoning as it can become too computationally intensive. However, not including all event and usage data types might result in the loss of valuable information. This approach is therefore not used.

CBM based on log data is often implemented with machine learning (ML) models (Gutschi et al., 2018; Sipos et al., 2014; Wang et al., 2017). This is due to log data often being of a high dimensional nature, i.e. many different types of log messages. Han, Kamber & Pei (2012) define high dimensional data as data that has 10 or more different characteristics. In the case of this research the error types, warning types, counter codes and parameter types are the characteristics. The Preheat 3 unit, Preheat 1,2 unit and the Printhead Seneca have 35, 30 and 19 data characteristics respectively (Table 9). ML models are best suited to deal with this type of data (Susto et al., 2015). Wang et al. (2017) state that many ML models can be used if the data is processed into a format for Machine Learning. The difference between the research of Gutschi et al. (2018), Sipos et al. (2014) and Wang et al. (2017) is therefore mainly in how the data is processed before using the ML model.

Gutschi et al. (2018) and Sipos et al. (2014) both use system logs, which are different from event logs as these include all system messages, not just errors and warnings. An example of a system message is: "System warming up". Gutschi et al. (2018) state that their data processing method does not use logs based on sensor data. This research uses event data that is based on sensor data. It is therefore expected that the data processing method proposed by Gutschi et al. (2018) cannot be applied to the data of this research. The approach of Sipos et al. (2014) can include sensor data. However, Gutschi et al. (2018) and Sipos et al (2014) both assume continuous data collection. This is not the case at CPP as data is collected every 7 days. The data processing methods of Gutschi et al. (2018) and Sipos et al. (2014) are therefore not used.

The data processing method proposed by Wang et al. (2017) is created for periodical data collection. In addition, their data processing method is based on event logs. Furthermore, the data processing method proposed by Wang et al (2017) has five parameters so that it can easily be adapted for different systems. This makes it possible for CPP to apply the data processing method to more parts and different printers. The data processing method by Wang et al. (2017) is created for periodical data collection, easily customizable for different systems and based on event logs. It is therefore used for this research.

5.3 Data processing steps

The data processing method as proposed by Wang et al (2017) determines if a system is "likely to fail" or "not likely to fail" within a predefined time window. These are defined as classes. For the ML model these classes are described as 1 ("likely to fail") or 0 ("not likely to fail"). This is therefore known as binary classification. To determine if a system is "likely to fail" or "not likely to fail", instances and features are needed. An instance is a learning or testing object for the ML model (Kohavi & Provost, 1998). The features describe the instance for the ML model (Kohavi & Provost, 1998). Each instance receives a class label: "likely to fail" or "not likely to fail". For example, CPP wants to predict if a part is going to fail the next day based on the total number of errors that occurred on this day. The classes are: "likely to fail the next day" (1) and "not likely to fail the next day" (0). Assume 30 days of data is available for one printer. Each day is an instance, while the total number of errors that occurred on that day is the feature. On day 29 a failure occurs. The instance of day 28 receives the class label: 1, as for this instance the part is "likely to fail the next day", while the other instances receive the class label: 0.

Figure 9 shows the steps that this research takes to process the data for classification machine learning.

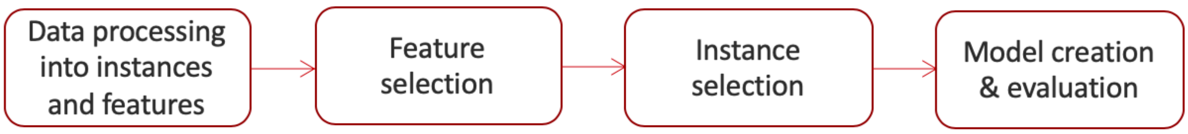


Figure 9 – Data processing steps for classification machine learning

5.3.1 Data processing into instances and features
The first step is to process the data into instances and features. Each instance receives a class label: “likely to fail” or “not likely to fail” in a predefined time window. The data processing method as proposed by Wang et al. (2017) is used for this step.

5.3.2 Feature selection
Not all features that are generated based on the available data might be relevant to predict when a part is “likely to fail”. With feature selection, the set of relevant features are kept, and the irrelevant features are dropped from the dataset. For the example, assume that instead of the total number of errors, each error type is counted individually. If one of these error types never occurs before a failure it is irrelevant. Feature selection would then drop it from the dataset.

5.3.3 Instance selection
It is also possible that the dataset is imbalanced. This means that the classes in the dataset are not evenly distributed. For the example, there is only one instance with label 1 and the remaining instances are labeled 0. The classes are not evenly distributed, and the dataset is imbalanced. Instance selection is used to create more instances for the minority class or remove instances of the majority class to make the dataset more balanced.

5.3.4 Model creation & evaluation
When the instances and relevant features are selected the ML model is built and evaluated. The ML model is built on a train set and evaluated on a test set. These sets are created by dividing the original dataset into two parts. For example: the original dataset of 30 instances could be divided in a train set of 15 instances and a test set of 15 instances. The ML model is built on the train set and has to determine the class labels of the instances in the test set. As the actual label is known, it is possible to evaluate the ML model performance.

The remainder of this chapter first explains the method to process data into instances and features (Section 5.4). This method is applied to the dataset in Section 5.5. Different feature selection methods are discussed in Section 5.6. This is followed by a discussion of different instance selection methods in Section 5.7. Different ML models for are discussed in Section 5.8 and different evaluation methods in Section 5.9. In Section 5.10 an experimental setup is explained in which several combinations of feature selection, instance selection and ML models are tested. The results of this experiment are explained in Section 5.11. Finally, Section 5.12 contains the chapter conclusion.

5.4 Data processing into instances and features
Many ML models can be used to predict failures; however the challenge is processing the event data to instances, features and labels (Wang, Li, Han, Sarkar, & Zhou, 2017). Figure 10 shows a simplified example of the data for one printer with one error type (E1) and one warning (W1) type. The time a part fails is defined as the moment the customer requests a service visit.

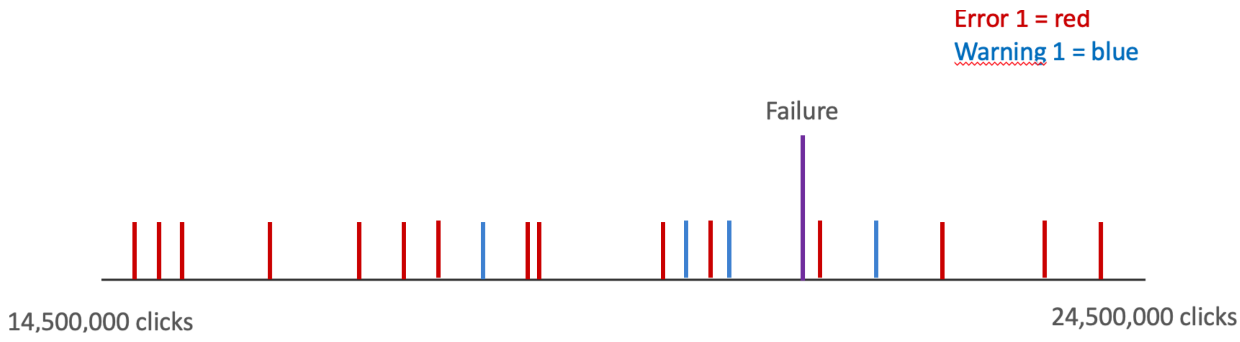


Figure 10 - Simplified example of the data

Figure 11 shows an example of the data processing method as proposed by Wang et al. (2017). The authors create the instances based on days. However, the customers of CPP use the printer's capacity differently. The service product specialists of the VP6000 state that degradation of the selected parts is mostly based on the usage and much less on age. It is therefore better to create the instances based on clicks.

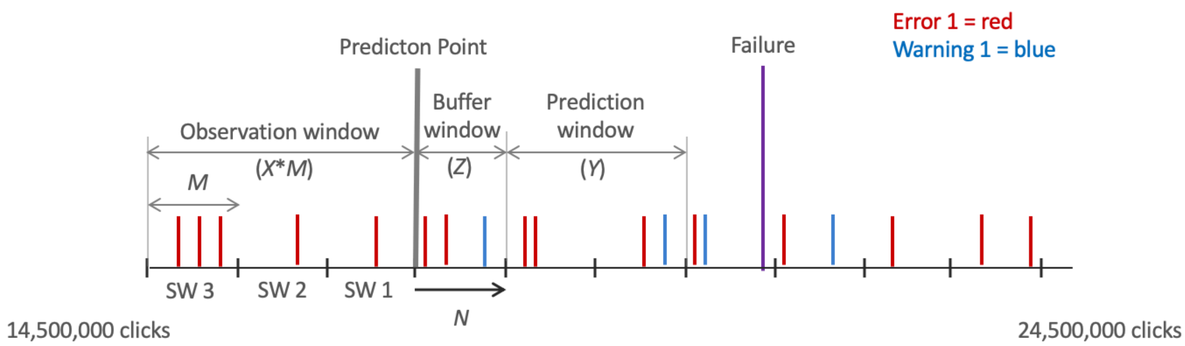


Figure 11 – Data processing example (*X*=3, M = 1 mil clicks, *Z* = 1 mil clicks, *Y* = 2 mil clicks N = 1 mil clicks)

The prediction point is at a moment in time (in clicks) where the prediction is made by the ML model (grey bar in Figure 11). At each prediction point an instance is generated. Before each prediction point there is an observation window (OW). The event data that occurs in this window is used for the creation of features. The observation window consists of X sub-windows (SW) to differentiate between errors and warnings that occur closely to the prediction point and further away. As it is expected that the moment an error or warning occurs closely to the prediction point, then it is more important in determining if a part is "likely to fail". In Figure 11, the observation window consists of three sub-windows (SW) (X = 3). Each SW has the size M. X and M are determined based on the expectations of when the significant errors and warnings occur in relation to the prediction point.

For each instance it is determined if a part is "likely to fail" or "not likely to fail" in the prediction window. If there is a failure in the prediction window the part is labeled: 1 ("likely to fail"), otherwise it is labeled: 0 ("not likely to fail"). The prediction window size is defined as Y. Y is determined based on a business requirement.

The part can fail at any time in the prediction window. A buffer window is therefore implemented to allow CPP time to get to the customer. The size of the buffer window is defined as *Z*. *Z* is determined based on the time that is needed to get to the customer.

The next instance is generated by moving the prediction point forward in time. The number of clicks that the prediction point is moved forward is defined as *Z*. The observation window, buffer window and prediction window are moved forward with the prediction point. The next instance is created based on the new observation window and the new prediction window. Finally, *N* represents how often predictions are made and this is based on how often the data is collected. It is assumed that every time the data is collected a prediction is made for every printer. *N* is therefore determined based on how often the data is collected. *X*, *M*, *Z*, *Y* and *N* are defined as the data processing parameters. Wang et al. (2017) suggest that these data processing parameters need to be optimized. This is discussed in Chapter 6.

The remainder of this section describes the feature sets that are generated for each instance and provides an example of how this process is conducted. Appendix F shows an explanation of all feature sets and their mathematical notations.

5.4.1 Basic statistical features

The number of errors and warnings in each sub-window are counted and added as the basic statistical features. This is done for all error types and warnings per SW. One feature based on Figure 11 is the number of times error type 1 occurs in SW 3 (E1-3). Error type 1 occurs 3 times in SW 3, so for this instance E1-3 is 3. The instance based on Figure 11 is shown in Table 10. To create the next instance the prediction point is moved *N* clicks forward.

Table 10 - Instance 1

Instance	E1-3	E1-2	E1-1	W1-3	W1-2	W1-1	Label
1	3	1	1	0	0	0	0

Figure 12 shows the next instance. As can be seen from the figure, the prediction point is moved forward and the observation window, buffer window and prediction window are moved with it. For this instance, the failure is in the prediction window (*Y*) and it is therefore labeled 1.

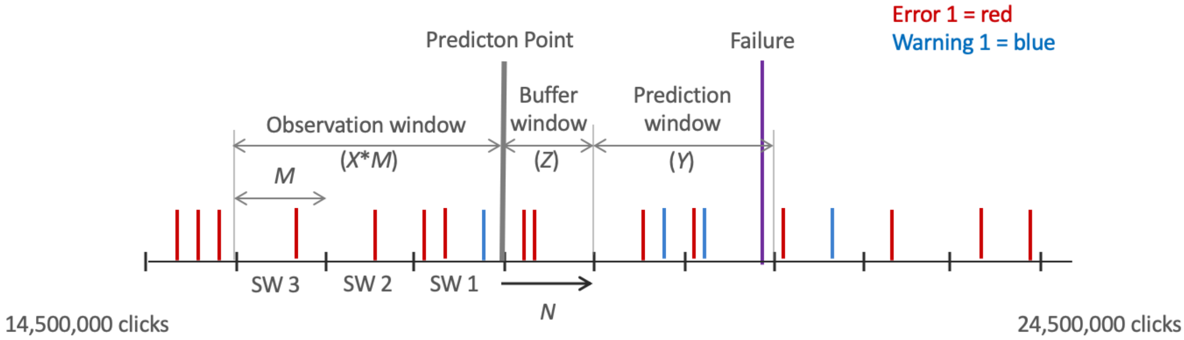


Figure 12 - Generating instance 2

The second instance is created and added to the dataset as can be seen in Table 11. This process is repeated over the entire timeline in clicks for every selected printer.

Table 11 - Instance 2

Instance	E1-3	E1-2	E1-1	W1-3	W1-2	W1-1	Label
1	3	1	1	0	0	0	0
2	1	1	2	0	0	1	1

5.4.2 Pattern-based features

The pattern-based features are used by Wang et al. (2017) to show the relationship between certain types of event data. First, all patterns that occur in each instance are collected. A pattern is defined as the unique error sand warnings that occur in an instance and their combinations. For instance 1 in Table 11 the pattern is: <E1>, while for instance 2 the patterns are: <E1>, <W1>, <E1,W1>. It is expected that including the order in which errors occur would make the patterns too unique. As a result, each pattern would only occur in one instance making them useless for predicting failures of other instances. Therefore, the order of occurrence is not used for the creation of the patterns.

Once the patterns of each instance are collected. The ratio of the pattern's occurrence in positive instances to the occurrence in all instances is determined. A ratio below 0.5 means that the pattern occurs more often in the instances labeled 0. The goal is to predict instances with class label 1, so a threshold value is used to determine which patterns are included. The threshold is set on 0.8 as this ensure that only patterns are included that are most commonly seen in positive instances. It is not expected that there are patterns that only occur in positive instances. Setting the threshold higher therefore makes it unlikely that any patterns are included. Setting it lower increases the risk of adding features that are likely to result in the wrong prediction. The patterns that pass the threshold are included as features. If the selected pattern occurs in the instance the feature receives a 1, otherwise a 0.

In the example, assume that pattern <E1,W1> is included as a feature (Table 12). Instance 1 has no W1, so this pattern does not occur, and it is marked. Instance 2 has E1 and W1, so this pattern occurs, and this feature is marked 1.

Table 12 - Pattern based features (marked grey)

Instance	E1-3	E1-2	E1-1	W1-3	W1-2	W1-1	<E1,W1>	Label
1	3	1	1	0	0	0	0	0
2	1	1	2	0	0	1	1	1

5.4.3 Failure similarity feature

Two failures on the same printer might be preceded by similar event data. The failure similarity feature is added to account for this. First determine all the unique error and warning codes in both the observation windows. Count the number of error and warning types that occur in both instances and divide them by the total number of unique error and warning types from both instances. This is known as the Jaccard index, which is defined in Equation 5.1 (Jaccard, 1912). There is only one failure in the example, so the failure similarity is 0 for both (Table 13). However, assume that a failure occurred on the same printer and only W1 occurred. The Failure similarity score can now be calculated as 1 (due to W1 being in both sets) / 2 (W1 + E1), so it is 0.5.

$$Jaccard\ index = \frac{|A \cap B|}{|A \cup B|} \tag{5.1}$$

Table 13 - Failure similarity feature (marked grey)

Instance	E1-3	E1-2	E1-1	W1-3	W1-2	W1-1	<E1,W1>	F	Label
1	3	1	1	0	0	0	0	0	0
2	1	1	2	0	0	1	1	0	1

5.4.4 Profile features

For the profile features all the machine profile information is gathered. The profiles available for the VP6000 are the configuration type, the country where it is located and the software version. These are added to each instance. Most ML models cannot understand text, so these are added differently. Each country is added as a feature and the one that is true for this printer is marked as 1 the others as 0. This is the same for every software version and every printer model. Table 14 shows an example. Only one country, software version and printer model are included. They are the same as the instances are created for the same printer.

Table 14 - Profile based features (marked grey)

Instance	E1-3	E1-2	E1-1	W1-3	W1-2	W1-1	<E1,W1>	F	NL	SW 6.5	VP6330	Label
1	3	1	1	0	0	0	0	0	1	1	1	0
2	1	1	2	0	0	1	1	0	1	1	1	1

5.4.5 Advanced statistical features

Wang et al. (2017) also propose advanced statistical features. These are based on the distance between two errors of the same type and the distance between each error to the prediction point. These do not match the data at CPP as there are few error and warning types that occur in every observation window. This results in empty fields in each instance as a distance for an error that does not occur cannot be calculated. Most ML models cannot handle empty fields, so these need to be filled. The normal methods to deal with empty fields are to remove the entire instance or to impute the missing value. The first solution will result in all the instances being removed from the dataset as every instance has at least one empty field. The second solution would place a value in a field that should be empty. An example of imputing is using the average of a feature and filling the empty fields with this average. As both solutions will not work for this research, the advanced statistical features are not included.

5.4.6 Counter & parameter features

The VP6000 also collects usage data that Wang et al. (2017) do not include in their data processing method. Therefore, this research adds the counter and parameter features. At the prediction point all the counter values are determined and added as a feature. The same applies for the parameter values. For the example, let's assume one counter (C1) and one parameter (P1). Table 15 shows the addition of these features. The starting counter is 3,500,000 for the example and as N = 1,000,000 the next counter is 4,500,000. P1 is assumed to be 100 first, while it is adjusted to 200 the second instance. However, it can also remain on the same level.

Table 15 - Counter and parameter features (marked grey)

Instance	E1-3	E1-2	E1-1	W1-3	W1-2	W1-1	<E1,W1>	F	NL	SW 6.5	VP6330	C1	P1	Label
1	3	1	1	0	0	0	0	0	1	1	1	3,500,000	100	0
2	1	1	2	0	0	1	1	0	1	1	1	4,500,000	200	1

5.5 Data processing results

The first step before processing the data is determining reasonable data processing parameters (Wang et al., 2017). These are later optimized in Chapter 6. The data processing parameters *M*, *Z*, *N* and *Y* are chosen in multiples of 36,000 clicks. This is the average number of clicks printed per eight-hour workday. This makes it easier to interpret the data processing parameters for the stakeholders as 30 days of average production is easier to interpret than 1,080,000 clicks. The service product specialist (SPS) expects that the most relevant event data for predicting failures occurs closely to the failure. The SPS therefore advises to first look at an observation window of 30 days of average production (30 * 36,000 clicks). It is expected that this window includes the most significant event data. Furthermore, it is expected that *M* is 3 allows for enough differentiation between the significant errors and warnings that occur closely to the prediction point and further away.

The first *Z* is fixed on the minimum buffer window size, namely one. This allows CPP one day (average production) to get to the customer and replace the part. Furthermore, *Y* is fixed on 30. This is a business requirement by CPP. Finally, *N* is determined to be seven as this is the interval at which the event data is downloaded from the printers.

Table 16 describes the dataset after processing it into instances and features. The positive instances are instances in which a failure occurred in the prediction window. The negative instances had no failure in the prediction window.

Table 16 - Processed data description, parameters: *X*=10, (*M*=3, *Z*=1, *Y*=30, *N*=7) * 36,000 clicks

Part name	Negative Instances	Positive instances	Features
Preheat 1,2	107,996	751	374
Preheat 3	107,864	883	441
Printhead Seneca	108,018	729	273

5.6 Feature selection methods from literature

Applying feature selection to the dataset might help to remove irrelevant and redundant features (Khalid, Tehmina, & Shamila, 2014). Furthermore, it might improve the learning accuracy of the ML model and the learning speed (Khalid, Tehmina, & Shamila, 2014). Feature selection creates a subset of features that are the best for the classification; the features that are not in the subset are discarded. In the case of this research, many features are collected, and it is not known how relevant they are. It is therefore expected that the dataset contains irrelevant features. Feature selection is therefore used.

A fast and straightforward approach to feature selection is to take a predefined number (k) of top-ranked features (Berthold, Borgelt, Höppner, & Klawonn, 2010). This method scores all the features based on an evaluation criterion and selects the k top-ranked features. Yang & Pedersen (1997) compared five feature selection criteria for a text classification problem and found that the Chi^2 value provided the best classification results. The Chi^2 test determines the dependability between two variables. Feature selection based on the Chi^2 value therefore filters out features that are independent of the class label. The Chi^2 test works with frequencies and categorical values, it does therefore not work well with the counter features. It is expected that it will filter out these features.

More recently, Ibrahim & Osman (2014) applied feature selection based on the ANOVA F-test and found that it improved classification accuracy and reduced the false positive rate. A low false positive rate is important to CPP. Since, the Chi^2 evaluation criterion outperformed four other criteria and the ANOVA F-test reduced the false positives. This test is based on variance. As such it does not work well with the profile features and it is expected that these are filtered out.

Both feature selection methods are imperfect, however they work for most features. Both are therefore tested to determine which performs the best.

5.7 Instance selection from literature

The number of positive instances in Table 16 compared to the number of negative instances shows that the dataset is highly imbalanced. Training a ML model on an imbalanced dataset might cause the ML model to only predict the most common class as this will give it very high accuracy. However, the ML model is useless as it only predicts one class.

The imbalance in the dataset can be solved by either generating more failure instances or reducing the number of instances without a failure. The first approach is called over-sampling while the latter is called under-sampling. Over-sampling approaches are generally more accurate than under-sampling approaches (Batista, Prati, & Monard, 2004). A basic method for over-sampling is random over-sampling. In this approach the instances from the minority class are randomly duplicated to solve the imbalance. The result is a set of instances that is larger than the original. This method produces good result even when compared to more complex methods (Batista, Prati, & Monard, 2004).

However, the combination of over-sampling and under-sampling provided by SMOTE + ENN is better for highly imbalanced datasets. (Batista, Prati, & Monard, 2004). SMOTE

+ ENN combines the synthetic minority over-sampling technique (SMOTE) with edited nearest neighbor under-sampling. The data is first oversampled by creating new instances that are similar to the instances in the minority class. These new instances are not the same as the original instances. After the dataset is balanced by oversampling it is then reduced via under-sampling resulting in a set of instances that is smaller than the original dataset and evenly balanced. SMOTE+ENN cannot be used with categorical values, i.e. the profile features in this research.

In conclusion, random oversampling is a simple method that produces good results. As it duplicates instances it can be used with any type of data. SMOTE+ENN generates new instances that are closely related to the instances in the minority class. However, it cannot be used for all features. It is therefore decided that random oversampling is used for this research.

5.8 Machine learning models from literature

As stated in Section 5.2, once the data has been converted into a machine learning format it becomes possible to use many different machine learning models.

5.8.1 Random Forest

Fernández-Delgado, Cernadas and Barro (2014) compared 179 ML models for classification on 121 datasets. They concluded that the Random Forest model is the most likely to perform the best. The basic idea of the Random Forest is to create many different uncorrelated decision trees that together can make an accurate prediction. The standard version of the Random Forest creates 100 decision trees (Scikit-learn, 2020) and uses a majority vote for classification. It is called random as each tree is build based on a random sample with replacement from the dataset. This process is used to prevent correlation between the trees. Random Forests employ another technique to reduce correlation between trees as a random selection of features is used at each node instead of all features. From this random selection the split is based on the feature that best separates the data at the split.

The benefits of Random Forest are its ability to handle a variety of data inputs (nominal, categorical and ordinal data), its ability to handle large feature sets, its prevention of overfitting and its high accuracy (Qi, 2012). Furthermore, they are fast as they can efficiently handle large datasets (Han, Kamber, & Pei, 2012, p. 383). The Random Forest is a decision tree learner, as such it does not require the data the be scaled before use, i.e. normalization or standardization of the data. Furthermore, it can be used for binary classification. As it can handle high dimensional data, different data inputs and does not need scaling, it can directly be applied on the processed dataset. It is therefore included in the research. Wang et al. (2017) and Gutschi et al. (2018) use the Random Forest model for CBM based on log data.

5.8.2 Support Vector Machines

The second-best classifier is the Support Vector Machine (SVM) with a Gaussian kernel according to Fernández-Delgado, Cernadas and Barro (2014). SVMs are considered highly accurate, and also prevent overfitting (Han, Kamber, & Pei, 2012, p. 408). They can be used for both linear and non-linear data. However, their major downside is that they are extremely slow (Han, Kamber, & Pei, 2012, p. 408). They are therefore not used in this research.

5.8.3 AdaBoost

AdaBoost is another popular ML model (Han, Kamber, & Pei, 2012, p. 380). Wang et al. (2017) also include this ML model in their research. It is also based on a boosting principle, which combine many weak learners to create an accurate prediction. A weak learner performs slightly better than a random guess (50% change in binary classification). AdaBoost gives every instance in the dataset the same weight to start. It then samples a new dataset based on selection with replacement. It builds a classifier on this set and tries to classify the same set. If an instance is correctly classified its weight is decreased, while if it is incorrectly classified its weight is increased. The weights are used to sample another new set the next round, in which higher weights are favored. Higher weights are the instances that are more difficult to classify. The next classifier is therefore built on the more difficult instances. This way a set of classifiers are build that complement each other and can provide accurate predictions together (Han, Kamber, & Pei, 2012, p. 380).

A disadvantage of AdaBoost is that it is sensitive to outliers and overfitting. Wang et al. (2017) also found that AdaBoost performs worse than XGBoost (explained in the next section) and the Random Forest in predicting failures. It will therefore not be included in this research.

5.8.4 XGBoost

Since the research by Fernández-Delgado, Cernadas and Barro (2014) another ML model has been rising in popularity. This ML model is called XGBoost. XGBoost is outperforming many other ML models in ML contests (Chen & Guestrin, 2016). It can be used for binary classification. Like AdaBoost it combines many weak learners to give an accurate prediction. The difference is that the new learner is trained to predict the residuals of the previous learner.

Its major benefit over other ML models is its speed. It is extremely fast and can handle datasets of hundred million of instances on a desktop (Chen & Guestrin, 2016). Furthermore, it is a decision tree learner and does not require data scaling before use. It can handle different data types and large feature sets. XGBoost can therefore be immediately applied on the processed dataset. Wang et al. (2017) tested multiple machine learning models and found that XGBoost outperformed Random Forest, SVM, and AdaBoost. It is therefore also used in this research.

5.8.6 Selected models for this research

For this research the Random Forest ML model and XGBoost are selected. Both of these models can efficiently handle large datasets, are considered fast. This is important as a number of ML models needs to be trained for the data processing parameter optimization in Chapter 6. The SVMs are considered extremely slow and are therefore not included. AdaBoost is not included as it is sensitive to overfitting. Furthermore, XGBoost and Random Forest are found to outperform AdaBoost in predicting part failures. In addition, Random Forest and XGBoost can handle large feature sets. This is useful as the feature sets grow based on the data processing parameters as for each error and warning type X features are generated.

5.9 Model evaluation

To determine the ML model's performance the dataset is divided into a train and test set. The train set is used to build the ML model. The ML model then has to determine the class label ("likely to fail" or "not likely to fail") for each instance in the test set. These class labels are known and can thus be used to evaluate the model's performance. The test set is unseen by the ML model and no feature or instance selection is applied to this set so that it best represents the real-world situation.

In appendix G, three methods to create train and test sets are explained and compared. It is decided that tenfold randomized stratified cross validation (SCV) is used for this research. This method divides the dataset randomly into ten groups and uses nine groups for training the ML model and one for testing it. This process is repeated ten times so that every group of the dataset is used for the testing process once. Witten et al. (2011) state that numerous tests on different datasets have shown that using ten folds gets the best estimate of performance. This research therefore also uses ten folds. The stratification ensures that each class is equally represented in each of the ten groups.

In Section 3.3.1 it is discussed that a ML model can make a correct prediction (true positive (*TP*)), a prediction that is wrong (false positive (*FP*)) and it can fail to predict a failure that does occur false negative (*FN*)). Furthermore, the ML model can correctly predict that there is no failure (true negative). The positive predictions result in a CBM visit as the ML model classifies the instance as "likely to fail". The false negatives result in a CM visit as the ML model classifies the instance as "not likely to fail", while in reality a failure occurs. The true negatives do not result in a visit and in reality, no failure occurs. There are no costs related to these and they are therefore not included in the evaluation of the ML model.

In Section 3.3.1 it is assumed that there are no false negatives. The performance is therefore evaluated based on the precision of the ML model. The precision is the ratio of the number of correct predictions to all predictions made by the ML model (Equation 5.2). However, in reality this assumption will not hold. Another performance measure is therefore used, namely recall. Recall is the percentage of the actual failures that the ML model is able predict (Equation 5.3). Based on these two scores the ML model is evaluated. However, to evaluate a model's performance based on two scores is a multi-criteria decision-making problem. In Section 5.10 an experiment is conducted to determine the best combination of a feature selection method, instance selection method and a ML model. The goal of this experiment is to find the best combination that can be used for all parts. The relative importance of precision is part specific as can be seen in Section 3.3 as different levels of precision are needed to save costs. As a result, the relative importance of recall is also part specific. Therefore, it is decided that to determine the best combination for all parts the importance of precision and recall are valued equally. For this the F1-score is used, which is the harmonic mean of the precision and recall (Equation 5.4). Harmonic means are used to determine the average of rates. It is therefore the average of the precision and recall score.

(5.2)
$$Precision = \frac{TP}{TP + FP}$$

(5.3)
$$Recall = \frac{TP}{TP + FN}$$

(5.4)
$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

5.10 Experimental setup

There are different ML models, feature selection and instance selection methods that might work to predict failures. It is hard to determine the best options based on literature. The goal is therefore to find the combination of a feature selection method, an instance selection method and a ML model that can be used for all parts. An experiment is therefore conducted. The setup of the experiment is shown in Figure 13. All instance selection, feature selection, stratified cross validation (SCV), and the Random Forest are implemented with the SKlearn package for Python (Scikit-learn, 2020). The XGBoost model is implemented with the XGBoost package (XGBoost, 2020).

The feature selection methods are implemented to select the 150 best features. Wang et al. (2017) found that to be the optimal number. This dataset is based on their data processing method and as such 150 is also used for the test. Setting k as 150 features is large enough to ensure all the most important features are selected, yet small enough to see if feature selection provides better performance. The value for the number of features (k) will be optimized if the tests show that it provides better performance with feature selection than without.

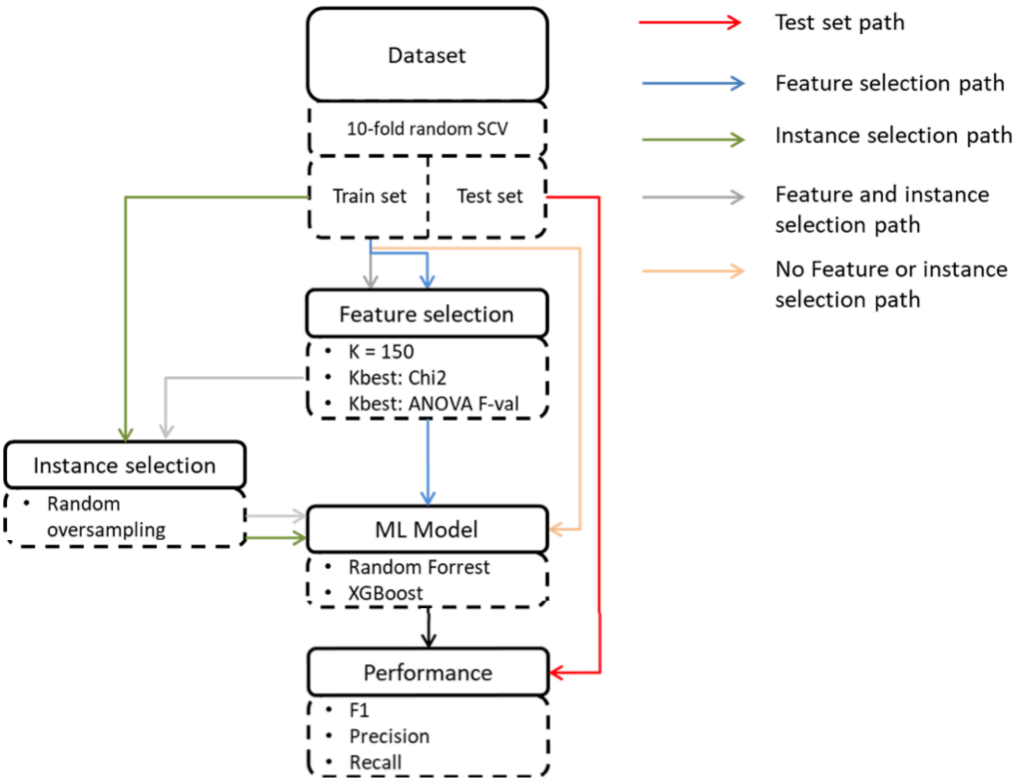


Figure 13 - Experiment setup

Figure 13 shows that there are five paths defined. Each of these paths is tested to determine which ML model, with or without instance and feature selection, performs the best before the data processing method parameter optimization. The feature and instance selection paths are be tried with the selected methods. The best performing methods are used in the combined feature and instance selection scenario. Finally, for every path both ML models are trained and tested to find the best performing ML model.

5.11 Experiment results

The best experiment results (based on the F1-score) are shown in Table 17. The results of all the experiments can be found in appendix H. All of the experiments are performed with X = 10, M = 3 *36,000 clicks, Z = 1 * 36,000 clicks, Y = 30 * 36,000 clicks and N = 7 * 36,000 clicks. These are reasonable first values determined with the service product specialists in Section 5.5. All are multiplied by 36,000 as that is the average number of clicks produced per day. This made it easier for the service product specialists to come up with reasonable values and improves the understandability of the parameters.

Table 17 – Best experiment results randomized SCV (RO = Random Oversampling)

			XGBoost			Random Forest		
Part	Feature Selection	Instance Selection	F1	Precision	Recall	F1	Precision	Recall
Preheat 1,2	None	RO	0.059	0.031	0.694	0.576	0.803	0.449
Preheat 3	None	RO	0.041	0.021	0.626	0.523	0.833	0.382
Printhead	None	RO	0.066	0.034	0.875	0.770	0.851	0.702

As can be seen from Table 17 the best performance is always achieved by the Random Forest (RF) model combined with random oversampling. Appendix H shows that the random oversampling with the RF model increases the F1-score for the Preheat 1,2 unit by 25.39%, Preheat 3 unit by 18.59% and the Printhead Seneca by 2.67% compared to a RF model without instance or feature selection. It has to be noted that the F1-score of the Printhead Seneca was already high.

Another conclusion that can be drawn from Table 17 is that the Random Forest performs significantly better than XGBoost on the precision score. The Random Forest is a decision tree learner that creates 100 decision trees (default setting) and uses a majority vote to classify an instance. It might be that the Random Forest is able to recognize patterns that are printer specific and would hold little value in the real world. Each instance is created by moving the prediction point N clicks forward. The prediction window is 30 * 36,000 while N = 7 *36,000 as a result on average four positive instances are created for the same failure. These are not the same as the observation window is also moved N clicks forward, however it might be that the RF model is able to recognize these patterns and make a prediction based on the information learned from the instances related to the same failure. Due to the randomization performed when applying the randomized stratified cross validation (SCV) the instances related to the same failure most likely end up in different groups. The test set will therefore most likely only contain one positive instance related to the failure while the train set will contain the other three. This would not be a problem if as expected the failure share similar patterns in their feature set, however if every failure has a unique feature set that does not relate to other failures then this would be a problem. It is therefore hypothesized that the RF only performs well because it can recognize patterns within instances related to the same failure and not due to patterns that are shared between instances related to different failures. This hypothesis is tested by applying normal SCV. As a result, the positive instances related to the same failure are placed in the same group and the RF model has to predict failures on printers from which it has not seen any features. The results are shown in Table 18.

Table 18 - Best experiment results normal SCV (RO = Random Oversampling)

			XGBoost			Random Forest		
Part	Feature Selection	Instance Selection	F1	Precision	Recall	F1	Precision	Recall
Preheat 1,2	None	RO	0.019	0.010	0.293	0.027	0.093	0.016
Preheat 3	None	RO	0.026	0.013	0.386	0.037	0.101	0.023
Printhead	None	RO	0.026	0.014	0.271	0.035	0.036	0.034

As can be seen from Table 18, all the ML models perform significantly worse compared to the results in Table 17. The hypothesis is therefore considered true. It is concluded that the failures have a relatively unique feature set. As the positive instances of the same failure are similar, however they are not the same. Still these instances are dissimilar enough to the other instances for the RF to be able to identify a failure based on these features. The data processing method as proposed by Wang et al (2017) should therefore not be combined with random SCV. Instead, SCV without randomization should be used. This is defined as normal SCV.

The results of all combinations of feature selection, instance selection and ML models are poor. These performance scores are so low that no meaningful conclusions can be drawn about the best combination of feature selection, instance selection and ML models. The Random Forest performed slightly better than the XGBoost model. It is therefore decided to use it for the remainder of this research. As the dataset is imbalanced, it is expected that random oversampling will improve the Random Forest's performance if a better set of data processing parameters is found. Based on the results of the random SCV it is concluded that random oversampling significantly improves performance of the Random Forest model if there are patterns in the dataset. It is therefore also used for the remainder of this research. In the next chapter the data processing parameters and the Random Forest model's parameters are optimized to see if better performance can be achieved.

5.12 Conclusion

The available data per part is of a high dimensional nature due to the many available error types, warning types, counters and parameters. ML models are best suited to deal with this type of data and are therefore used for this research. It is concluded that the data processing method as proposed by Wang et al. (2017) best fits the available data types and the periodical data collection. It is therefore used for this research.

There are few positive instances compared to the large set of negative instances, therefore the dataset is imbalanced. The standard method to evaluate a ML model given imbalanced is randomized SCV. However, it is concluded that the randomized SCV is not compatible with the combination of the data processing method as proposed by Wang et al. (2017) and the Random Forest model. This is due to the fact that there are several positive instances created per failure. These have similar feature sets and the Random Forest is able recognize these patterns and predict a failure based on the positive instances related to the same failure. To ensure that the positive instances related to the same failure are always together in the train or test set, the SCV method without randomization is used.

The experimental results are considered too poor to draw meaningful conclusions on which combination of feature selection, instance selection and ML model performs the best. For the remainder of this research the Random Forest with random oversampling is used. It has the highest F1-score. Furthermore, due to the imbalance in the dataset it is expected that random oversampling improves the result if a better combination of data processing parameters is found.

6. Maintenance decision making for the CBM program

This chapter will answer the final research question:

"How can maintenance decisions be made based on the processed data?

Each set of data processing parameters is evaluated to determine the best performing one. Section 6.1 discusses this evaluation process. In Section 6.2, the heuristic to determine the best performing parameters is described. Section 6.3 provides the results of the data processing parameter optimization for all parts. Furthermore, the section provides an example of the process for the Preheat 3 unit. In Section 6.4 the Random Forest parameters are optimized. Finally, Section 6.4 provides the chapter conclusion.

6.1 Data processing parameter evaluation

This chapter optimizes the different parameters of the data processing method as proposed by Wang et al. (2017). Each combination of the data processing parameters results in a different set of instances and features. A Random Forest model with random oversampling is built on each set. The best combination of data processing parameters is then determined based on the performance of the Random Forest models. As stated in Section 5.9 the performance of the Random Forest model is evaluated based on the precision and recall scores. The precision is the percentage of correct predictions made by a ML model in relation to all predictions made. The recall score is the percentage of failures a ML model is able to predict. The best combination of data processing parameters is selected based on the precision and recall scores from the corresponding Random Forest model.

In Section 5.10 the F1-score is used to determine the ML model with the best possible combination of the precision and recall scores. However, the F1-score assumes that the precision and recall are equally important. As is concluded in Section 3.3, each part needs a different level of precision before it can save costs. The importance of the precision and recall is therefore part specific.

It is proposed to evaluate the precision and recall scores based on the cost savings function (Equation 3.7). The best combination of the scores is then determined based on the highest cost savings. Equation 3.7 requires the number of correct predictions (true positives (TP)) and the number of wrong predictions (false positives (FP)). The TP and FP can be determined based on the precision and recall scores. For each of the selected parts it is known how many replacements there were in the dataset. It is therefore possible to determine how many correct predictions the Random Forest model made (true positive), how many times it was wrong (false positive) and how many failures it did not predict (false negative). The total number of actual failures (rep) is equal to TP plus FN (Equation 6.1). The TP (Equation 6.2), FP (Equation 6.3) and FN (Equation 6.4) are determined by rewriting the equations for precision and recall from Section 5.9.

$$rep = TP + FN \tag{6.1}$$

$$TP = recall * \mathrm{rep} \tag{6.2}$$

(6.3) $$FP = \frac{TP}{precision} - TP$$

(6.4) $$FN = \text{r}ep - TP$$

The major benefit of this approach over the F1-score is that the best combination of precision and recall scores are determined based on their cost savings. As the goal of CPP is reducing maintenance costs this approach ensures that the final Random Forest model is selected based on the highest cost savings. Another benefit is that the cost functions are part specific, so the selection of the best performing Random Forest model is also part specific. Furthermore, savings are more easily interpretable for the stakeholders than the F1-score. One limitation of this approach occurs in the case where there are only Random Forest models that result in negative savings to select from. Selecting the Random Forest model with the least negative savings is expected to result in selecting the model that makes less predictions, thus fewer wrong predictions, in favor of a model that has higher precision and recall. This prevents the Random Forest model from potentially gaining positive savings as worse Random Forest models are preferred to minimize the negative savings. In this case it is proposed to use the F1-score until one option with positive savings can be selected.

6.2 Data processing method parameter optimization heuristic

The data processing parameters are: *X*, *M*, *Z*, *Y* and *N*. The parameters: *M*, *Z*, *Y*, *N* and *OW* should all be multiplied by 36,000 clicks which is the average production per day on a VP6000. To improve readability these are given as day values. Each combination of data processing parameters is evaluated by training a Random Forest model with random oversampling based on tenfold stratified cross validation. The best combination in each step is determined based on the approach described in Section 6.1.

Wang et al. (2017) propose the following five steps to find the optimal data processing parameters:

1. Determine the *OW*. Fix *M*, *Z*, *Y* and *N* with reasonable values and change *X*.

First the reasonable values are determined as follows:

- *M* = 5, this allows for easy steps to increase the *OW*. It also makes it possible to make the OW a size that can be divided into multiple combinations of X and M in step 3. Wang et al. (2017) also use *M* = 5.
- *Z* = 1, this is the minimum value for the buffer window and provides CPP one day to get to the customer.
- *Y* = 30, this is a business requirement. CPP does not want the *PW* to be larger than 30 days of average production.
- *N* = 7, as this is the frequency with which CPP downloads the information of the printers.

Table 19 shows the selected observation windows for the parameter optimization. *X* is also shown as *M* is fixed so changing the *OW* is based on *X*.

Table 19 - *X* and *OW* for parameter optimization

X	OW (X*M)
3	15
6	30
8	40
12	60
16	80
20	100

The Service Product Specialist expects that significant errors and warnings happen closely to the failure. The first observation window is therefore determined to be 15. However, the observation window is increased to determine if there are significant errors and warnings further before a failure.

The basic statistical features (Section 5.4.1) increase linearly as *X* increases. As for each error and warning, *X* features are generated. In combination with the number of instances (around 100,000 in case of *N*=7) this can become too memory intensive. A maximum of *X* = 20 is therefore used as increasing X further resulted in memory errors for the Preheat 3 unit. Furthermore, it is not expected that significant errors and warnings occur outside of 100 days of average production.

As stated before, the observation windows are determined by *X***M* with *M* = 5. As a result, the observation window sizes between 15 and 100 are multiples of 5 that can be divided into multiple combinations of *X* and *M*.

2. Determine the prediction window. Fix *X*, *M*, *Z* and *N* with the best values from the previous step and change *Y*.

Wang et al. (2017) conclude that increasing *Y* leads to better model performance. Even though *Y* is fixed on 30, different values are therefore tried to see the effect on the model performance. Additionally, *Y* = 45, 60, 90 and 120 are therefore tested. *Y* = 45 and 60 are tried to see if smaller increases to *Y* can have a significant impact on the model's performance. Then Y is increased to 90 and 120 to see if these large prediction windows increase the model's performance.

3. Determine the best combination of *X* and *M*. Fix *Y*, *Z*, *N* and *OW* with the best values from the previous step and change *X* and *M*.

The *OW* is fixed on the optimal value in step 1. Now several combinations of *X* and *M* are tried to find the optimal values. As stated in step 1, *X* used as larger than 20, as it becomes too memory intensive.

4. Determine the best buffer window. Fix *X, M*, *Y* and *N* with the best values from the previous step and change *Z*.

Increasing Z always reduces model performance (Wang, Li, Han, Sarkar, & Zhou, 2017). Z is therefore fixed on the minimum value of 1. This provides CPP the one day of average production to get to the customer.

5. Determine the best N. Fix X, M, Z and Y with the best values from the previous step and change N.

Since CPP retrieves the information every 7 days for the connected printers, N is also fixed on 7 to represent the current situation at CPP.

6.3 Data processing method parameter optimization results

The best results from the data processing parameter optimization for the selected parts are shown in Table 20.

Table 20 - Best results of data processing parameter optimization

Part name	X	M	Z	Y	N	OW (X*M)	F1	Precision	Recall		Savings
Preheat 1,2 unit	1	30	1	30	7	60	0.034	0.036	0.033	€	-488,525.58
Preheat 3 unit	16	5	1	30	7	80	0.040	0.258	0.022	€	-22,878.15
Printhead Seneca	6	10	1	30	7	60	0.042	0.124	0.025	€	-97,392.43

As can be seen from Table 20, the precision and recall values are low for each part. As a result, the savings for all the parts are negative. It is therefore concluded that for these parts CBM policy based on event and usage data is not possible.

Table 21 shows the steps in optimizing the data processing parameters for Preheat 3 unit. All steps in the data processing parameter optimization for the other selected parts can be found in Appendix I. Appendix J shows the potential of the method based on the results from the random SCV. This appendix shows how to data processing parameter optimization should be conducted based on the proposed evaluation method in Section 6.1.

Table 21 – Data processing parameter optimization Preheat 3 unit based on the F1-score

X	M	Z	Y	N	OW (X*M)	F1	Precision	Recall	Savings
3	5	1	30	7	15	0.014	0.018	0.011	€ -261,517.28
6	5	1	30	7	30	0.037	0.086	0.024	€ -106,510.12
8	5	1	30	7	40	0.020	0.068	0.012	€ -68,491.49
12	5	1	30	7	60	0.019	0.215	0.010	€ -13,745.80
16	**5**	**1**	**30**	**7**	**80**	**0.040**	**0.258**	**0.022**	**€ -22,878.15**
20	5	1	30	7	100	0.010	0.126	0.005	€ -15,442.34
16	**5**	**1**	**30**	**7**	**80**	**0.040**	**0.258**	**0.022**	**€ -22,878.15**
16	5	1	45	7	80	0.019	0.063	0.011	€ -72,074.19
16	5	1	60	7	80	0.017	0.055	0.010	€ -73,337.97
16	5	1	90	7	80	0.018	0.081	0.010	€ -48,395.03
16	5	1	120	7	80	0.023	0.082	0.014	€ -64,243.83
1	80	1	30	7	80	0.028	0.065	0.018	€ -109,301.15
2	40	1	30	7	80	0.033	0.107	0.019	€ -66,293.79
4	20	1	30	7	80	0.028	0.139	0.015	€ -38,757.42
8	10	1	30	7	80	0.025	0.121	0.014	€ -42,122.39
10	8	1	30	7	80	0.023	0.119	0.013	€ -39,116.16
16	**5**	**1**	**30**	**7**	**80**	**0.040**	**0.258**	**0.022**	**€ -22,878.15**
20	4	1	30	7	80	0.023	0.128	0.013	€ -35,651.12

In Section 5.1 it is proposed that in case of only negative savings the best combination of data processing parameters is selected based on the F1-score. As it is expected that the least negative savings are often caused by the model that makes the least predictions. As a result, the data processing parameters that are selected might have lower precision and recall than other sets with higher negative savings. It then becomes more unlikely that improvements can be on the performance in the next optimization steps. Therefore, the F1-score is used. The first part of Table 21 shows an example where this is true. In the first part the OW is optimized. Based on the lowest negative savings OW is 60 should be selected. However, OW is 80 has a higher precision and recall score. Selecting OW is 80 instead of 60 therefore increases the likelyhood that in the next optimization steps the performance increases and positive savings might be achieved.

In the second step, different values of Y are tried. It can be concluded that increasing Y does not lead to better model performance in this case.

Finally, in the last step the optimal combination of X and M is found to be 16 and 5*36,000. The final RF model results in a loss of €22,878.15.

6.4 Random Forest's parameter optimization

The Random Forest model has three important parameters that can be optimized to improve the model's performance (Liu, Chamberlain, & Cardosa, 2017). These parameters are: the number of trees in the forest, the depth of each tree, and the sample size used to create each tree. Optimizing Random Forest's parameters can increase the model performance by a small amount (Probst, Wright, & Boulesteix, 2019). However, it should perform well with the default values (Probst, Wright, & Boulesteix, 2019) (Fernández-Delgado, Cernadas, & Barro, 2015). Even though the Random Forest is resistant to overfitting, it can still overfit. It is decided to apply the Random Forest parameter optimization as a test to determine if it has overfitted on the data. A selection of values is made for each Random Forest parameter. All possible combinations of the selected parameters are tested with tenfold normal stratified cross validation. Appendix K shows the all the optimization results for the Preheat 1,2 unit, Preheat 3 unit and Printhead Seneca.

Section 6.4.1 describes what the number of trees parameter and which values are used for the optimization. Section 6.4.2 describes the depth of each tree and Section 6.4.3 describes the sample size. Section 6.4.4 shows the results of the Random Forest parameter optimization.

6.4.1 The number of trees

A Random Forest models by default builds 100 decision trees (Scikit-learn, 2020). Each tree makes a prediction and a majority vote is used to determine the final prediction. The number of trees can influence the model's performance. The Random Forest model is based on the assumption that the combined prediction of many trees is better than the prediction of a single tree. It is therefore expected that increasing the number of trees allows for better predictions. One downside of increasing the number of trees it becomes more computationally intensive. It is decided to one test one set of fewer trees to see if the results remain the similar and computational time might be reduced. Then the number is trees is increased in large steps. The number of trees used for the parameter optimization are: 50, 100, 250, 500 and 1000. Increasing it to a higher number is expected to become too computationally intensive and it is expected that it will not further improve the performance.

6.4.2 The depth of the tree

The depth of the tree determines the number of splits there are in each tree. The higher the depth the larger the tree becomes. The Random Forest by default allows the tree to grow to the maximum depth, i.e. when there are no more splits possible. By reducing this parameter, the chances of overfitting are reduced. It might therefore improve the performance. The depths used for optimization are: 5, 10 and 20. A lower depth is expected to result in underfitting, while higher depth is expected to result in the same performance as the default feature. Three options are used to reduce the time needed for the Random Forest parameter optimization.

6.4.3 The number of samples used to create the tree

Each tree is built based on a sample from the training dataset. By reducing this sample, the chances of overfitting are reduced. Reducing it too much increases the chances of underfitting. The number of samples used for the optimization are: 20%, 40%, 60%, 80% and 100% of the total sample size. It is decided to use large steps to see the potential effects.

6.4.4 Random Forest parameter optimization results

The results from the Random Forest parameter optimization are shown in Table 22.

Part name	Number of trees	Depth	Sample size (%)	F1	Precision	Recall
Preheat 1,2 unit	100	20	40%	0.036	0.023	0.087
Preheat 3	1000	20	100%	0.049	0.116	0.031
Printhead Seneca	1000	5	80%	0.133	0.076	0.541

Table 22 - Random Forest parameter optimization best results based on the F1-score

It is concluded that the results after the Random Forest parameter optimization are still poor (Table 22). As expected, the results only increased slightly compared to the F1-score of the parts after the data processing parameter optimization (Table 20). One important conclusion that can be drawn based on the results is that overfitting was not the reason the Random Forest performed so poorly. Furthermore, it is concluded that a CBM policy should not be implemented for the selected parts.

6.5 Conclusion

This chapter proposes a method to optimize the data processing parameters based on the cost savings function. The best combination of data processing parameters can therefore be determined based on the cost savings. One limitation of this method occurs when all the combinations to select from have negative savings. The lowest cost might be related to a combination of data processing parameters that do not have the highest precision and recall values. Instead, it might select the option with lower values as it makes fewer predictions and therefore fewer wrong predictions. As a result, it becomes less likely that further optimization of the data processing parameters will result in cost savings. It is therefore concluded that the F1-score is a better alternative in case a selection needs to be made based on only negative savings.

It is concluded that different sets of data processing parameters did not result in cost savings for any of the parts. Furthermore, the Random Forest parameter optimization also did not improve the model's precision to the levels required for cost savings (Section 3.5). The precision and recall scores are still considered poor. It is therefore concluded that a CBM policy is not possible for the selected parts based on event and usage data.

7. Conclusion & recommendations

This chapter contains the conclusion, limitations, future research and recommendations of this research.

7.1 Conclusion

This section answers the main research question by first answered the five research questions defined for this research.

Research Question 1 is: *How to determine which parts are suitable candidates for this research?* CPP loses income during the time the printer is not operational due to the customer not using consumables. Therefore, parts are ideally selected based on the highest average yearly response time. However, the response time has to be manually determined by analyzing visit logs. It is too time consuming to determine the response time for all parts, therefore it is only done for a selection of parts. The selection is first made based on high part costs and low failure frequency. These are the criteria CPP values the most in selecting parts for CBM. It is expected that selecting parts based on their price gives no indication of the potential cost savings. A further study is therefore conducted for these parts to determine if they are suitable for CBM. It is concluded that this approach allows for an efficient selection of parts for this research.

Research Question 2 is: *Which parts of the VP6000 are suitable candidates for this research?* To answer this research question, the method for parts selection from Research Question 1 is applied. The first selection of parts is made based on high part price and low failure frequency. These are further filtered based on a showstopper analysis. It is concluded that the majority of the parts is filtered out as it is not technically feasible to implement a CBM policy for these parts. Finally, a sensitivity analysis is proposed to determine the economic feasibility for the selected parts. Based on this sensitivity analysis it is concluded that there is a large difference in the potential cost savings for each part. As expected, the criteria high part costs and low failure frequency do not help in selecting parts that can result in large cost savings. This is a major limitation of selecting parts based on these criteria. However, the economic feasibility study ensures that only parts that might potentially save costs are included in the research.

Research Question 3 is: *What event and usage data should be monitored to determine the condition of the selected parts?* All the errors, warnings, counters and parameters that are related to the selected parts are collected. For the Preheat 1,2 unit, Preheat 3 and Printhead Seneca are 30, 35 and 19 event and usage data types respectively.

Research Question 4 is: *How can the selected event and usage data per part be processed so that it can be used for maintenance decision making?* The event and usage data types collected to answer the previous research question might all be relevant in predicting if a part is going to fail. Furthermore, there might be combinations between these data types that are useful in predicting part failures. A machine learning model is therefore used as it efficiently finds patterns for these combinations of data types. A method is therefore used to process the data so that it can be used for machine learning.

After the data processing it is concluded that the dataset is imbalanced, as there are far more instances without a failure than with a failure. This is expected as parts with low failure frequency are selected for this research. Randomized stratified cross validation is used to evaluate the machine learning model with an imbalanced dataset. However, it is concluded that this evaluation method cannot be combined with the data processing method used. Instead, stratified cross validation without randomization is used for this research. This is due to the method of instance generation. Several positive instances are generated per failure and the model was able to recognize the similarity of these instances. Based on the results of the randomized stratified cross validation it is concluded that each failure has a unique pattern that can be used to identify instances of related to the same failure, however not instances related to other failures. In addition, an experiment is conducted to determine the best combination of a feature selection method, an instance selection method and ML models. The results of this experiment were too poor to draw meaningful conclusions on. For the remainder of the research it is decided to use Random Forest with random oversampling.

Research Question 5 is: *How can maintenance decisions be made based on the processed data?* To answers this research question, the data processing method parameters are optimized to find the combination that results in the largest cost savings. If the cost savings are positive, then CBM policy based on the best performing machine learning model can be implemented. Otherwise CM is a better policy for the part. It is concluded that for the selected parts different data processing parameters do not increase performance. Furthermore, the same conclusion is drawn for the optimization of the Random Forest's parameters. For the selected parts a CBM policy is therefore not possible based on event and usage data.

The main research question is: *How can event and usage data be used to monitor the condition of the system and how can maintenance decisions be made based on this data?* Event and usage data can be used to monitor the condition of a part by processing it for machine learning. The data processing and the evaluation of the ML model are automated in a Python program. The maintenance decisions are made based on the potential cost savings of the ML model.

This research contributes to academia in several ways. Firstly, a case study is conducted in which the data processing method as proposed by Wang et al. (2017) is applied. Secondly, this research proposes random oversampling to improve the performance of a Random Forest model in case of imbalanced data. Imbalanced data is often the case when creating a CBM program as the ideal parts for CBM have high downtime and low failure frequency. This research found that random stratified cross validation cannot be used in combination with the data processing method as proposed by Wang et al. (2017). Instead, normal stratified cross validation should be used to evaluate the ML model. Finally, the data processing parameter optimization is adapted to make decisions based on the highest potential savings by implementing a CBM policy. Furthermore, in the case of negative savings it can also be decided that a CBM policy is not optimal even though the ML model has high precision and recall scores. Maintenance decision making is therefore added to the data processing method as proposed by Wang et al. (2017). Finally, the usage data features are added to the feature sets proposed by Wang et al. (2017).

7.2 Limitations and future research

Several limitations of the research are described in this section.

A major limitation of this research is that the model did not work for the selected parts. It can therefore not be concluded if it will work for other parts. There are many combinations of the features that might predict a failure. These cannot be checked beforehand. For example, a failure might be predicted by the combination of a certain error type occurring, combined with a usage counter that is at least a certain level and a certain model type of the printer. There are any number of combinations of the features that might predict a failure and it is difficult to see these beforehand. That is why Machine Learning is used. However, the data processing is automated, and it can be tried for other parts.

Another limitation is that this research used two feature selection methods that do not work for all features in the dataset. At the time these decisions were made. A better feature selection method is mutual information gain, which can be used for all the features. Furthermore, due to the time constraints of this research the number of features to include in the dataset are not optimized.

In addition, a limitation is that the data processing parameters are optimized via heuristic approach. Furthermore, as the parameter X increases the number of features increases and it causes memory errors. However, if a computer is used with more processing power and more memory then more combinations of method parameters can be tested, and a better solution might be found.

Future research should be conducted to find out if the data of other parts has as little predictive value as the data used for this research. The model can be adapted to other parts and could be used to determine if the data for these parts is better. Section 7.3 discusses how these parts are best selected.

Furthermore, research should be conducted as to why the errors and warnings have so little predictive value. The errors and warnings might be wrongly defined. These are based on sensor signals, however they are only triggered if a certain threshold is passed. If these thresholds are too low, then the errors and warning would occur too often and not just before a failure. If they are too high, then they might not be triggered often enough.

7.3 Recommendations

In this section several recommendations are made.

Implementation plan

Figure 14 shows the steps CPP has to take to use the Python program.

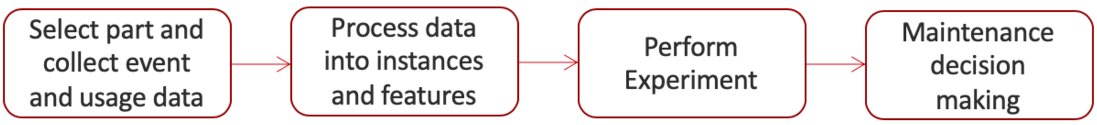


Figure 14 - Steps to use the program

The first step is to select a part and collect all event and usage data. This can then be processed by the Python program.

The second step, processing, requires reasonable values for the method by Wang et al. (2017). The most important is the size of the of the observation window. The size should be the period in which it is expected that significant errors and warnings for predicting a failure occur.

In the third step the experiment can be conducted to see which combination of feature selection, instance selection and ML models performs best. This combination should then be used for the maintenance decision making.

In the fourth and final step, the heuristic approach has to be used to optimize the method parameters. If the potential cost savings are at least positive, then the model can be used.

If CPP wants to implement a model for any part, then the model should be fitted on all the available data. The model should then be exported and imported into their maintenance planning tool. Furthermore, as more data becomes available the model should be refitted. If this process can be automated, then it should be conducted every week when the data is retrieved. Otherwise, it should be trained at least every time there is a new failure of the selected part. Furthermore, it is recommended that if a modification is made to a part for which a model has been implemented then it should be assumed that the model is no longer capable of making accurate predictions. Modifications to a part might change how it degrades, i.e. when errors and warnings are generated. In this case the model's predictions cannot be trusted anymore.

Part selection

The parts for this research are selected based on the maintenance categorization matrix as proposed by van Elderen (2016) combined with the maintenance funnel by Tiddens et al. (2018). The parts selected for this research did not work nor would they have resulted in the highest potential cost savings. It is therefore recommended that the parts are selected based on expert knowledge of the field service technicians (FST) and the service product specialists (SPS). First a selection of parts should be made that are expected to have the highest average yearly response time. Then for these parts, the expert knowledge of the FSTs should be used to determine if they are often replaced by the FST based on the event and or usage data. As all replacements are performed by the FSTs it is expected that this can provide a good approximation if CBM based on event and usage is possible for the selected parts.

Additional recommendations

- For optimal part selection the response time is needed per part. It is therefore recommended that CPP starts to collect the response time of parts.
- The data is collected in seven-day intervals, however it is expected that the most significant errors occur closely to the failure. It is therefore recommended to collect the data continuously.

Bibliography

Alcorta, M. (2017). *MSc thesis: Maintenance policy selection and optimization based on warnings and errors at Vanderlande Industries B.V.* Eindhoven: TU Eindhoven.

Arts, J. (2017, March). Maintenance modeling and Optimization. Eindhoven: Technische Universiteit Eindhoven.: BETA publicatie : working papers; Vol. 526.

Avizienis, A., Laprie, J., Randell, B., & Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing, 1(1)*, 11-33.

Batista, G., Prati, R., & Monard, M. (2004). A study of behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter, 6(1)*, 20-29.

Berthold, M., Borgelt, C., Höppner, F., & Klawonn, F. (2010). *Guide to intelligent data analysis; How to intelligently make sense of real data.* Springer-Verlag London Limited.

Botman, L. (2017). *MSc thesis: Machine failure diagnostics and prognostics using error information and sensor data.* Eindhoven: TU Eindhoven.

Brahimi, M., Medjaher, K., Leouatni, M., & Zerhouni, N. (2016). Critical components selections for a prognostics and health management system design: an aplication to an overhead contact system. *Proceedings of Annual Conference of the Prognostics and Health Management* (pp. 1-8). Denver, USA: HAL.

Canon Production Printing. (2019). Retrieved 07 29, 2019, from cpp.canon.com: https://www.oce.com/about/corporate-information/oce-a-canon-company/

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 785-794). San Francisco, California, USA.

Deloux, E., Fouladirad, M., & Berenguer, C. (2016). Health and usage based maintenance policies for a partially observable deteriorating system. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 230(1)*, 120-129.

Ding, S.-H., & Kamaruddin, S. (2015). Maintenance policy optimization—literature review and directions. *The International Journal of Advanced Manufacturing Technology, 76(5-8)*, 1263–1283.

Fernández-Delgado, M., Cernadas, E., & Barro, S. (2015). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research, 15*, 3133-3181.

Gutschi, C., Furian, N., Suschnigg, J., & Neubacher, D. (2018). Log-based predictive maintenance in discrete parts manufacturing. *12th CIRP Conference on Intelligent Computation in Manufacturing Engineering.* Gulf of Naples, Itlay: Elsevier.

Han, J., Kamber, M., & Pei, J. (2012). What is Cluster Analysis? In *Data Mining Concepts and Techniques.* Morgan Kaufmann.

Ibrahim, O., & Isnab, A. (2014). A novel feature selection based on one-way ANOVA F-test for e-mail spam classification. *Research Journal of Applied sciences, Engineering and Technology, 7*, 625-638.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New phytologist, 11(2)*, 37-50.

Jardine, A., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing, 20(7)*, 1483-1510.

Jenks, G. (1977). Optimal data classification for choropleth maps. Lawrence, Kansas: University of Kansas, department of Geography.

Khalid, S., Tehmina, K., & Shamila, N. (2014). A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning. *Science and Information Conference* (pp. 27-29). London, UK: IEEE.

Kohavi, R., & Provost, F. (1998). Glossary of Terms. *Machine Language - Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*, 271-274. Retrieved from Robotics Standford: http://robotics.stanford.edu/~ronnyk/glossary.html

Labib, A. (1998). World-class maintenance using a computerised maintenance management system. *Journal of Quality in Maintenance Engineering*, pp. 66-75.

Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., & Siegel, D. (2014). Prognostics and health management design for rotary machinery systems - reviews, methodology and applications. *Mechanical systems and signal processing, 42(1-2)*, 314-334.

Liu, C., Chamberlain, B., & Cardosa, D. (2017). Generalising random forest parameter optimisation to include stability and cost. *oint European Conference on Machine Learning and Knowledge Discovery in Databases*, 102-113.

Lodewijks, C. (2016). *BSc thesis: Early quality issue identification of the VP6000.* Eindhoven: Eindhoven University of Technology.

Probst, P., Wright, M., & Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(3).*

Qi, Y. (2012). Random forest for bioinformatics. *Ensemble machine learning*, 307-323.

Real Statistics. (2019). *Real Statistics Resource Pack for Excel*. Retrieved 18 09, 2019, from http://www.real-statistics.com/free-download/real-statistics-resource-pack/

Sato, Y., Morimoto, A., & Takata, S. (2017). Determination of optimum criteria for condition-based maintenance of automatic ticket gates using remote monitoring data. *The 5th International Conference on Through-life Engineering Services* (pp. 190-195). Tokyo: Waseda University.

Scarf, P. (2007). A framework for condition monitoring and condition based maintenance. *Quality technology & quantitative management, 4(2)*, 301-312.

Scikit-learn. (2020). Retrieved 13 3, 2020, from Scikit-learn: https://scikit-learn.org/stable/

Seecharan, T., Labib, A., & Jardine, A. (2018). Maintenance Strategies: Decision Making Grid vs. Jack-Knife Diagram. *Journal of Quality in Maintenance Engineering, 24(1)*, pp. 61-78.

Singh, A., Thakur, N., & Sharma, A. (2016). A review of supervices machine learning algorithms. *Third International Conference on Computing for Sustainable Global Development (INDIACom)*, (pp. 1310-1315). New Delhi.

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).

Susto, G., Schirru, A., Pampuri, S., & Beghi, A. (2015). Machine Learning for Predictive Maintenance: A Multiple Classifiers. *IEEE Transactions on Industrial Informatics 11(3)*, 812-820.

Swanson, L. (2001). Linking maintenance strategies to performance. *International journal of production economics, 70(3)*, 237-244.

ten Have, M. (2019, 8, 1). Information about the VP6000.

Tiddens, W., Braaksma, A., & Tinga, T. (2018). Selecting Suitable Candidates for Predictive Maintenance. *International Journal of Prognostics and Health Management, 9(1)*, 1-14.

van Elderen, C. (2016). *Msc Thesis: Multi-item maintenance optimization based on failure, usage, and condition information.* Eindhoven: TU Eindhoven.

Wang, J., Li, C., Han, S., Sarkar, S., & Zhou, X. (2017). Predictive maintenance based on event-log analysis: A case study. *IBM Journal of Research and Development, 61(1)*, 121-132.

Witten, I., Frank, E., & Hall, M. (2011). Data mining: Practical machine learning tools and techniques. Burlington: Morgan Kaufmann Publishers.

XGBoost. (2020). Retrieved from XGBoost: https://xgboost.readthedocs.io/en/latest/python/python_intro.html

Yang, Y., & Pedersen, J. (1997). A comparative study on feature selection in text categorization, 97(1). *ICML*, 412-420.

Appendices

Appendix A: part usage cleaning

The data cleaning process is shown in Figure 15.

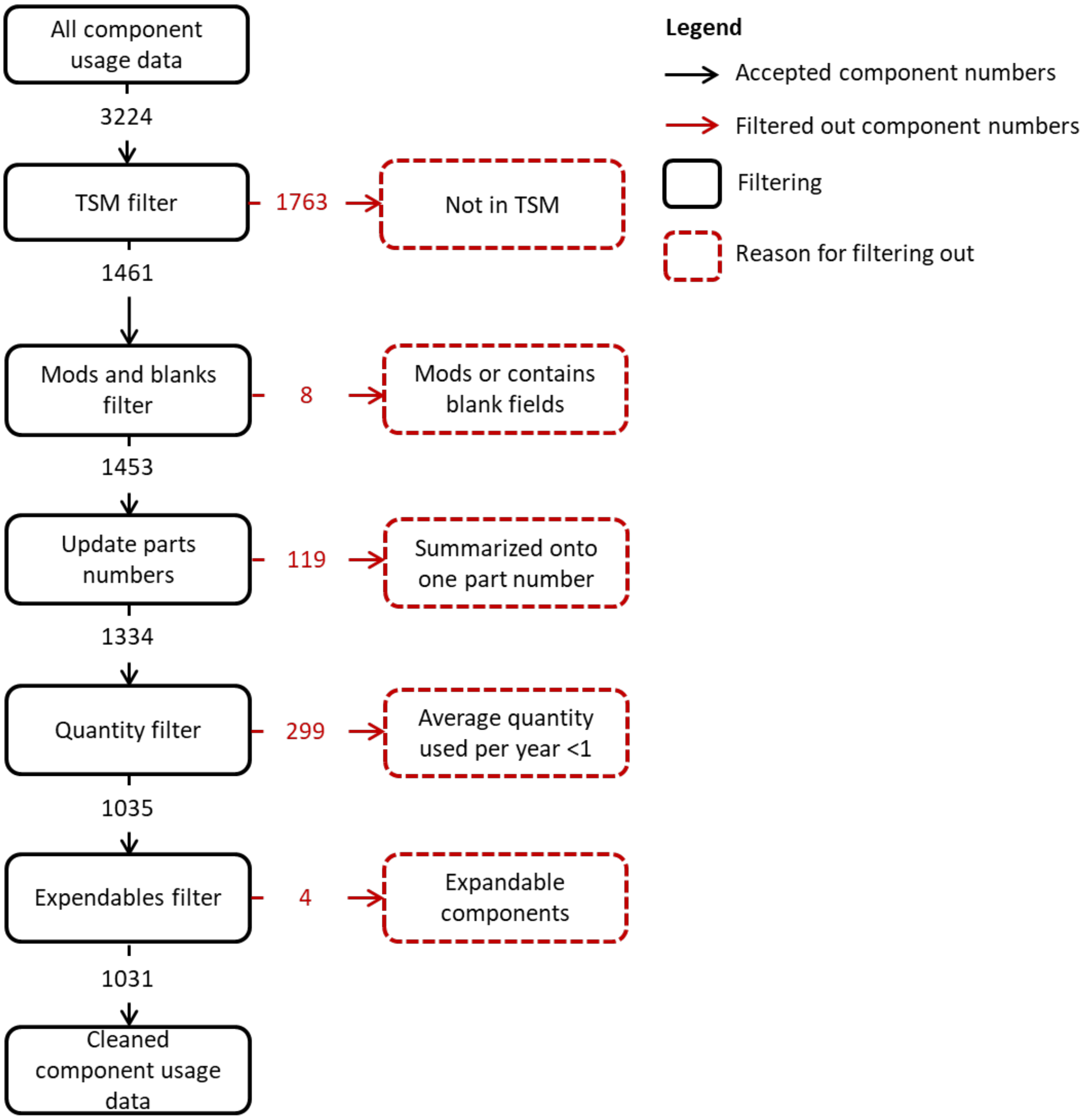


Figure 15 - Data cleaning process

In the period between the start of August 2016 and the end of July 2019, a total of 3,224 unique service parts have been registered for the VP6000 in Europe. Only the European event data is uploaded by the service mechanics and are accessible for CPP. For this reason, it has been decided that for the part usage data the focus will also be on Europe. The decision to use three years of data has been made after discussion with the service data analyst. Going farther back the data will contain failures of parts for which solutions have been implemented. This set contains everything that has been registered on the VP6000. The filters in Figure 15 are explained below.

- TSM filter: The parts that are not in the technical service manual are removed. These parts should not have been registered on the VP6000, however due to human errors they were. A large amount of part numbers is filtered out (54.68%), however mostly there is only one part used per part number. It is therefore not expected that many parts of the VP6000 are registered on other printers.

- Mods and blanks filter: All modifications and parts with blank fields are removed. The modification parts are removed, because these are only installed once on a printer and thus not fit for the research.
- Update part numbers: Parts can receive an update and consequently a new part number. This update is usually the addition of standardized packing to the part. The information of the parts with a changed part number is summarized into their latest part number and the old part number is removed from the dataset. The selected parts for this research are checked to see if they received updates other than packaging, since doing this for every part will be too time consuming.
- Quantity filter: Parts that fail less than once a year are removed from the dataset. The financial benefit of CBM for these parts is considered too low. In discussion with a service product manager it has been decided that if a part is used less than one time on average per year it is also removed from the dataset.
- Expendable filter: These are parts that CPP operates to failure.

After the data cleaning 1031 parts are left in the dataset.

Appendix B: Monitor Closely Quadrant

Table 23 shows all the parts in the Monitor Closely Quadrant. The parts that are not included for further research are made unrecognizable as per request of CPP.

Table 23 - All parts Monitor Closely Quadrant

Part name	Failure Frequency	Part cost	Avg. yearly part costs	Cum. perc. total yearly part costs
Printhead Seneca	0.088	€ 2,292.67	€ 315,623.87	16%
Powerunit	0.095	€ 2,057.99	€ 303,210.66	32%
Preheat 1,2	0.049	€ 2,136.80	€ 173,459.58	41%
Industrial Controller 1	0.032	€ 2,282.27	€ 155,039.07	49%
Preheat 3	0.059	€ 1,380.08	€ 144,489.35	56%
Printhead Cicero	0.029	€ 2,307.57	€ 104,609.99	62%
Main Node	0.047	€ 1,562.49	€ 86,237.35	66%
Develop Unit 1	0.026	€ 2,187.38	€ 84,578.35	70%
Develop Unit 2	0.026	€ 2,072.62	€ 82,940.77	75%
User Interface	0.038	€ 1,216.10	€ 73,371.69	79%
Part A	0.016	€ 2,282.46	€ 70,447.83	82%
Part B	0.007	€ 3,250.85	€ 57,526.67	85%
Part C	0.012	€ 2,187.40	€ 48,687.01	88%
Part D	0.007	€ 3,250.85	€ 43,333.64	90%
Part E	0.015	€ 1,691.20	€ 39,461.24	92%
Part F	0.008	€ 2,512.85	€ 31,829.38	94%
Part G	0.004	€ 2,450.24	€ 17,151.73	94%
Part H	0.004	€ 2,443.85	€ 16,292.38	95%
Part I	0.007	€ 1,596.96	€ 15,969.53	96%
Part J	0.001	€ 4,832.17	€ 11,855.94	97%
Part K	0.003	€ 1,286.35	€ 11,747.12	97%
Part L	0.003	€ 2,133.03	€ 10,665.14	98%
Part M	0.004	€ 1,850.35	€ 9,868.58	98%
Part N	0.002	€ 3,184.05	€ 9,552.16	99%
Part O	0.005	€ 1,104.90	€ 8,102.55	99%
Part P	0.000	€ 3,983.40	€ 4,928.40	100%
Part Q	0.001	€ 1,631.09	€ 3,805.87	100%
Part R	0.001	€ 1,468.33	€ 2,936.66	100%
Part S	0.001	€ 1,200.79	€ 2,717.41	100%
		Total	**€ 1,940,439.94**	**100%**

Appendix C: Printhead Seneca

The printhead Seneca is part of the print engine. The print engine uses a statically charged belt that attracts toner which can then be transferred to TTF Unit. However, if the entire belt was statically charged then the printer would only print black images. The printhead consists of many small LED lights. The light from these LEDs is used to remove the static charge on the belt in specific locations. By removing the static charge the belt does not attract toner in these places, which allows images to be formed on the belt which can be transferred to TTF Unit. Since, the VP6000 has two mirrored printing processes located in the engine there are also two printheads. The location of these printheads is shown in Figure 16. The primary and the secondary printhead are exactly the same component, the only difference being their location in the printer.

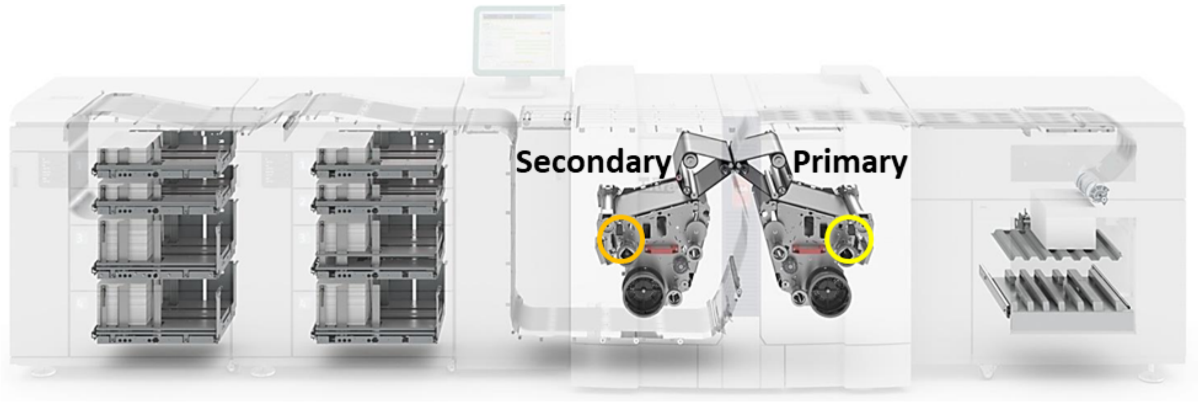


Figure 16 - VP6000 inside view with printhead locations (marked with circles)

For both the printheads individual error types, parameters and counters have been defined. There are twelve error types defined for the printheads, that might indicate degradation however these can be divided into six error types for the primary and six error types for the secondary printhead. These error types are shown in Table 24. Furthermore, there are two warnings defined (Table 25), one per printhead. In addition to the error types and warnings, three parameters have been defined for the printheads (Table 26) and two counters (Table 27).

Table 24 – Error types printhead Seneca

Error Code	Description
4511	Primary LPH communication error
4512	Primary LPH NTC 04R01 short circuit
4513	Primary LPH NTC 04R01 open circuit
4514	Primary LPH temperature too high
4516	Primary LPH logical connection lost
4517	Primary LPH data error

4551	Secondary LPH communication error
4552	Secondary LPH NTC 04R11 short circuit
4553	Secondary LPH NTC 04R11 open circuit
4554	Secondary LPH temperature too high
4556	Secondary LPH logical connection lost
4557	Secondary LPH data error

Table 25 - Warnings Printhead Seneca

Warning Code	Description
4914	Primary LPH temperature high
4954	Secondary LPH temperature high

Table 26 - Parameters Printhead Seneca

Parameter Code	Description
41103	Print counter
41110	Print Light Value Primary Process [0.1%]
41210	Print Light Value Secondary Process [0.1%]

Table 27 - Counters printhead Seneca

Counter Code	Description
48101	Prints made with primary LED printhead [prints]
48201	Prints made with secondary LED printhead [prints]

Appendix D: Preheat 1, 2 unit

The Preheat 1, 2 unit has two functions, namely transporting the paper to the TTF Units and heating the paper to the temperature required for optimal transfusion. Figure 17 shows the location of the Preheat 1, 2 unit in the VP6000. The Preheat 1, 2 Unit consists of two identical preheating units, hence the name Preheat 1, 2 unit.

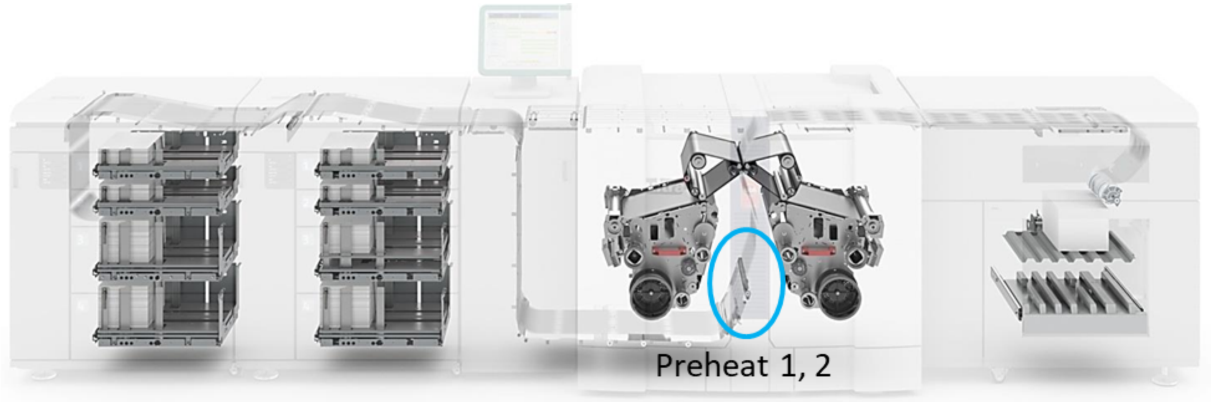


Figure 17 - VP6000 inside view with Preheat 1,2 unit locations (marked with blue circle)

For the Preheat 1, 2 Unit 20 error types are found that might indicate degradation. (Table 28). In addition, there are four warning types (Table 29). There is only one parameter defined (Table 30) and there are five counters (Table 31).

Table 28 - Error types Preheat 1,2 unit

Error Code	Description
13123	LWM: Remove sheets from Print Engine Module Preheat unit, Output unit and spiral cleaners
15151	Remove sheets from Print Engine Module Preheat unit, Output unit and TTF cleaners
15501	Preheater plate 1 temperature too low
15502	Preheater plate 1 temperature too high
15504	Preheater plate 1 NTC 15R1 short circuit
15505	Preheater plate 1 NTC 15R1 open circuit
15507	Warming up timeout preheater plate 1
15508	Preheater plate 1 clixon 15S1 open circuit
15511	Preheater plate 2 temperature too low
15512	Preheater plate 2 temperature too high

15514	Preheater plate 2 NTC 15R2 short circuit
15515	Preheater plate 2 NTC 15R2 open circuit
15517	Warming up timeout preheater plate 2
15518	Preheater plate 2 clixon 15S2 open circuit
15551	Sheet error in preheat trajectory
15564	Powerdown detected at local node preheat
2210001	Remove sheets from Print Engine Module Preheat unit, Output unit and TTF cleaners
2250024	Configuration error Preheat node
2250036	Communication error Preheat node
15519	PRE2MO 15M2 speed error

Table 29 - Warning types Preheat 1,2 unit

Warning Code	Description
15904	Preheater fan 15M3 speed error
15905	Preheater moisture fan 15M13 speed error
2290109	Inconsistent embedded software Preheat node
2290133	CAN BUS error detected by Preheat node

Table 30 - Parameter Preheat 1,2 unit

Parameter Code	Description
141003	Registration Sensor PWM Value [0.1%]

Table 31 - Counters Preheat 1,2 unit

Counter Code	Description
158011	Sheets made with preheater belt 1 [prints]

158012	Sheets made with preheater plate 1 [prints]
158013	Sheets made with preheat 1-2 rolls [prints]
158021	Sheets made with preheater belt 2 [prints]
158022	Sheets made with preheater plate 2 [prints]

Appendix E: Preheat 3 unit

The Preheat 3 unit is located after Preheat 1,2 unit. It has a similar function to the Preheat 1,2 unit as it heats the paper to the optimal temperature for transfusion and transports it to the TTF units. However, the Preheat 3 unit actually hands the paper over to the TTF units. The TTF units transfer the image to the paper. The paper has to be sent to the TTF units at exactly the right time otherwise the image will not be transferred to the paper at the right position. This process is called alignment. If the paper is sent to the TTF units too early, then the image will be printed lower on the paper than intended and sending it too late will result in the image being printed higher than intended. In both cases a part of the image might not even be printed on the paper.

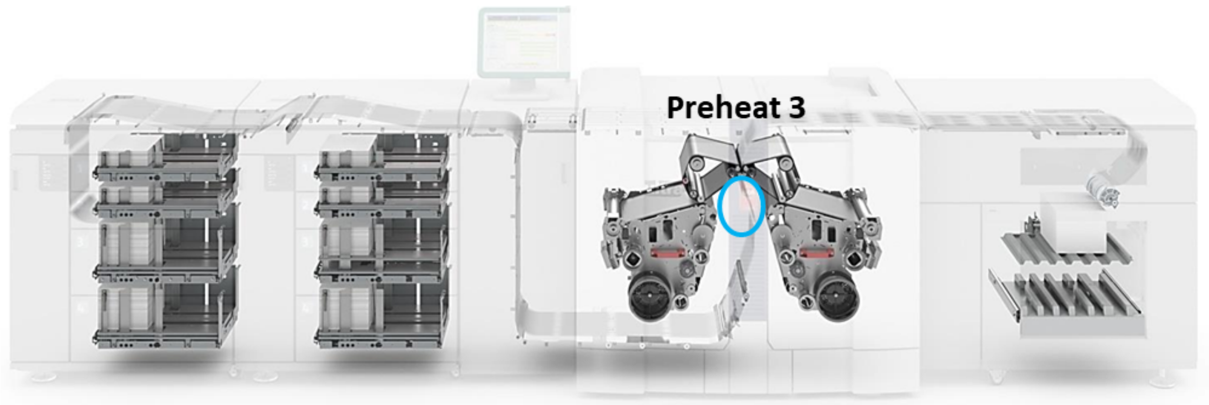


Figure 18 - VP6000 inside view with Preheat 3 unit location (marked with blue circle)

For the Preheat 3 Unit there are 22 error types that might indicate degradation (Table 32). In addition there are nine warning types (Table 33), one parameter (Table 34) and three counters (Table 35).

Table 32 - Error types Preheat 3

Error Code	Description
13123	LWM: Remove sheets from Print Engine Module Preheat unit, Output unit and spiral cleaners
15151	Remove sheets from Print Engine Module Preheat unit, Output unit and TTF cleaners
15521	Preheater plate 3 temperature too low
15522	Preheater plate 3 temperature too high
15524	Preheater plate 3 NTC 15R10 short circuit
15525	Preheater plate 3 NTC 15R10 open circuit
15527	Warming up timeout preheater plate 3

15528	Preheater plate 3 clixon 15S10 open circuit
15529	PRE3REGMO 15M11 speed error
15530	PRE3MO 15M10 speed error
15551	Sheet error in preheat trajectory
15560	Primary SOP sync warning
15561	Secondary SOP sync warning
15562	Primary synchronization speed too low
15563	Secondary synchronization speed too low
15564	Power down detected at local node preheat
15721	PRE3REGSE 15B3 Too Late Free
15722	Unexpected sheet at PRE3REGSE 15B13
15723	Sheet too late at PRE3REGSE 15B13
2210001	Remove sheets from Print Engine Module Preheat unit, Output unit and TTF cleaners
2250024	Configuration error Preheat node
2250036	Communication error Preheat node

Table 33 - Warning types Preheat 3

Warning Code	Description
15901	Led current PRE3REGSELED 15B12 too high.
15902	PRE3REGSELED 15B12 / PRE3REGSE 15B13 polluted.
15903	PRE3REGSE 15B13 read error
15904	Preheater fan 15M3 speed error
15905	Preheater moisture fan 15M13 speed error
15912	Preheater 3 drive motor 15M11 speed too low
15913	Preheater 3 drive motor 15M11 speed too high
2290109	Inconsistent embedded software Preheat node

2290133	CAN BUS error detected by Preheat node

Table 34 - Parameter Preheat 3

Parameter Code	Description
141003	Registration Sensor PWM Value [0.1%]

Table 35 - Counters Preheat 3

Counter Code	Description
158031	Sheets made with preheater belt 3 [prints]
158032	Sheets made with preheater plate 3 [prints]
158033	Sheets made with preheat 3 rolls [prints]

Appendix F: Feature sets

A ML model is trained with instances and features. An instance is learning or testing object for the model (Kohavi & Provost, 1998). The features describe the instance for the machine learning model (Kohavi & Provost, 1998). Wang et al. (2017) propose five feature sets, namely: basic statistical features, advanced statistical features, pattern features, failure similarity features and profile features.

(F.1) $$B = \{c_{ij}, i \in [1,T], j \in [1,X]\}$$

The authors propose a basic statistical feature set B (Equation F.1). In this set the occurrences per error type per sub-window are counted. This results in feature set B with T error types and X sub-windows.

(F.2) $$A = \{\min(D_i), \max(D_i), mean(D_i), mean(V_i), stdDev(V_i), i \in [1,T]\}$$

The next set proposed by Wang et al. (2017) are the advanced statistical features A (Equation F.2). For these features the distance of each error per error type i to the prediction point is calculated. The minimum, maximum and mean of the set D per error type i are used as features. In addition, for each error type i the intervals between all the consecutive error occurrences are determined. The mean and standard deviation of this set V per error type T are used as features of the instance. These do not match the data at CPP as there are few errors that occur in every observation window. This results in empty fields in each instance as a distance for an error that does not occur cannot be calculated. Machine Learning models cannot handle empty fields, so these need to be filled. The normal method to deal with empty fields is to remove the entire instance or to impute the missing value. The first solution will result in all the instances being removed from the dataset as every instance has at least one empty field. The second method would place a value in a field that should be empty. An example of imputing is using the average of a feature and filling the empty fields with this average. In this case a distance that does not exist cannot be filled with the average distance of that feature. This would train the model with false information. Consequently, this feature set is not included.

(F.3) $$P = \{p_i, i \in [1,Q]\}$$

The authors also incorporated patterns as features P (Equation F.3). An apriori algorithm is used to mine for frequently occurring patterns and determine their confidence in predicting a failure (Wang, Li, Han, Sarkar, & Zhou, 2017). Patterns that pass a certain threshold for confidence are selected. The selected patterns are gathered in set Q. If a pattern of set Q is in the observation window then p_i is one, otherwise p_i is 0.

Failure similarity features

(F.4) $$Jaccard\ index = \frac{|A \cap B|}{|A \cup B|}$$

Furthermore, the authors propose failure similarity feature F. The set of error and warning types that occur in the observation window is compared the set from the observation window of the most recent failure of the same printer. The Jaccard index between these two sets is used as feature F (Wang, Li, Han, Sarkar, & Zhou, 2017). The Jaccard index is defined in Equation F.4 (Jaccard, 1912).

Profile features

In the feature set R all the machine profile information is gathered. The profiles available for the VP6000 are the configuration type, the country where it is located and the software version.

(F.5) $$S = \{B, A, P, F, R, L\}$$

Each instance S (Equation F.5) consists of all the features described before and has a label L. The label indicates whether there was a failure in the instance's prediction window or not.

The authors based their features on error and warning messages. However, the VP6000 also collects parameters and counters. Consequently, more features can be added to the features proposed by Wang et al. (2017).

(F.6) $$C = \{co_i, i \in [1, K]\}$$

Firstly, the counter features C (Equation F.6) are added to the instance. For all the counters in K the value at the end of the observation window is included. These counters are for subparts of the part or the part itself and they have the same increase in clicks as the overall counter for the selected parts. The observation window is based on the overall counter, so all the counters will have the same increase in clicks. Including the increase in clicks for these counters will therefore not result in more information for the model, thus they are not included. Finally, the overall counter value at the end of the observation window is included.

(F.7) $$W = \{w_i, i \in [1, U]\}$$

Secondly, the set of parameter features W (Equation F.7) is added to the instance. The set U includes all the parameter types. Their current level is included as a feature.

Appendix G: Model evaluation

Each combination of ML models with feature selection and or instance selection has to be evaluated. Furthermore, each combination of method parameters has to be evaluated. This is the only way to compare the different scenarios and draw conclusions which performs the best. There are three methods commonly used for model evaluation. These are: the hold-out method, k-fold cross validation (CV) and stratified k-fold cross validation (SCV). This appendix explains the differences between these model evaluation methods. All of these methods divide the data into a train and a test set. The train set is used to train the model, while the test set is used to test the model performance on unseen data. The test set is never altered in any way by for example feature or instance selection. The dataset is created by generating each instance per printer. As a result the dataset is ordered chronologically per printer based on overall counter of the printer at the end of the observation window.

Train set : Marked in blue

Test set : Marked in green

Table 36 - The hold-out method (based on Preheat 1,2 unit)

Partition	1	2	3	4	5	6	7	8	9	10
Size of partition	10875	10875	10875	10875	10875	10875	10874	10874	10874	10874
Positive instances	109	97	100	87	78	57	117	28	43	35

Table 36 shows an example of the hold-out method with a 70 – 30 split for train and test data. model is thus trained based on the blue part of Table 36 and tested on the green part. The major drawback is that you cannot be certain if the train set and test are representative of the total dataset (Witten, Frank, & Hall, Data mining: Practical machine learning tools and techniques, 2011, p. 152). As can be seen in Table 36, the dataset is divided into ten groups of equal size and the positive instances per group are counted. This is done to illustrate that fewer positive instances occurred in the last 30% of the data when compared to the first 70%. It can therefore be concluded that this split is not representative of the entire dataset. Wang et al. (2017) apply this method, however in the case of this research it is not a good fit.

Table 37 - Tenfold cross validation first three iterations of ten (based on Preheat 1,2 unit)

First iteration	1	2	3	4	5	6	7	8	9	10
Size of partition	10875	10875	10875	10875	10875	10875	10874	10874	10874	10874
Positive instances	109	97	100	87	78	57	117	28	43	35
Second iteration										
Size of partition	10875	10875	10875	10875	10875	10875	10874	10874	10874	10874
Positive instances	109	97	100	87	78	57	117	28	43	35
Third iteration										
Size of partition	10875	10875	10875	10875	10875	10875	10874	10874	10874	10874
Positive instances	109	97	100	87	78	57	117	28	43	35

An alternative to the hold-out method is cross validation. This method makes sure that every instance is used for testing once and all the other iterations it is used for training. As can be seen from Table 37 the dataset is divided into ten groups. Each iteration (fold) of the tenfold CV method uses nine of the groups for training data and one of the groups to test the model. The evaluation scores of each iteration are averaged to provide the final evaluation scores. This method ensures that all the data is used for training and testing and thus it is more representative of the dataset. The drawbacks of this approach are that it is more computationally intensive than the hold-out method as a model is trained ten times instead of once. Furthermore, as can be seen from Table 37 a group might not be a good representation of the dataset. For example, group 8 only has 28 positive instances. Given the data available for this research this approach is not optimal as the test set might return very different evaluation scores each iteration. This approach will therefore also not be used.

Table 38 - Stratified tenfold cross validation first three iterations of ten (based on Preheat 1,2)

First iteration	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Size of partition	10875	10875	10875	10875	10875	10875	10874	10874	10874	10874
Positive instances	75	75	75	75	75	76	75	75	75	75
Second iteration										
Size of partition	10875	10875	10875	10875	10875	10875	10874	10874	10874	10874
Positive instances	75	75	75	75	75	76	75	75	75	75
Third iteration										
Size of partition	10875	10875	10875	10875	10875	10875	10874	10874	10874	10874
Positive instances	75	75	75	75	75	76	75	75	75	75

Stratified tenfold cross validation operates in a similar manner as normal CV. However, the major difference is that it places an equal number of positive instances in each group (Table 38). The standard approach to model evaluation is tenfold SCV where the data is randomly divided into ten parts (Witten, Frank, & Hall, Data mining: Practical machine learning tools and techniques, 2011, p. 153). The stratification combined with the randomization increases the likelihood that the train and test sets are representative of the total dataset. This dataset is ordered on serial numbers and then on the click counter of the prediction point. Randomization will ensure that each group contains a combination of instances from different serial numbers and at different clicks for the prediction point. In case of limited data (few positive instances) this method is preferred over standard CV. It will therefore be used for this research.

Appendix H: Experiment results

This appendix shows the experiment results for the selected parts with randomized stratified cross validation and normal stratified cross validation.

Experiment results randomized stratified cross validation

C : Chi2

F : ANOVA F-val

RO : Random Oversampling

Table 39 - Experiment results Preheat 1,2 unit random SCV

			XGBoost			Random Forest		
Scenario	Feature Selection	Instance Selection	F1	Precision	Recall	F1	Precision	Recall
1	None	None	0.021	0.517	0.011	0.445	0.817	0.306
2	C	None	0.029	0.592	0.015	0.354	0.833	0.225
3	F	None	0.026	0.717	0.013	0.397	0.820	0.262
4	None	RO	0.059	0.031	0.694	0.576	0.803	0.449
5	F	RO	0.058	0.030	0.676	0.537	0.786	0.407

The best results based on the F1-score are achieved by scenario 4 with the Random Forest model for the Preheat 1,2 unit (Table 39). The Random Forest in scenario 2 has 0.030 higher precision, however the recall is much lower. Random Forest combined with random oversampling therefore performs the best. The random oversampling method increases the F1-score with 25.39% for the RF model without feature or instance selection (scenario 1).

Table 40 - Experiment results Preheat 3 unit random SCV

			XGBoost			Random Forest		
Scenario	Feature Selection	Instance Selection	F1	Precision	Recall	F1	Precision	Recall
1	None	None	0.022	0.492	0.011	0.441	0.840	0.299
2	C	None	0.013	0.450	0.007	0.306	0.732	0.194
3	F	None	0.013	0.500	0.007	0.347	0.811	0.221
4	None	RO	0.041	0.021	0.626	0.523	0.833	0.382
5	F	RO	0.084	0.055	0.179	0.407	0.603	0.307

As can be seen from Table 40, the Random Forest ML model outperforms the other ML models on the F1-score in every scenario. The Random Forest model combined with random oversampling has the best performance on the initial set. It has the highest F1-score, the second highest precision and the highest recall. The difference with the highest precision (scenario 1) is just 0.007, while the recall is 0.083 higher than this scenario. It can be concluded that the F1-score is increased by 18.59% by random oversampling for the RF model compared to RF without feature or instance selection.

Table 41 - Experiment results Printhead Seneca random SCV

			XGBoost			Random Forest		
Scenario	Feature Selection	Instance Selection	F1	Precision	Recall	F1	Precision	Recall
1	None	None	0.013	0.367	0.007	0.750	0.861	0.664
2	C	None	0.016	0.367	0.008	0.737	0.857	0.646
3	F	None	0.019	0.350	0.010	0.721	0.825	0.641
4	None	RO	0.066	0.034	0.875	0.770	0.851	0.702
5	C	RO	0.064	0.033	0.878	0.755	0.844	0.683

It can be concluded from Table 41 that scenario four with the Random Forest Model again performs the best based on the F1-score. The precision is 0.01 lower than in scenario 1, however the recall is 0.038 higher than scenario 1 for the Random Forest model.

In conclusion, the Random Forest model has the highest performance compared to XGBoost. Furthermore, the random oversampling significantly improved the F1-scores for Preheat 1,2 and Preheat 3. It also provided a small increase of the F1-score for the Printhead Seneca. As a result, it is concluded that random oversampling improves the performance of ML model for this imbalanced dataset.

Experiment results normal stratified cross validation

C : Chi2

F : ANOVA F-val

RO : Random Oversampling

Table 42 - Experiment results Preheat 1,2 normal SCV

			XGBoost			Random Forest		
Scenario	Feature Selection	Instance Selection	F1	Precision	Recall	F1	Precision	Recall
1	None	None	0.000	0.000	0.000	0.024	0.127	0.013
2	C	None	0.002	0.010	0.001	0.022	0.113	0.012
3	F	None	0.000	0.000	0.000	0.015	0.109	0.008
4	None	RO	0.019	0.010	0.293	0.027	0.093	0.016
5	C	RO	0.019	0.010	0.284	0.017	0.028	0.012

As can be seen from Table 42, the results with normal SCV are much worse for the Preheat 1,2 unit compared to randomized SCV. It can be observed that highest F1-score is 0.027 is achieved by the Random Forest combined with random oversampling. These also have the highest precision, which is most valued by CPP. It has to be noted that although it is the highest it is still low.

Table 43 – Experiment results Preheat 3 normal SCV

			XGBoost			Random Forest		
Scenario	Feature Selection	Instance Selection	F1	Precision	Recall	F1	Precision	Recall
1	None	None	0.000	0.000	0.000	0.026	0.080	0.016
2	C	None	0.000	0.000	0.000	0.006	0.043	0.003
3	F	None	0.001	0.002	0.001	0.010	0.061	0.006
4	None	RO	0.026	0.013	0.386	0.037	0.101	0.023
5	F	RO	0.024	0.012	0.390	0.023	0.050	0.015

Table 43 shows the experiment results for Preheat 3 based on normal SCV. The highest F1-score is again achieved by the Random Forest combined with the random oversampling. Furthermore, it also achieves the highest precision.

Table 44 - Experiment results Printhead Seneca normal SCV

			XGBoost			Random Forest		
Scenario	Feature Selection	Instance Selection	F1	Precision	Recall	F1	Precision	Recall
1	None	None	0.000	0.000	0.000	0.022	0.018	0.027
2	C	None	0.000	0.000	0.000	0.023	0.019	0.027
3	F	None	0.000	0.000	0.000	0.026	0.030	0.023
4	None	RO	0.026	0.014	0.271	0.035	0.036	0.034
5	C	RO	0.029	0.015	0.289	0.034	0.036	0.033

Table 44 shows the experiment results based for the Printhead Seneca based on normal SCV. The highest F1-score is achieved by the RF model combined with random oversampling.

Appendix I: Parameter optimization based on normal SCV

This appendix shows the results of the parameter optimization for the parts based on normal SCV. These results do represent the model performance fairly. The parameter optimization is conducted based on the F1-score. Due to the model performance selecting the best parameters on savings would result in selecting the lowest recall to reduce the FP. However, the goal is now to see how well the model can perform.

Preheat 1,2 unit

In the dataset there have been 204 failures of the Preheat 1,2 unit. All of these were CM visits with a total cost of €674,759.66 (204 * €3,307.65 (Table 4)). The results for the combinations of the different parameters are shown in Table 45. The savings are in relation to the total cost of a CM policy.

Table 45 - Parameter optimization Preheat 1,2 unit normal SCV (*M, Z, Y, N*) * 36,000 clicks

X	M	Z	Y	N	OW (X*M)	F1	Precision	Recall	Savings
3	5	1	30	7	15	0.017	0.025	0.013	€ -278,601.02
6	**5**	**1**	**30**	**7**	**30**	**0.026**	**0.056**	**0.017**	**€ -161,233.86**
8	5	1	30	7	40	0.013	0.022	0.009	€ -231,130.60
12	5	1	30	7	60	0.009	0.020	0.006	€ -152,482.97
16	5	1	30	7	80	0.024	0.044	0.017	€ -202,977.76
20	5	1	30	7	100	0.013	0.047	0.008	€ -86,757.85
6	**5**	**1**	**30**	**7**	**30**	**0.026**	**0.056**	**0.017**	**€ -161,233.86**
6	5	1	45	7	30	0.008	0.018	0.005	€ -162,670.38
6	5	1	60	7	30	0.016	0.079	0.009	€ -57,015.48
6	5	1	90	7	30	0.019	0.032	0.013	€ -221,880.30
6	5	1	120	7	30	0.016	0.030	0.011	€ -191,388.91
1	**30**	**1**	**30**	**7**	**60**	**0.034**	**0.036**	**0.033**	**€ -488,525.58**
2	15	1	30	7	60	0.027	0.090	0.016	€ -87,673.77
3	10	1	30	7	60	0.019	0.046	0.012	€ -136,117.74
5	6	1	30	7	60	0.024	0.048	0.016	€ -172,963.55
6	5	1	30	7	60	0.026	0.056	0.017	€ -161,233.86
10	3	1	30	7	60	0.027	0.093	0.016	€ -84,769.99
15	2	1	30	7	60	0.030	0.099	0.017	€ -85,291.49

The best results are achieved based on $X = 1$, $M = 30 * 36{,}000$, $Z = 1 * 36{,}000$, $Y = 30 * 36{,}000$ and $N = 7 * 36{,}000$. However, the results are very poor as the negative savings show.

Printhead Seneca

A total of 190 failures of the Printhead Seneca occurred in the dataset. The costs for only CM visits are determined to be €594,251.66 (190 * €3,095.06 (Table 4)). The results of the different combinations of the parameters are shown in Table 46. The savings are in relation to the CM only policy.

X	M	Z	Y	N	OW (X*M)	F1	Precision	Recall	Savings
3	5	1	30	7	15	0.034	0.107	0.020	€ -91,820.81
6	5	1	30	7	30	0.033	0.034	0.033	€ -513,196.09
8	5	1	30	7	40	0.037	0.108	0.022	€ -99,541.47
12	**5**	**1**	**30**	**7**	**60**	**0.040**	**0.127**	**0.024**	**€ -89,292.24**
16	5	1	30	7	80	0.023	0.021	0.026	€ -681,550.95
20	5	1	30	7	100	0.037	0.124	0.022	€ -84,314.09
12	**5**	**1**	**30**	**7**	**60**	**0.040**	**0.127**	**0.024**	**€ -89,292.24**
12	5	1	45	7	60	0.041	0.118	0.025	€ -100,750.04
12	5	1	60	7	60	0.023	0.104	0.013	€ -61,660.88
12	5	1	90	7	60	0.051	0.175	0.030	€ -76,653.06
12	5	1	120	7	60	0.046	0.200	0.026	€ -55,290.94
1	60	1	30	7	60	0.040	0.119	0.024	€ -95,901.26
2	30	1	30	7	60	0.042	0.119	0.025	€ -101,312.86
4	15	1	30	7	60	0.040	0.120	0.024	€ -95,337.57
6	**10**	**1**	**30**	**7**	**60**	**0.042**	**0.124**	**0.025**	**€ -97,392.43**
10	6	1	30	7	60	0.042	0.120	0.025	€ -100,819.57
12	5	1	30	7	60	0.040	0.127	0.024	€ -89,292.24
15	4	1	30	7	60	0.040	0.129	0.024	€ -87,876.22

Table 46 - Parameter optimization Printhead Seneca normal SCV, (*M, Z, Y, N*) * 36,000 clicks

The best combination of parameters is $X = 6$, $M = 10*36{,}000$, $Z = 1*36{,}000$, $Y = 30*36{,}000$ and $Z = 7 * 36{,}000$. However, these results are very poor.

Appendix J: results based on randomized SCV

This appendix shows the results of parameter tuning for the parts based on the randomized SCV model evaluation. These results do not fairly represent the ML model's performance due to the random SCV. However, these are included to show the potential of the methods if there were recognizable patterns in the dataset.

Since, the increasing *Y* always leads to better model performance according to Wang et al. (2017) and the models perform well it has been decided to fix *Y* on 30 * 36,000.

Preheat 1,2 unit

In the dataset there have been 204 failures of the Preheat 1,2 unit. All of these were CM visits with a total cost of €674,759.66 (204 * €3,307.65 (Table 4)). The results for the combinations of the different parameters are shown in Table 47. The savings are in relation to only CM visits.

Table 47 - Method parameters Preheat 1,2 random SCV (*M, Z, Y, N*) * 36,000 clicks

X	M	Z	Y	N	OW (X*M)	F1	Precision	Recall		Savings
3	5	1	30	7	15	0.515	0.708	0.405	€	-43,976.42
6	5	1	30	7	30	0.579	0.801	0.453	€	-7,988.10
8	5	1	30	7	40	0.577	0.784	0.456	€	-14,887.22
12	**5**	**1**	**30**	**7**	**60**	**0.610**	**0.867**	**0.470**	**€**	**16,484.96**
16	5	1	30	7	80	0.630	0.847	0.502	€	10,024.76
20	5	1	30	7	100	0.627	0.827	0.505	€	2,087.84
1	**60**	**1**	**30**	**7**	**60**	**0.679**	**0.882**	**0.552**	**€**	**25,366.26**
2	30	1	30	7	60	0.667	0.876	0.539	€	22,447.45
4	15	1	30	7	60	0.656	0.885	0.521	€	25,052.24
6	10	1	30	7	60	0.628	0.872	0.491	€	19,022.47
10	6	1	30	7	60	0.622	0.872	0.483	€	18,712.53
12	5	1	30	7	60	0.610	0.867	0.470	€	16,484.96
15	4	1	30	7	60	0.614	0.865	0.476	€	15,991.38

The first section of Table 47 shows that *X* and *OW* are changed (marked in red). The best OW size is 60 * 36,000 clicks. In the second section of Table 47 the optimal combination of *X* and *M* (marked in red) with *Z, Y, N* and *OW* fixed is determined. It is found that *X* = 1 and *M* = 60 * 36,000 clicks result in the highest potential savings.

In conclusion, *X* =1, *M* = 60 * 36,000, *Z* = 1 * 36,000, *Y* = 30 * 36,000 and *N* = 7 * 36,000 is found to give the best results with a precision score of 0.882 and recall of 0.552. This results in the potential savings of €25,366.25, a reduction of 3.76%.

Preheat 3 unit

The Preheat 3 unit is fails 237 times in the dataset. The total costs without CBM are €651,024.16 (237 * €2,746.94 (Table 4)). The results of different combination of the method parameters are shown in Table 48.

Table 48 - Method parameters Preheat 3 unit random SCV (M, Z, Y, N) * 36,000 clicks

X	M	Z	Y	N	OW (X*M)	F1	Precision	Recall	Savings
3	5	1	30	7	15	0.481	0.764	0.351	€ 26,730.89
6	5	1	30	7	30	0.530	0.829	0.390	€ 47,279.88
8	5	1	30	7	40	0.525	0.819	0.386	€ 44,298.01
12	5	1	30	7	60	0.512	0.806	0.375	€ 39,792.10
16	**5**	**1**	**30**	**7**	**80**	**0.557**	**0.847**	**0.415**	**€ 54,983.09**
20	5	1	30	7	100	0.546	0.823	0.409	€ 48,003.53
1	**80**	**1**	**30**	**7**	**80**	**0.768**	**0.863**	**0.692**	**€ 98,335.27**
2	40	1	30	7	80	0.743	0.864	0.651	€ 92,892.51
4	20	1	30	7	80	0.687	0.866	0.569	€ 81,859.75
8	10	1	30	7	80	0.583	0.847	0.444	€ 58,825.28
10	8	1	30	7	80	0.553	0.839	0.412	€ 52,548.56
16	5	1	30	7	80	0.557	0.847	0.415	€ 54,983.09
20	4	1	30	7	80	0.548	0.832	0.408	€ 50,241.44

The first section of Table 48 shows that the optimal OW is 80 * 36,000 clicks. It is found by changing X (marked in red). In the second section the optima combination of X and M is found by trying multiple combinations (marked in red). It can be observed that X =1 and M = 80 * 36,000 clicks are optimal.

In conclusion the optimal $X = 1$, $M = 80$ * 36,000 clicks, $Z = 1$ * 36,000 clicks, $Y = 30$ * 36,000 and $N = 7$ * 36,000. This resulted in the potential savings of €98,355.27, a reduction of 15.14%, based on a precision of 0.863 and recall of 0.692.

Printhead Seneca

A total of 190 failures of the Printhead Seneca occurred in the dataset. The costs are determined to be €594,251.66 (190 * €3,095.06 (Table 4)). The results of the different combinations of the parameters are shown in Table 49.

Table 49 - Method parameters Printhead Seneca (*M, Z, Y, N*) * 36,000 clicks

X	M	Z	Y	N	OW (X*M)	F1	Precision	Recall	Savings
3	5	1	30	7	15	0.788	0.846	0.738	€ -43,612.03
6	5	1	30	7	30	0.778	0.851	0.717	€ -39,644.77
8	5	1	30	7	40	0.773	0.842	0.714	€ -44,388.86
12	**5**	**1**	**30**	**7**	**60**	**0.778**	**0.864**	**0.707**	**€ -32,248.01**
16	5	1	30	7	80	0.775	0.854	0.709	€ -37,600.07
20	5	1	30	7	100	0.762	0.840	0.697	€ -44,411.05
1	60	1	30	7	60	0.823	0.867	0.783	€ -33,997.72
2	30	1	30	7	60	0.817	0.862	0.776	€ -36,536.18
4	15	1	30	7	60	0.806	0.863	0.756	€ -35,038.13
6	10	1	30	7	60	0.793	0.866	0.732	€ -32,317.08
10	6	1	30	7	60	0.779	0.858	0.713	€ -35,681.19
12	**5**	**1**	**30**	**7**	**60**	**0.778**	**0.864**	**0.707**	**€ -32,248.01**
15	4	1	30	7	60	0.775	0.858	0.707	€ -35,380.93

As can be seen from Table 49, implementing CBM would result in large losses compared to the current CM policy. It can be concluded that the method and ML model work well, since the precision score is 0.864 and recall is 0.707. However, since the difference in costs for the CBM visit and the CM visit are too small; each false positive results in a large amount of extra costs, while a true positive only results in a small cost reduction. CBM is not a good policy for this part based on this ML model

Appendix K: Random Forest parameter optimization results

This appendix shows all the results from the Random Forest parameter optimization per part. These scores are based on tenfold normal stratified cross validation.

K.1 Preheat 1,2 unit Random Forest parameter optimization

	Depth	Number of trees	Sample size	Precison	Recall	F1
0	5	50	20%	0.012	0.453	0.024
1	5	50	40%	0.013	0.459	0.025
2	5	50	60%	0.013	0.450	0.026
3	5	50	80%	0.013	0.447	0.026
4	5	50	100%	0.013	0.469	0.026
5	5	100	20%	0.013	0.470	0.026
6	5	100	40%	0.013	0.457	0.026
7	5	100	60%	0.014	0.455	0.027
8	5	100	80%	0.014	0.461	0.026
9	5	100	100%	0.013	0.458	0.026
10	5	250	20%	0.014	0.467	0.027
11	5	250	40%	0.014	0.475	0.027
12	5	250	60%	0.013	0.462	0.026
13	5	250	80%	0.013	0.457	0.025
14	5	250	100%	0.012	0.445	0.024
15	5	500	20%	0.014	0.469	0.028
16	5	500	40%	0.014	0.469	0.028
17	5	500	60%	0.014	0.470	0.028
18	5	500	80%	0.014	0.467	0.027
19	5	500	100%	0.014	0.463	0.027
20	5	1000	20%	0.013	0.463	0.026
21	5	1000	40%	0.014	0.466	0.027
22	5	1000	60%	0.014	0.467	0.026
23	5	1000	80%	0.014	0.469	0.026
24	5	1000	100%	0.014	0.470	0.027
25	10	50	20%	0.013	0.284	0.025
26	10	50	40%	0.013	0.276	0.025
27	10	50	60%	0.014	0.304	0.026
28	10	50	80%	0.014	0.310	0.026
29	10	50	100%	0.014	0.313	0.027
30	10	100	20%	0.013	0.270	0.024
31	10	100	40%	0.013	0.269	0.024
32	10	100	60%	0.014	0.296	0.026
33	10	100	80%	0.014	0.305	0.027
34	10	100	100%	0.014	0.288	0.026
35	10	250	20%	0.014	0.282	0.026

36	10	250	40%	0.014	0.285	0.026
37	10	250	60%	0.013	0.278	0.026
38	10	250	80%	0.014	0.281	0.026
39	10	250	100%	0.014	0.304	0.028
40	10	500	20%	0.014	0.285	0.026
41	10	500	40%	0.014	0.277	0.026
42	10	500	60%	0.014	0.276	0.026
43	10	500	80%	0.014	0.282	0.026
44	10	500	100%	0.014	0.282	0.026
45	10	1000	20%	0.013	0.281	0.026
46	10	1000	40%	0.013	0.277	0.025
47	10	1000	60%	0.014	0.288	0.026
48	10	1000	80%	0.014	0.280	0.026
49	10	1000	100%	0.014	0.285	0.026
50	20	50	20%	0.017	0.093	0.028
51	20	50	40%	0.018	0.080	0.030
52	20	50	60%	0.016	0.076	0.026
53	20	50	80%	0.015	0.071	0.025
54	20	50	100%	0.015	0.075	0.025
55	20	100	20%	0.019	0.081	0.031
56	20	100	40%	0.023	0.087	0.036
57	20	100	60%	0.013	0.069	0.022
58	20	100	80%	0.020	0.077	0.032
59	20	100	100%	0.012	0.055	0.020
60	20	250	20%	0.019	0.076	0.030
61	20	250	40%	0.020	0.080	0.032
62	20	250	60%	0.018	0.068	0.029
63	20	250	80%	0.021	0.075	0.032
64	20	250	100%	0.017	0.063	0.027
65	20	500	20%	0.020	0.077	0.031
66	20	500	40%	0.019	0.076	0.030
67	20	500	60%	0.017	0.065	0.027
68	20	500	80%	0.019	0.073	0.030
69	20	500	100%	0.018	0.068	0.028
70	20	1000	20%	0.020	0.076	0.032
71	20	1000	40%	0.019	0.077	0.031
72	20	1000	60%	0.018	0.069	0.029
73	20	1000	80%	0.018	0.072	0.029
74	20	1000	100%	0.017	0.069	0.027

K.2 Preheat 3 unit Random Forest parameter optimization

	Depth	Number of trees	Sample size	Precision	Recall	F1
0	5	50	20%	0.013	0.479	0.025
1	5	50	40%	0.013	0.457	0.025
2	5	50	60%	0.013	0.460	0.026
3	5	50	80%	0.013	0.466	0.026
4	5	50	100%	0.013	0.456	0.026
5	5	100	20%	0.013	0.476	0.025
6	5	100	40%	0.013	0.465	0.025
7	5	100	60%	0.013	0.462	0.025
8	5	100	80%	0.013	0.464	0.025
9	5	100	100%	0.013	0.456	0.026
10	5	250	20%	0.012	0.464	0.024
11	5	250	40%	0.013	0.461	0.025
12	5	250	60%	0.013	0.458	0.025
13	5	250	80%	0.013	0.458	0.025
14	5	250	100%	0.013	0.456	0.025
15	5	500	20%	0.012	0.460	0.024
16	5	500	40%	0.013	0.466	0.025
17	5	500	60%	0.013	0.462	0.024
18	5	500	80%	0.013	0.466	0.025
19	5	500	100%	0.013	0.467	0.025
20	5	1000	20%	0.012	0.458	0.024
21	5	1000	40%	0.012	0.461	0.024
22	5	1000	60%	0.013	0.469	0.025
23	5	1000	80%	0.013	0.463	0.024
24	5	1000	100%	0.013	0.469	0.025
25	10	50	20%	0.012	0.250	0.022
26	10	50	40%	0.012	0.237	0.022
27	10	50	60%	0.011	0.230	0.021
28	10	50	80%	0.013	0.261	0.024
29	10	50	100%	0.012	0.250	0.024
30	10	100	20%	0.011	0.235	0.022
31	10	100	40%	0.012	0.227	0.023
32	10	100	60%	0.012	0.234	0.023
33	10	100	80%	0.012	0.234	0.024
34	10	100	100%	0.013	0.233	0.024
35	10	250	20%	0.011	0.231	0.022
36	10	250	40%	0.013	0.236	0.025
37	10	250	60%	0.012	0.223	0.022
38	10	250	80%	0.012	0.212	0.022
39	10	250	100%	0.012	0.208	0.022

40	10	500	20%	0.012	0.244	0.023
41	10	500	40%	0.012	0.228	0.023
42	10	500	60%	0.012	0.228	0.023
43	10	500	80%	0.012	0.219	0.022
44	10	500	100%	0.012	0.212	0.022
45	10	1000	20%	0.012	0.235	0.022
46	10	1000	40%	0.011	0.216	0.021
47	10	1000	60%	0.012	0.219	0.022
48	10	1000	80%	0.012	0.223	0.023
49	10	1000	100%	0.011	0.216	0.022
50	20	50	20%	0.011	0.027	0.016
51	20	50	40%	0.109	0.030	0.047
52	20	50	60%	0.021	0.027	0.023
53	20	50	80%	0.038	0.035	0.036
54	20	50	100%	0.017	0.032	0.022
55	20	100	20%	0.025	0.033	0.028
56	20	100	40%	0.035	0.026	0.030
57	20	100	60%	0.008	0.018	0.011
58	20	100	80%	0.044	0.030	0.035
59	20	100	100%	0.027	0.030	0.028
60	20	250	20%	0.059	0.026	0.036
61	20	250	40%	0.043	0.024	0.031
62	20	250	60%	0.061	0.024	0.035
63	20	250	80%	0.064	0.027	0.038
64	20	250	100%	0.049	0.030	0.037
65	20	500	20%	0.047	0.031	0.037
66	20	500	40%	0.047	0.031	0.037
67	20	500	60%	0.048	0.031	0.038
68	20	500	80%	0.049	0.027	0.035
69	20	500	100%	0.050	0.030	0.037
70	20	1000	20%	0.063	0.031	0.041
71	20	1000	40%	0.047	0.027	0.034
72	20	1000	60%	0.049	0.031	0.038
73	20	1000	80%	0.065	0.027	0.038
74	20	1000	100%	0.116	0.031	0.049

K.3 Printhead Seneca unit Random Forest parameter optimization

	Depth	Number of trees	Sample size	Precision	Recall	F1
0	5	50	20%	0.026	0.525	0.050
1	5	50	40%	0.031	0.523	0.059
2	5	50	60%	0.026	0.542	0.050
3	5	50	80%	0.041	0.515	0.076
4	5	50	100%	0.041	0.524	0.076
5	5	100	20%	0.072	0.520	0.127
6	5	100	40%	0.054	0.528	0.098
7	5	100	60%	0.056	0.542	0.102
8	5	100	80%	0.054	0.554	0.099
9	5	100	100%	0.046	0.565	0.086
10	5	250	20%	0.064	0.510	0.114
11	5	250	40%	0.068	0.513	0.121
12	5	250	60%	0.046	0.542	0.085
13	5	250	80%	0.053	0.525	0.097
14	5	250	100%	0.044	0.542	0.082
15	5	500	20%	0.072	0.535	0.127
16	5	500	40%	0.065	0.523	0.116
17	5	500	60%	0.065	0.544	0.116
18	5	500	80%	0.050	0.541	0.092
19	5	500	100%	0.046	0.545	0.085
20	5	1000	20%	0.075	0.535	0.132
21	5	1000	40%	0.071	0.530	0.125
22	5	1000	60%	0.074	0.541	0.131
23	5	1000	80%	0.076	0.541	0.133
24	5	1000	100%	0.075	0.544	0.132
25	10	50	20%	0.033	0.310	0.060
26	10	50	40%	0.032	0.318	0.058
27	10	50	60%	0.030	0.296	0.054
28	10	50	80%	0.043	0.299	0.074
29	10	50	100%	0.041	0.292	0.071
30	10	100	20%	0.047	0.287	0.080
31	10	100	40%	0.039	0.311	0.070
32	10	100	60%	0.060	0.282	0.100
33	10	100	80%	0.052	0.297	0.088
34	10	100	100%	0.059	0.294	0.098
35	10	250	20%	0.059	0.286	0.097
36	10	250	40%	0.037	0.287	0.065
37	10	250	60%	0.058	0.292	0.097
38	10	250	80%	0.052	0.290	0.089
39	10	250	100%	0.052	0.293	0.089

40	10	500	20%	0.056	0.289	0.093
41	10	500	40%	0.044	0.286	0.077
42	10	500	60%	0.069	0.285	0.111
43	10	500	80%	0.052	0.282	0.087
44	10	500	100%	0.058	0.280	0.097
45	10	1000	20%	0.052	0.287	0.088
46	10	1000	40%	0.045	0.276	0.078
47	10	1000	60%	0.066	0.277	0.107
48	10	1000	80%	0.053	0.280	0.090
49	10	1000	100%	0.060	0.275	0.099
50	20	50	20%	0.068	0.039	0.050
51	20	50	40%	0.110	0.042	0.061
52	20	50	60%	0.112	0.055	0.074
53	20	50	80%	0.112	0.058	0.076
54	20	50	100%	0.118	0.048	0.068
55	20	100	20%	0.110	0.039	0.058
56	20	100	40%	0.111	0.041	0.060
57	20	100	60%	0.111	0.042	0.061
58	20	100	80%	0.111	0.039	0.058
59	20	100	100%	0.112	0.051	0.070
60	20	250	20%	0.111	0.041	0.060
61	20	250	40%	0.111	0.038	0.057
62	20	250	60%	0.111	0.041	0.060
63	20	250	80%	0.112	0.044	0.063
64	20	250	100%	0.111	0.044	0.063
65	20	500	20%	0.111	0.042	0.061
66	20	500	40%	0.111	0.041	0.060
67	20	500	60%	0.111	0.042	0.061
68	20	500	80%	0.111	0.042	0.061
69	20	500	100%	0.112	0.039	0.058
70	20	1000	20%	0.011	0.037	0.017
71	20	1000	40%	0.111	0.041	0.060
72	20	1000	60%	0.011	0.035	0.017
73	20	1000	80%	0.111	0.039	0.058
74	20	1000	100%	0.111	0.039	0.057