Eindhoven University of Technology

MASTER

Analyzing machine data for predictive maintenance of electro chemical machining electrodes

Wang, F.

Award date:
2016

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Analyzing Machine Data for Predictive Maintenance of Electro Chemical Machining Electrodes

Fengjun Wang

Supervisors:
prof. dr. Mykola Pechenizkiy TU/e
ir. Kees Wouters Philips Research
dr. Qi Gao Philips Research

Assessment committee:
prof. dr. Mykola Pechenizkiy TU/e
ir. Kees Wouters Philips Research
dr.ir. Marc Geilen TU/e
dr. Alp Akçay TU/e

Public Version
Abstract

This Master Thesis Project is an experimental project that applies machine learning approaches to analyze machine data and explore predictive maintenance questions in Philips Shaver Production Plant use case. The use case is in the context of MANTIS - Cyber Physical System based Proactive Collaborative Maintenance. As the MANTIS project book (Mondragon Goi Eskola Politeknikoa S. Coop, 2014) mentions, “The overall concept of MANTIS is to provide a predictive maintenance service platform architecture that allows to estimate future performance, to predict imminent failures and to schedule pro-active maintenance and that consists of distributed processing chains that efficiently transform raw data into knowledge while minimizing the need for bandwidth”. In the MANTIS project book, this processing chain can include topics on local data collecting from smart sensors or events logs, distributed models on data validation and decision-making with machine learning methodologies, as well as cloud-based processing and data availability.

In our use case, Philips Consumer Lifestyle Drachten (location of the shaver factory) takes in charge of collecting and storing data from one pilot production line and the maintenance workshop; the Master Thesis Project focuses on machine learning for data validation and decision-making. In this project, we create maintenance piece (MP) split methods on the original data source and define MP classes by maintenance type classification tests; we give solutions on the collaboration drift problem of the original data from different machines; we apply different smoothing methods on the noisy data and check the performance improvements on prediction models; our classification models on impending maintenance actions show promising results to forecast some maintenance actions on dataset subsets; our regression models on the tool remaining useful life estimation are limited by the ground truth problem of response variables (the tool status).
## Glossary

Table 1: Main abbreviations/terms in this Thesis

<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECM</td>
<td>Electro Chemical Machining</td>
</tr>
<tr>
<td>TIP</td>
<td>the top part of the electrode</td>
</tr>
<tr>
<td>HOUDER</td>
<td>one part of the electrode that can hold TIP</td>
</tr>
<tr>
<td>$C_{1}$</td>
<td>the number of caps made by the tool from the last inspection</td>
</tr>
<tr>
<td>$C_{2}$</td>
<td>the total number of caps made by the TIP, till current product</td>
</tr>
<tr>
<td>$L$</td>
<td>the life length limit - the maximum number of caps each TIP can make</td>
</tr>
<tr>
<td>$R$</td>
<td>regular inspection interval of tool</td>
</tr>
<tr>
<td>$R_{Q}$</td>
<td>quality inspection interval of products</td>
</tr>
<tr>
<td>Life Length</td>
<td>the total number of products (caps) one TIP made</td>
</tr>
<tr>
<td>RC</td>
<td>Rest Caps one TIP can make (from current product)</td>
</tr>
<tr>
<td>RUL</td>
<td>as a response/target variable: Rest life length (defined based on the total number of caps one TIP make) ratio one TIP has (from current product)</td>
</tr>
<tr>
<td>RUL</td>
<td>as a topic: Remaining Useful Life</td>
</tr>
<tr>
<td>HHT</td>
<td>Hilbert Huang transform</td>
</tr>
<tr>
<td>MV</td>
<td>Moving Average</td>
</tr>
<tr>
<td>process running</td>
<td>all running phases without machine warm-up phase</td>
</tr>
<tr>
<td>ERROR_ACCUM</td>
<td>counts the number of key errors each TIP experienced till current product</td>
</tr>
<tr>
<td>ERROR_ACCUM_ONLYPROCESS</td>
<td>only counts the number of key errors each TIP experienced during 'process running' phases</td>
</tr>
<tr>
<td>CHANGE</td>
<td>the maintenance action (take tool out, do maintenance operations, and then put the tool back into one free machine randomly)</td>
</tr>
<tr>
<td>NO_MAINTENANCE</td>
<td>how many times one TIP has been sent to the maintenance center till current product</td>
</tr>
<tr>
<td>MP</td>
<td>Maintenance Piece. For each TIP, each maintenance piece is the records between two maintenance inspections.</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
</tr>
<tr>
<td>AUC</td>
<td>Area Under Curve (ROC curve in this thesis)</td>
</tr>
</tbody>
</table>
Contents

1 Introduction .......................................................... 1
   1.1 Background and Motivation .................................... 1
   1.2 Thesis Objective and Main Implementations ................. 1
   1.3 Thesis Structure ............................................... 2

2 Problem statement .................................................. 3
   2.1 Data Sources .................................................. 3
      2.1.1 Data Logs ................................................. 3
      2.1.2 Maintenance Logs ...................................... 5
   2.2 Research Problem ............................................. 7
      2.2.1 Challenges for Research Problems ....................... 7
      2.2.2 Main Research Problem - Distinguishing Maintenance Types . 7
      2.2.3 Additional Research Problem - TIPs’ RC/RUL estimation ... 8

3 Literature Survey .................................................... 9
   3.1 Predictive Maintenance ........................................ 9
      3.1.1 Overview of RUL estimation approaches ................ 9
      3.1.2 Data Description in Time Series ......................... 9
      3.1.3 Prediction with Classification Method .................. 10
      3.1.4 Prediction with Pattern Mining Method .................. 10
   3.2 Relevant Methodologies for our Research Questions ......... 11
      3.2.1 Artificial Neural Network ............................... 11
      3.2.2 Random Forest - Overview and Comparison with Other Algorithms 11

4 Development Approach on Data Preparation ...................... 14
   4.1 Data Pre-processing .......................................... 14
      4.1.1 Data Logs Format Unification ......................... 14
      4.1.2 Extract TIP Identification from Maintenance Logs ....... 14
      4.1.3 Label Replacement Points and Correct REV in Data Logs .... 15
   4.2 Split Data into Maintenance Pieces ........................ 16
   4.3 Filling Missing Values ........................................ 17

5 Feature Observation and Extraction ............................. 19
   5.1 Process Error Analysis ...................................... 19
      5.1.1 Key Errors Occurrences Density ........................ 19
      5.1.2 Key Errors Accumulation Slope/Trends .................. 20
      5.1.3 Process Attributes when Error Occurs. ................. 23
   5.2 Process Feature Trends Extraction and Observation ......... 24
      5.2.1 Feature Trends Extraction Based on Hilbert-Huang Transform ... 25
      5.2.2 Feature Trends Extraction with Moving Average ........... 29
      5.2.3 Trends Observation on Process Attributes ............... 29
   5.3 Feature Extraction Results .................................. 29
List of Figures

2.1 Shaver cap .................................................. 4
2.2 Electrode, product and support .................................. 4
2.3 Standard lines for quality parameters .............................. 5
2.4 Information in maintenance logs .................................. 6
3.1 Time series data representation .................................. 10
3.2 Single node behavior of ANN .................................. 11
4.1 A record (instance) before format unification .................... 15
4.2 A record (instance) after format unification .................... 15
4.3 Information in relevant maintenance log .......................... 15
4.4 Life length distribution ....................................... 16
4.5 An example: T142101 X ....................................... 17
5.1 Process key errors occurrences/frequencies and Life length .... 20
5.2 Process key errors trends of class 8 (see each subgraph/TIP’s last maintenance piece) 21
5.3 Process key errors trends of class 3 (see each subgraph/TIP’s last maintenance piece) 22
5.4 T142101 X_9, original ......................................... 23
5.5 T142101 X_10, original ........................................ 24
5.6 T142101 X_10, after remove outliers ............................. 24
5.7 T142116 X_1, applying HHT ................................... 26
5.8 T142116 X_9, applying HHT ................................... 27
5.9 T142112 X_1, with trend extraction ............................. 28
5.10 Some examples show level drops/increases ....................... 30
5.11 Two examples on feature offsets ................................ 31
5.12 T142112 X_1 moving derivative ( W S_{dev} = 500) ............... 32
5.13 T142112 X_1 ................................................. 33
5.14 Explanation on main process attributes .......................... 34
5.15 Jumps inside one Maintenance Piece ............................ 34
6.1 A sketch of the experiment design ................................ 39
6.2 Confusion matrix for classifier evaluation ........................ 40
6.3 F1-score on all interesting groups ................................ 43
6.4 Results on class 3 and class 4 ................................... 43
6.5 Results of groups {3, 8}, {4, 8}, {3, 4} ............................ 44
6.6 Results of group {1, 3} ......................................... 44
6.7 Results (average of the 72 experiments) on {3, 8}, with different Feature Subsets .... 45
6.8 Results on {3, 8}, with different Feature Subsets ................ 46
6.9 Results with different trend extractions on predictors, of group {3, 8} .................. 47
6.10 ROC - compare trend extraction influence on group {3, 8}, pick tail size = 4000 .... 47
6.11 AUC and F1 score - Compare different ‘Pick Tail’ Strategies ........ 48
6.12 Results with full pieces ........................................ 49
LIST OF FIGURES

6.13 Results on \{N, E\}, with feature subset = a, \( n_{\text{estimators}} = 20 \), pick tail size = 3000, \( \text{min}_\text{samples}_{\text{split}} = 30 \), \( \text{min}_\text{weighted}_\text{frac}_{\text{leaf}} = 0 \) ........................................ 52
6.14 Results on \{N, E\}, with feature subset = b, \( n_{\text{estimators}} = 20 \), pick tail size = 3000, \( \text{min}_\text{samples}_{\text{split}} = 30 \), \( \text{min}_\text{weighted}_\text{frac}_{\text{leaf}} = 0 \) ........................................ 53
6.15 Results on \{N, E\}, with feature subset = c, \( n_{\text{estimators}} = 20 \), pick tail size = 3000, \( \text{min}_\text{samples}_{\text{split}} = 30 \), \( \text{min}_\text{weighted}_\text{frac}_{\text{leaf}} = 0 \) ........................................ 54
6.16 Results on \{N, E\}, with feature subset = d, \( n_{\text{estimators}} = 20 \), pick tail size = 3000, \( \text{min}_\text{samples}_{\text{split}} = 30 \), \( \text{min}_\text{weighted}_\text{frac}_{\text{leaf}} = 0 \) ........................................ 55
A.1 Results on \{3, 8\}, with feature subset = a, \( n_{\text{estimators}} = 10 \), pick tail size = 5000, \( \text{min}_\text{samples}_{\text{split}} = 10 \), \( \text{min}_\text{weighted}_\text{frac}_{\text{leaf}} = 0 \) ................................. 62
A.2 Results on \{3, 8\}, with feature subset = b, \( n_{\text{estimators}} = 10 \), pick tail size = 5000, \( \text{min}_\text{samples}_{\text{split}} = 20 \), \( \text{min}_\text{weighted}_\text{frac}_{\text{leaf}} = 0 \) ................................. 63
A.3 Results on \{3, 8\}, with feature subset = c, \( n_{\text{estimators}} = 10 \), pick tail size = 5000, \( \text{min}_\text{samples}_{\text{split}} = 20 \), \( \text{min}_\text{weighted}_\text{frac}_{\text{leaf}} = 0 \) ................................. 63
A.4 Results on \{3, 8\}, with feature subset = d, \( n_{\text{estimators}} = 10 \), pick tail size = 5000, \( \text{min}_\text{samples}_{\text{split}} = 10 \), \( \text{min}_\text{weighted}_\text{frac}_{\text{leaf}} = 0 \) ................................. 63
A.5 Results on \{3, 8\}, with feature subset = e, \( n_{\text{estimators}} = 3 \), pick tail size = 4000, \( \text{min}_\text{samples}_{\text{split}} = 5 \), \( \text{min}_\text{weighted}_\text{frac}_{\text{leaf}} = 0 \) ................................. 64
## List of Tables

<table>
<thead>
<tr>
<th></th>
<th>Table Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Main abbreviations/terms in this Thesis</td>
<td>iii</td>
</tr>
<tr>
<td>6.1</td>
<td>Maintenance codes</td>
<td>37</td>
</tr>
<tr>
<td>6.2</td>
<td>MP classes</td>
<td>37</td>
</tr>
<tr>
<td>6.3</td>
<td>New maintenance pieces classes define - combined results</td>
<td>50</td>
</tr>
<tr>
<td>6.4</td>
<td>Classification results on distinguishing {N, E}</td>
<td>50</td>
</tr>
<tr>
<td>6.5</td>
<td>Classification results on distinguishing {N, M, E}</td>
<td>51</td>
</tr>
<tr>
<td>A.1</td>
<td>Classification results on interesting groups {1, 3}, {1, 4}, {1, 8}, {1, 10}, {3, 10}, {4, 10}</td>
<td>65</td>
</tr>
<tr>
<td>A.2</td>
<td>Classification results on interesting group {3, 4}</td>
<td>65</td>
</tr>
<tr>
<td>A.3</td>
<td>Classification results on interesting group {3, 8}</td>
<td>66</td>
</tr>
<tr>
<td>A.4</td>
<td>Classification results on interesting group {4, 8}</td>
<td>66</td>
</tr>
<tr>
<td>A.5</td>
<td>Classification results on interesting group {1, 3, 4}</td>
<td>66</td>
</tr>
<tr>
<td>A.6</td>
<td>Classification results on interesting group {3, 8}, with/without moving average</td>
<td>67</td>
</tr>
<tr>
<td>A.7</td>
<td>Top classification results on interesting group {3, 8}, when choose different predictors</td>
<td>68</td>
</tr>
<tr>
<td>A.8</td>
<td>Classification results on interesting group {1, 3}, apply sampling to make balance dataset</td>
<td>68</td>
</tr>
<tr>
<td>A.9</td>
<td>Classification results on interesting group {3, 8}, when picking different tail size</td>
<td>69</td>
</tr>
<tr>
<td>A.10</td>
<td>Classification results of experiments in Section 6.4.2 on interesting group {3, 8}</td>
<td>70</td>
</tr>
<tr>
<td>A.11</td>
<td>Classification Results of Experiments in Section 6.4.2 on Interesting Group {3, 8}</td>
<td>71</td>
</tr>
<tr>
<td>A.12</td>
<td>Classification Results of Experiments in Section 6.4.2 on Interesting Group {3, 8}</td>
<td>72</td>
</tr>
<tr>
<td>A.13</td>
<td>Classification Results of Experiments in Section 6.4.2 on Interesting Group {3, 8}</td>
<td>73</td>
</tr>
<tr>
<td>A.14</td>
<td>Classification Results of Experiments in Section 6.4.2 on Interesting Group {3, 8}</td>
<td>74</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background and Motivation

Philips factory uses Electro Chemical Machining (ECM) as a finishing technology for the mass production of shaving heads (also called caps, shown in Figure 2.1). The result is a copy/projection of the negatively charged electrode in the positively charged shaving heads. The electrodes for the ECM process are very precise and expensive. Additionally, the electrodes are very delicate and are easily damaged. Furthermore, the ECM process itself is a highly complex and very accurate manufacturing process where the room for error is relatively small. The slightest damage to a tool results in poor quality of products. Since the electrodes are expensive, any potential increase in the lifetime could be an exciting improvement.

Maintenance operations are significant to apply proper repairs on electrodes and guarantee the product quality. The current tool maintenance system is largely based on reactive maintenance and regular inspection. The reactive maintenance is mostly performed after a damage of the tool, a disruption of the machine or detections of unqualified products. The regular inspection on the tool is performed every fixed number of caps produced, and the service period (regular interval) is around \( R \) caps from the last inspection. The regular inspection on product quality is performed around every \( R_{Q} \) hours. For each batch of caps, quality inspectors measure products samples' quality (geometrical attributes of products) on off-line machines.

Philips factory expects to use collected production line data to make predictions about product quality, process disruptions, and impending maintenance actions. Some researchers are in charge of the product quality prediction part. The final common target for all projects is changing from reactive to predictive maintenance by optimizing or removing the regular inspection cycle, maximizing tool life, minimizing downtime due to change-overs, optimizing the level of tools and subparts in stock, and finally expect to make maintenance plan available to operators.

1.2 Thesis Objective and Main Implementations

The Master Thesis Project is part of the start exploration for the predictive maintenance system. The target is to use advanced machine learning algorithms to make predictions on maintenance actions of delicate electrodes (TIPs) and to estimate the Remaining Useful Life (RUL) for each TIP in use. We expect to predict impending maintenance actions, especially the replacement of tools as soon as possible, to avoid costs on unqualified products that made by damaged tools. As the factory requires, the prediction models should have a prediction horizon of at least 8 hours to schedule actions in the upcoming shifts.

In this thesis, we choose different methods (include Hilbert-Huang Transform, Moving Average) to extract process features' trends; to realize the prediction on maintenance actions, we build classification models with the random forest algorithm to distinguish different maintenance types;
and we build regression models with the random forest algorithm to do the RUL estimation on TIPs.

1.3 Thesis Structure

Chapter 2 introduces the original data source in detail and analyzes our research problems as well as challenges. Chapter 3 includes two parts: the first part represents an overview of our literature survey on the predictive maintenance topic, and the second part explains methods we choose for our research problems. Chapter 4 includes the data pre-processing, in which the main part is the maintenance pieces split in Section 4.2. Chapter 5 shows the data visualization on original features and processed features, as well as feature extraction. Chapter 6 describes the maintenance piece classifications, and designed controlled experiments (e.g., comparing different strategies on the predictor variables representation, comparing different model parameter settings). The results in Chapter 6 describe the most significant contributions of the thesis. Chapter 7 shows the RUL estimation model results and analysis. Chapter 8 concludes our project and gives recommendations and future work. Appendices contain detailed results of implementations and some examples of attributes distributions.
Chapter 2

Problem statement

In Section 2.1, we describe the data sources we have - data logs and maintenance logs. In Section 2.1.1, we state the specific research problems as well as the challenges throughout this project.

2.1 Data Sources

Philips factory provides maintenance logs, and gathers large amounts of data from one ECM pilot production line: process control platform and sensors in machine (process data), products (product_id, quality data), the electrodes (tooling data), the materials (material data) and the electrolyte (central electrolyte facility data). For each produced cap, we register all the above attributes (from several sources) during its production as one instance (data log/record) on the database. The data is currently only used for manual, after-the-fact analysis process disruptions and not all data is centrally accessible and synchronized across the different platforms. As a consequence, this project only applies machine learning algorithms on summarized data logs, and successful models can be applied to real-time data streams in the future.

2.1.1 Data Logs

The number of currently accessed records in the data logs is 4.56 million, and each record consists of data for a unique product. Since some measurements are only implemented on product samples, not all records are complete; a complete record contains 131 original features from machine_id, product_id, process data, tooling data, material data, quality data, and electrolyte data, etc.

The current reactive maintenance strategy focuses on 3 subparts (shown in Figure 2.2) of the tool: TIP, HOUDER, and flushing house. The TIP is the most delicate part and is the object that we concern in this project. Each TIP can be installed on one HOUDER, and the HOUDER-TIP pair can be installed inside a flushing house. Each TIP, HOUDER, or flushing house has unique identifiers (ID). All data logs are summarized from the pilot production line which runs 8 ideally same machines and produces one kind of shaving head. Thus, the HOUDER, TIP, and flushing house are interchangeable during maintenance operations. The lifetime of the flushing house is longer than HOUDER, and the lifetime of HOUDER is longer than TIP, so there are some cases that when a TIP is damaged, the maintenance operators discard the TIP, keep the healthy HOUDER, install a new TIP on it, put the new HOUDER-TIP in a flushing house (maybe it is the HOUDER’s former flushing house or another flushing house), and then send the tooling back to the production line. When a replacement occurs, the TIP ID changes so we can distinguish the new TIP from the old one.

The main data sources are listed as follows, and a data log record example can be found in Table 4.1.

- General data includes product_id, time-stamp, and machine label.
• Electrolyte data describe parameters on temperature, composition, conductivity, and contamination on the electrolyte that used for ECM process.

• Tooling data describe tooling run length (\( C_1 \) - the number of caps made by the tool from the last inspection, and \( C_2 \) - the total number of caps made by the tool), and tooling geometry information. The geometry information is measured once upfront before each new tooling is put on the production line.

• Material data contains sample data of the material used for products. The material measurements take 1% samples from each batch.

• Process data are collected from process measurements and sensors. It contains information like process force, process motion, process speed, etc. A complete ECM process includes Shaving Side Profile ECM-ing (PSV), Tooth Profile ECM-ing (TPE), Slotwidth Profile ECM-ing (GPE), and Gloss and Deburring (GAO). This project focuses on the electrodes applied in GPE part, so we only need the GPE process data.

• Quality data are recorded by the measuring devices after measuring geometry parameters of caps. For each quality attribute, we have five standard lines as Figure 2.3 shows: lower specification limit (LSL), lower control limit (LCL), target (TGT), upper control limit (UCL), and upper specification limit (USL). If the quality is outside limits (below LSL or above USL), then the product is considered as unqualified and maintenance operators perform tooling inspection. The LCL and UCL are control limits which are used as thresholds on tuning process control attributes.
2.1.2 Maintenance Logs

Maintenance Activities

In each maintenance inspection, the maintenance operators can do a quick repair, clean, renew profile (need external reworks on the TIP), or discard a tool since the tool is unrepairable damaged or reach maximum caps limit or other reasons. In the current maintenance system, we have four kinds of triggers for a maintenance inspection: reaching regular inspection interval, unqualified products, process errors related to the electrode and special cases like a holiday, in which the whole production line is shut down for planned maintenance. We describe the current maintenance inspections as follows:

- **Regular inspection:** after the last maintenance inspection + R (regular inspection interval) caps

  In practice, one TIP can produce L caps within quality specifications. Thus, in a maintenance inspection, experts first check if the TIP has produced more than L-R caps; if so, the operators still check the tool but the TIP is discarded anyway by the maintenance center, even it might still healthy.

  This maintenance strategy leads to one problem in our project: there is no ground truth of each TIP’s rest caps (RC). In current maintenance system, we set a threshold L to limit the maximum number of caps each TIP can produce. Thus, many TIPs are not fully utilized till the discarding time so we do not have a reasonable value that reflects each TIP real life.

- **Unplanned inspection:** products outside of quality specifications

  As we described in Section 2.1.1, the factory has 5 product quality standards thresholds: LSL, LCL, TGT, UCL, and USL. If one of the quality measures (mostly referred to as Critical to Quality, CTQ) is outside the LCL and UCL, the operators must take action. This action is called Out-of-control Action Plan (OCAP), and may involve changing of tools. This depends on the CTQ. A change of tools always leads to inspection/maintenance of the tool. If the CTQ is outside LSL/USL, then the same applies. However, in these cases, all products produced must be blocked, and are most of the time scrapped.

  In conclusion, unqualified products in CTQ triggers OCAP actions. The OCAP actions may trigger maintenance inspections on the tool. Maintenance inspections may involve discarding of tools. Normally, the CTQ is conducted once every R.Q hours, so the quality data has a high missing rate in original data logs.

- **Unplanned inspection:** process errors/machine disruptions

  The data logs include many different kinds of process error codes. Not all errors are related to the tools, and not all errors cause machine disruptions. The error code is simply the error that was given when each product was produced. The code is not used in any kind of feedback loop or tuning. In cases of process/machine errors, the operator will decide on
whether or not he will start Technical Out-of-control action plan (TOCAP) actions, which may include maintenance inspection on the tool and then tool discarding or return.

Maintenance Log Contents

Maintenance logs record all maintenance activities and response times. Each maintenance log includes some operations codes (some codes are listed in Table 6.1) on one HOUDER-TIP pair.
CHAPTER 2. PROBLEM STATEMENT

2.2 Research Problem

In this section, we first explain three challenges throughout this project, and then describe the 2 research topics that we focus on. We also give reference sections which solve the problems or deal with the challenges.

2.2.1 Challenges for Research Problems

There are 3 challenges in our project:

**Tool Status definition is linearly and based on production counter**

Experience of production indicates there are relationships between the tooling degradation, process behavior, and product quality behavior. Thus, one meaningful point for this project is modeling the relationships and building tool status prediction rules. However, we do not have the tool degradation data. Tool physical deterioration can only be measured with optical instruments by maintenance experts. Since the measurements are off-line and time-consuming, we do not have the tracking data of the TIPs degradation. Normally, experts only measure each TIPs initial physical parameters (tooling data) once before the first time it is installed on the production line. Maintenance engineers take off-line inspections but just record the operations, not the tool geometric parameters; once damage is observed, repair operations are carried out, or the tool is discarded if the damage is too severe.

**Process data from different machines are not calibrated**

When one tool needs maintenance inspections, operators do tooling CHANGE: take the tool out from the machine and send it to the maintenance department; if the tool is healthy or could be repaired, we put it back into one free machine randomly. We know that different machines (ideally same) could have different properties. Even for same machine, it could have component repairs or replacements. Thus, process features are not absolutely calibrated among different maintenance pieces (we define the maintenance pieces as the data logs of each TIP between 2 maintenance actions). In Section 5.2.2, we use feature offsets to represent process features’ characters.

**The sensory data (e.g., process parameters) are non-stationary**

The process signals are not periodic and are hard to apply frequency analysis on trends extraction. Section 4.2, Figure 4.5, Figure 5.6, Figure 5.7, and Figure 8.2 show some examples of several process parameters. Section 5.2 show methodologies we apply on trends extraction.

2.2.2 Main Research Problem - Distinguishing Maintenance Types

The maintenance logs record all maintenance types (including maintenance triggers and actions) on tools, and the data logs record features (from machine sensor, quality test, etc.) of each product as time passes. Based on this information, we plan to extract relations between maintenance types and feature trends and define the research problem as a classification problem:

- Classification model building: in the terminology of machine learning [1], classification is considered as an instance of supervised learning. More precisely, classification aims to identify a new observation’s category (class), on the basis of a training set of data containing observations/instances whose category membership is known.

We have the following research questions: extract trend information and features from the original attributes and apply them as predictor variables for our classification model; define several classes that reflect different kinds of maintenance types; build classification models to distinguish these classes. With success classifiers, we can distinguish normal running and abnormal running processes, and then give suggestions on following maintenance actions.
beforehand to minimize downtime of the production machine for tool changeovers. Chapter 5 includes feature extraction for the classification problem, and Chapter 6 includes the classification models building and experiments results.

2.2.3 Additional Research Problem - TIPs’ RC/RUL estimation

The second research problem is implementing a data-driven proactive maintenance strategy by building a prediction model of the replacement/maintenance time for each electrode TIP in use. Since the tooling degradation tracing is unrealistic as we discussed in Section 2.2.1, we define the tool status based on the number of caps each TIP has produced (feature “C_2”). Since the production rate is stable, we plan to estimate the reasonable replacement/maintenance time by predicting the RC (the number of rest caps each TIP can make in the future) or the RUL (Remaining Useful Life percent of each TIP). Chapter 7 shows the implementations on this research question. We build regression model in vector space: for each time-series attribute (from data logs of each product in time order), we apply some methods like moving average, moving derivative to represent the original attribute as features (predictors) of samples.

- Regression model building: regression analysis is generally applied to estimate continuous response variables, as opposed to the discrete target variables used in classification problems [2]. In our project, with RC/RUL values as response/target variables, we can apply regression algorithms to predict the specific value of each TIP’s rest life.
Chapter 3

Literature Survey

This chapter includes 2 parts: the Section 3.1 shows a literature survey on the general predictive maintenance background; the Section 3.2 includes the explanation and comparison of candidate algorithms that we choose for solving the research questions.

3.1 Predictive Maintenance

In this section, we first give an overview of RUL estimation approaches, and then describes some current popular methods on predictive maintenance topic in Section 3.1.2, Section 3.1.3, and Section 3.1.4.

3.1.1 Overview of RUL estimation approaches

The RUL depends on the current age of the asset, the operation environment and the observed condition monitoring (CM) or health information. Based on the review of existing literature, statistical data-driven approaches can be classified by the observed CM data into direct CM and indirect CM. Direct CM data is the data which can describe the underlying state (e.g., wear and crack sizes) of the system directly so that the RUL estimation is the prediction of the CM data to reach a predefined threshold level. Indirect CM data is the data which can only indirectly or partially indicate the underlying state of the system so failure event data may be needed in addition to CM data for an RUL estimation purpose[3].

For our use case, we only have indirect data, since there is no tracking geometrical data on the tool wear out degree. The indirect data we have are process data, tooling data, material data, and quality data. As a result, we do a literature survey on indirectly observed data based models. These kind of models are also called as the partially observed state process models since there is a stochastic relationship between the observed CM processes and the unobservable state. There are several types of models that can be used on RUL prediction with indirect data, such as stochastic filtering-based models, covariate based hazard models. A typical covariate based hazard model is Proportional Hazards Modelling (PHM), which was first proposed by Cox[4] and models the way explanatory or concomitant variables, also referred to as covariances, affect the life of the asset.

Except for traditional statistical analysis, with the development of machine learning, we have more effective and widely used algorithms to solve our research problem by building classification/regression models. In Section 3.2, we focus on several machine learning methodologies that we implement or compare in our project.

3.1.2 Data Description in Time Series

Based on the review of existing literature, data used for predictive maintenance are usually gathered from smart sensors, and operation logs. Failure events can be extracted from operation logs, e.g., a replacement of one component indicates a failure. In some cases, it is hard to
label all failures just according to operation logs. In this case some researchers apply unsupervised learning and dig out more failures type before training the machine learning model.

For time-series data, sliding window is generally used to process consecutive short time records of the investigated system. As Figure 3.1 [5] shows, each box (instance) represents a piece of data in a time interval. In some use cases (e.g., in paper [6]), before a failure, the machine usually works irregularly, and switches between normal/abnormal (negative/positive). Considering this kind of fact, some scientists propose multi-instance methods. In multi-instance learning, the learner receives several bags that with positive or negative labels. One bag wrap several continuous instances inside; once there exists a positive window, then the whole bag is labeled as positive. [7] From a log-based predictive maintenance that Wang et al. (2014) [6] applied, they do window-based prediction and if the prediction score of a bag is beyond a given threshold, then it is positive. When training models, Wang et al. (2014) [6] use the mean value of a positive bag as a positive learning example, which aims to reduce influences by noisy points.

![Figure 3.1: Time series data representation](image)

3.1.3 Prediction with Classification Method

Many researchers propose prediction methods based on the multi-instance representation that we introduced in Section 3.1.2. The Multi-instance (MI) support vector machine (SVM) algorithms family is a typical example that has excellent performance on anomaly detection and prediction. The baseline algorithm in this family is MI-SVM[8], which modifies the standard SVM formulation to fit the constraints on each bag. Following that, sparse Multi-instance learning algorithms (sMIL, stMIL, and sbMIL) are proposed [9]; they intentionally bias SVM formulations to handle the assumption that there are very few positive instances in each positive bag. In the case of sparse balanced MIL (sbMIL), prior knowledge on the “sparsity” of positive bags can be specified or found via cross-validation. Besides the sparse MIL family, Miss-SVM[10] is proposed that uses a semi-supervised learning approach, treating the instances in positive bags as unlabeled data.

3.1.4 Prediction with Pattern Mining Method

Sequential pattern mining is applicable for those datasets that with a large amount of event data. BI-Directional Extension based frequent closed sequence mining (BIDE) is proposed by Wang and Han (2004)[11]. The “closed” means only report those patterns who cannot be extended with additional items without lowering support. To realize the closed pattern mining, this algorithm recursively extends patterns, while their frequency is above the minimum support, checking closure properties of the extensions. Besides BIDE, a Margin-closed Frequent Sequential Pattern Mining is proposed by Fradkin, Dmitriy, and Fabian Moerchen (2010)[12]. The Margin-closed means pattern cannot be extended by additional items without lowering the support significantly, as determined by a relative or absolute threshold. This algorithm is more efficient and robust by reducing more (almost) redundant patterns. Except for Margin-closed BIDE, Li, and Wang (2008)[13] proposed gap-constraint BIDE, which defines a gap size constraint inside each pattern. They use a gap-constraint g(M,N) to represent a gap (a sequence of wild-cards) with size in the range [M, N]. For example, there are two pattern sequence S1 = ACTTACAGTT, S2 = ACCCATATG. Then sequence $P = \langle A, g(2; 3), T, g(2; 3), G \rangle$ matches sequence S1, and ACTTACAG is an appearance of P in S1, but P does not match sequence S2.
3.2 Relevant Methodologies for our Research Questions

3.2.1 Artificial Neural Network

Artificial neural network (ANN) has been considered to be one of the most promising approaches for prediction of RUL due to their adaptability, nonlinearity, and the ability of arbitrary function approximation[15].

A typical neural network consists of a single input layer, one or more hidden layers and an output layer, each comprising of one or more nodes. Connections between nodes in adjacent layers are weighted. An activation function is associated with each node that defines if and how information is transmitted to subsequent nodes. Calculated values of each nodes activation function are then used as inputs to any subsequent nodes (see Figure 3.2 single node behavior). As processing (computing the activation function) can be performed by the nodes in parallel, neural networks are computationally very efficient. The global behavior of a particular network is determined by its architecture (nodal arrangement), synaptic weights and parameters of the nodal activation function[16].

3.2.2 Random Forest - Overview and Comparison with Other Algorithms

Random forest is a notion of the general technique of random decision forests[17]. As an ensemble learning method for classification, regression, and other tasks, random forests operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Some researchers [18] have proved that random forests correct decision trees’ habit of over-fitting to their training set by applying bootstrapping aggregation when selecting training samples for each tree.

The first algorithm for random decisions forests is created by Tin Kam Ho [17] and it applies random subspace method [19] during the random selection of features. After that, Leo Breiman[20] and Adele Cutler[21] extend the algorithm by combining Breiman’s bagging idea and random selection of features proposed by Tin Kam Ho [19].

Tree Bagging and Feature Bagging

The tree bagging idea is to repeatedly (the repeat time is the number of trees) select a random sample with replacement from the data set and fits trees to these samples. With the arranged samples, the algorithm trains several classification or regression trees. After the training, final
predictions results for test samples can be made by the majority voting (for classification problem) or average predictions (for regression problem) from all the individual trees.

As for the training procedure on each single tree, random forests apply feature bagging: in each node’s split, the algorithm randomly selects several features; and then from these selected features, the algorithm chooses the feature that could reduce most Gini impurity (see Section 6.3.2) or information gain (entropy) to split the node. If a few features are strong predictors for the target variable (output), then they will be selected by many trees [22].

Advantages and Disadvantages

There are many reasons why we choose random forests for our project:

- Compared with the decision tree, random forests use bootstrapping bagging method which reduces the variance of the model without increasing the bias, thus increase the robust of the model. More precisely, even when a single tree’s prediction results are highly sensitive to noise in its training set, the average results of the forest are not, since the trees are not correlated. [20]

- Compared with Support Vector Machine (SVM), random forests are more easily in the implementation without depending sensitively on parameters settings. The SVM models could perform quite variously with different kinds of kernel functions and different penalty parameters (C in Formula 3.1) of the cost functions. In other words, if our SVM classifiers perform badly or underperform, it could due to the improper kernel or penalty parameters, and we cannot say the categories are not distinguishable. Since our project is a start exploration on the predictive maintenance of ECM tools and we have no idea of the distinguishability of the maintenance categories, random forests are more suitable to quickly give a promising direction.

Learning an SVM model has been formulated as a constrained optimization problem, and the cost function for binary-class SVM is:

\[
J(\theta) = \min C \sum_{i=1}^{m} [y^{(i)} \text{cost}_1(\theta^T f^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T f^{(i)})] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2 \tag{3.1}
\]

In Formula 3.1, there are 2 classes, \(y = 0\), and \(y = 1\). The \(y^{(i)}\) means the class label for \(i^{th}\) sample. The \(f^{(i)}\) means transformed feature vector for \(i^{th}\) sample. The functions \(\text{cost}_0\) and \(\text{cost}_1\) calculate predication costs when \(y = 0\) or \(y = 1\) respectively. With linear kernel, \(f^{(i)} = x^{(i)}\), where \(x^{(i)}\) is the original feature vector. With Gaussian kernel, \(f^{(i)} = \text{similarity}(x, l^{(i)}) = \exp(-\frac{||x - l^{(i)}||^2}{2\sigma^2})\). The second part of Formula 3.1 is the regularization item \(\frac{1}{2} \sum_{j=1}^{n} \theta_j^2\). This item is used to add penalty on error terms.

- Compared with random forests, Artificial Neural Network (ANN) cannot provide information about the relative significance of the various parameters. Except for that, ANN performances are quite relevant to its architecture (nodal arrangement) as we discuss in Section 3.2.1. Determining the most appropriate model is largely trial and therefore can be time-consuming.

In Section 2.2.2, we explained one research question about classification on different maintenance operations. In Section 6.2, we discussed the categories definition and we prefer to find a baseline algorithm to test all interesting groups, so we choose random forests. After we find promising classification groups and get more data in the future, then we can apply other algorithms like ANN to see if it is possible to improve the performance.

In addition, random forest have the following advantages which are also common in many machine learning algorithms: it runs efficiently on large databases (since the random forest algorithm is constructed using bagging or boosting); it can handle thousands of input features without variable pre-selection; it gives features importance rank; the generated forests can be saved for future use on other data; it offers an experimental method for detecting variable interactions.

However, there are also some drawbacks on random forests:
• The random forest calculate each feature’s importance score by averaging the difference in out-of-bag error before and after the permutation over all trees. Thus, if the data contain groups of correlated features of similar relevance for the output, then smaller groups are more easily to get higher scores than over larger groups.[23]. However, this drawback is not a problem in our project, since we know the physical meaning of each feature and we know the possible groups (of related features); besides, we concern many top features and do not need to care the specific scores.

• The random forest model is not easy to interpret. We know the final prediction results of random forest are calculated by results from all trees and each tree only trains a sample of the dataset, so it is not reasonable to interpret the model by a single tree. However, gaining a full understanding of the decisions by examining all trees is infeasible when the forest is large.
Chapter 4

Development Approach on Data Preparation

In this chapter, we introduce the data pre-processing on our use case, and maintenance piece split of the original data. At the end of this chapter, we give analysis on the missing value and describe the filling methods we applied.

4.1 Data Pre-processing

4.1.1 Data Logs Format Unification

The database used to store the data was set-up back in several years ago, and still contains some errors. Since the data storage method is under improving, we have different versions of data logs. The old version is during last year, in which data collected from different machines have different formats. After several months, a new data storage strategy is applied to all machines. The data versions have different attributes naming methods, and have different formats on some attributes (e.g., different timestamp format like yyyy-mm-dd hh:mm:ss and dd-mm-yyyy hh:mm:ss). Besides, a few new attributes are not available for the first version since they were not logged at that time.

The data logs have some nominal attributes (e.g., machine label) and all other attributes are numerical. Figure 4.1 shows an example of one record. Figure 4.2 shows the record’s format unification result. After performing format unification on the old version logs, we could combine them with the new one, and form a unified dataset.

- Format unification on history data logs: match the attributes’ name with current version and alter former name → unify (according to the current format) nominal attributes’ format of each record → arrange records in time-order
- Combination: check the last product_id/timestamp in history logs → join new records with history logs

4.1.2 Extract TIP Identification from Maintenance Logs

As described in Section 2.2, we want to predict tool status and maintenance actions on electrode TIP. Thus, the TIP identification (ID) is necessary for future analysis although the first version of the database did not include it. The database now has been started recording the TIP ID. To complete the history data logs, we create an EVS_TIP attribute and fill in the values with information from maintenance logs. Table 4.3 shows an example of one maintenance record that we use to match and extract TIP ID for relevant data log (the record in Table 4.2).

Input: Maintenance logs and data logs.
Output: New data logs with TIP information. In this example, the record inserts a new attribute EVS_TIP with value “T142141”.

4.1.3 Label Replacement Points and Correct REV in Data Logs

In each data log record, $C_1$ describes the number of caps made with this tool after last maintenance, and original $C_2$ describes the total number of caps made with each HOUDER. We recalculate the $C_2$ and makes it represents the number of caps made with each TIP. The general methods are as follows:

- Delete those records that miss nominal attributes or have wrong (negative or empty) $C_1$ values.
CHAPTER 4. DEVELOPMENT APPROACH ON DATA PREPARATION

- Recalculate \( C_2 \) attribute for data logs: extract records for each TIP in time-order → check if the \( C_2 \) starts with a correct number from maintenance logs (for each TIP, its initial \( C_2 \) should be set as 0), and correct the initial number of \( C_2 \) to calculate \( C_2 \) by accumulating (see the method details below) \( C_1 \) till a HOUDER/TIP’s replacement occur → return the replaced HOUDER/TIP’s identifier and its largest \( C_2 \) as its life length.

In time order, the \( C_1 \) should be a sequence looks like 0, 1, 2, ..., /1, 0, 1, 2, ..., /2, 0, 1, 2, ..., /3. However, we notice one problem that the \( C_1 \) is not strictly increasing with the timestamp since small errors exist due to out-of-sync related issues (e.g., one product’s \( C_1 \) is 35, a later produced product’s \( C_1 \) is 09). To tackle with this problem, we apply a slack margin during the accumulation check. The accumulating method for each TIP is:

**Correction on \( C_2 \):**

**INPUT:** a slack margin, a set of all records of one TIP, and the number of records \( N \).

**Calculation:**

1. for \( i = 1 \) to \( N-1 \):
2. if \( C_1[i] < C_1[i-1] - \text{margin} \):
3. \( C_2[i] = C_2[i-1] + C_1[i] \)
   //this is for the condition that after a new maintenance operation, the EVS.ST.ONH restart from 0 or a very small value. But the EVS.ST.REV should still increase.
4. else:
5. \( C_2[i] = C_2[i-1] + (C_1[i] - C_1[i-1]) \)
   //this is for normal condition, that \( C_2 \) is increasing.

- Add new attribute - “RUL”, “RC” in data logs. The RC means rest number of caps one TIP can make, while the RUL means rest life ratio one TIP left. With the life length information (Figure 4.4), we simply calculate each record’s RUL and RC with the following equations.

\[
RUL = 1 - \frac{C_2}{\text{lifelength}} \quad (4.1)
\]
\[
RC = \text{lifelength} - C_2 \quad (4.2)
\]

![Figure 4.4: Life length distribution on TIPS](image)

4.2 Split Data into Maintenance Pieces

As we described in Section 2.2.1, process features are not absolutely calibrated among different maintenance pieces. Thus, it is necessary to consider splitting each TIPs’ data into several pieces
according to tooling CHANGE and do further feature analysis/creations inside each maintenance piece. With information extracted from maintenance logs, for each TIP, we label the number of its maintenance inspections from 0 to its final maintenance. According to the time information and TIP label, we trace the data logs and give corresponding instances (product records) batch the same maintenance label as the value of a new attribute NO\_MAINTENANCE (green lines in Figure 4.5).

![Figure 4.5: An example: T142101 X_10, original](image)

The x-axis is the counter attribute that records how many caps one TIP produced. The top subgraph shows the original data of the feature X_10 on TIP T142101. The middle subgraph shows “Error Code Accum”, which is the accumulation value of key errors (see Section 5.1 about the definition of feature ERROR\_ACCUM and analysis of key errors.). The bottom subgraph shows the “STD\_rollPer100”, which is the moving standard difference with sliding window size 100 caps.

### 4.3 Filling Missing Values

In shaving head production line, some measurements/operations are only applied to samples, so some attributes have high missing rates (e.g., only around 0.4% products have quality data). It is necessary to fill in the missing value with proper methods before building prediction models.

- Electrolyte data have high missing rate and we do not use them.
- Material data: as we discussed in Section 2.1.1, material measurements choose 1% samples from each batch. In the original dataset, some material data has been filled according to products material batch number, and the material attributes are quite stable. Since the material applied to the production are stable and could involve noises (since the low sampling rate), we do not plan to use them in our model.
- Tooling data: as we discussed in Section 2.1.1, experts only measure each TIPs initial physical parameters once before its first installation on the production line. We can then fill in the missing value by matching TIP ID. However, since the tooling data is only measured once for each TIP, it cannot reflect the trend of the TIP degradation over time.
CHAPTER 4. DEVELOPMENT APPROACH ON DATA PREPARATION

- Process Data: most process attributes have very low missing rates (less than 1%) and high variance, so for these attributes, we do not fill in the missing records but just filter those empty records (after we get trends information). Some process attributes have high missing rates or only occur in several TIPs data, so we simply do not use them (see Section 5.4). For special attributes, like the error code attribute \( E_{\text{code}} \), we fill missing position with 0, which means no error.

- Quality Data: quality attributes have very high missing rates because the quality test of the GPE process performs per \( R_{Q} \) hours on only eight products from the pilot production line. Inside each maintenance piece, we apply linear (with slope 0) interpolation backward to the last known value. However, from production experiences, the trend between quality sample tests is never linear, so the linear interpolation might involve noise. Thus, we test our model with different feature subsets (see feature subsets in Section 5.4, subset “a” and “c” includes quality data, “b” and “d” does not include quality data) to see how could the filled quality data influence the performance (see experiments in Section 6.4.2).

Another proposal is applying different data source (e.g., process data model, quality data model) in separate models. For example, when to build quality data related model, we only select those records that have quality data. As we introduced in Section 2.1.1, the original datasets have 4.56 million instances (product records). However, we only have thousands records that have quality data in the whole data set, which could be not enough for our maintenance piece classification model in Chapter 6, since there could be just hundreds training samples for each maintenance type/class.

Another reasonable method to reconstruct the quality data is by estimating the missing quality data based on the process parameters. This method highly relies on the quality prediction performance. Some researchers in Philips are working on the quality prediction project, and we expect to combine their contributions with this research in the future.
Chapter 5

Feature Observation and Extraction

In this chapter, we first analyze the process error information and then do visualizations on the original attributes to better understand the feature and possible issues of the data. We compare different methods on feature trends extraction and extract feature offsets as well as moving derivative information from the processed data. In Section 5.6, we conclude this chapter.

5.1 Process Error Analysis

In this section, we observe error occurrence density (frequency), error accumulation slope/trends, and process values when errors occur.

5.1.1 Key Errors Occurrences Density

The process attribute E_code records process error codes. The process errors are various and only some key errors, which may cause damage or pollution of tools over time, could trigger tooling maintenance inspections.

When we place a new electrode in one machine, or randomly place an electrode back to one machine after maintenance inspection; that machine usually requires a “warm-up” phase to adjust to the tool; the “warm-up” can involve several process errors in a very short time. After the ‘warm-up’, the machine runs normally and we call it “process running” phase.

We create a feature ERROR_ACCUM, which counts the number of key errors each TIP experienced. Similarly, we have ERROR_ACCUM_ONLYPROCESS which only counts the number of key errors each TIP experienced only during ‘process running’ phases.

Figure 5.1 shows the distributions of the key errors information and life length of each tip, from which we see that both the number of errors or the error density do not show linearly increasing/decreasing trends with the increasing of life length. However, our observations are from current database, which only has several TIPs, so we cannot conclude if there are relations between the key errors number and the tool life.

Except for the statistical results in Figure 5.1, we know that in our use case, the key errors do not absolutely cause damage on the tool. In the current maintenance system (as explained in Section 2.1.2), the key errors are applied as triggers for TOCAP operations, and if the operators judge there might be tool issues, then the tool is sent to maintenance. It is a fact that not all key errors could influence the tool performance or the impending maintenance actions. Thus, we do not select the key errors occurrence information as an feature/indicator for our prediction models on Chapter 6 and Chapter 7.
5.1.2 Key Errors Accumulation Slope/Trends

Except for the relationships between the errors density and life length as Figure 5.1 shows, we observe the ERROR_ACCUM feature trends (e.g., the middle subgraph of Figure 4.5), and find there are no directly observed common patterns of each maintenance class in Table 6.2. Figure 5.2 and 5.3 show two examples, in which the blue lines are ERROR_ACCUM_ONLYPROCESS while the black lines are ERROR_ACCUM. From the figures, we can see the error slopes can be both high or low in each TIP’s last maintenance pieces.
Figure 5.2: Process key errors trends of class 8 (see each subgraph/TIP’s last maintenance piece).

The figures show process error accumulation curve of TIPs whose last maintenance pieces belongs to class 8. The x-axis is production counter \( (C_2) \). The blue lines show \( \text{ERROR}_{\text{ACCUM,ONLYPROCESS}} \) while the black lines show \( \text{ERROR}_{\text{ACCUM}} \).
Figures show process error accumulation curves of TIPs whose last maintenance pieces belong to class 3. The x-axis are production counter (C_2). The blue lines show ERROR ACCUM ONLY PROCESS while the black lines show ERROR ACCUM.
5.1.3 Process Attributes when Error Occurs.

Figure 5.4 and Figure 5.5 show two examples of process features on TIP T142101, from which we can see there are many irregular values which rely on process errors. Those irregular values are caused by improper operations or machine issues during production, and we regard those irregular records as outliers.

As we explained in Section 5.1, in the shaving cap production line, even the same (with same error code) kind of errors can have different influences on electrodes. Our observations also offer evidence that key errors trends do not have direct relations with impending maintenance types. Besides, as we discussed with process engineers, we plan to distinguish normal process trends and abnormal process trends and then give suggestions on impending maintenance operations. The outliers strayed away from normal values, which is noise for feature trend extraction and we do not use them.

Based on the analysis above, we remove those irregular values according to process error codes before trend extraction. With this removal, we only keep the products that have experienced both no short-circuits and no other machine hardware issues. Figure 5.6 shows the results of Figure 5.5 after removing outliers.

![Figure 5.4: T142101 X_9, original](image-url)
5.2 Process Feature Trends Extraction and Observation

In our use case, if there is a clear pattern (e.g., on some process features) deviating from a normal trend, one might consider this to be “suspicious” and might describe a failing tool. If not, we believe the tool runs normally, and there is no need for a regular inspection. Thus, we plan...
to extract process feature trends for classification model building in Chapter 6. After removing
the outliers in Section 5.1.3, the original sensory data is still noisy and it is necessary to apply
filtering methods to extract trends. In this section, we introduce two methods: the Hilbert-Huang
transform (HHT) and the moving average (MA). The experiments in this section show that the
moving average is the simplest one and keeps most of the features in the original process signals. In
Section 6.4.3, we have controlled experiments to show how the moving average method influences
the classification results.

5.2.1 Feature Trends Extraction Based on Hilbert-Huang Transform

As introduced in MANTIS book (Appendix15 of Delivery 1.1, 2015), there are wide literature and
available methodologies on analyzing (high frequency) time series data:

- Basic analysis using chronological averages (on stock and flow typed time series), spread-like
  or element-difference based metrics (measuring volatility)
- ARMA-ARIMA based (Auto-Regressive Integrated Moving Average) models (stochastic ap-
 proaches, like the Box-Jenkins-model [25], mostly used for short-term analysis).
- Frequency - decomposition models (spectral analysis, wavelet analysis [26])
- Time series regression models (regression models fitted on time series data).

Due to time constraint, we focus on the study of frequency decomposition models [27]. In this
domain, Fourier transform and some improved transforms (e.g., wavelet transform) are commonly
used for signals decomposition; however, they require pre-specified basis functions. In our use
case, the process signals do not show stable periodic characteristics and it is difficult to define
basis functions for them.

In the shaving heads production process, the sensory data are non-linear as well as non-
stationary. Literature survey shows the Hilbert-Huang transform [28] could be a promising method
for us. The HHT is a method to decompose a signal into so-called intrinsic mode functions (IMF)
along with a trend and obtains instantaneous frequency data. From the HHT theory in [28],
we know the HHT is an empirical approach based on the “shape” (the envelope that formed by
maxima and minima of signals at each timestamp) of sensor data and is suitable for non-linear
and non-stationary data. The HHT first applies Empirical Mode Decomposition (EMD) method
to decompose the original signals to several IMFs and then compute the instantaneous frequency
using the Hilbert transform.

Figure 5.7 and Figure 8.2 show some examples when applying the HHT. From the two figures,
it is clear that the IMF components selection strategies are important in trend extraction. In
Figure 5.7c and Figure 5.7d of Figure 5.7, we see choosing different components can bring different
information: the Figure 5.7c is more smooth to reflect the whole signal trend, but it cannot fit the
last part very well in this example; the Figure 5.7d keeps more details of the signal but cannot
show the general trends. To realize automatic components selection for trends extraction, paper
[29] proposed one method. After we test this method, we find the cross energy ratio threshold
(ε in that paper) is hard to settle on our project. For example, even for one process attribute,
different TIPs’ data require different ε to get reasonable components selection results that reflect
the real trend that we observed. However, manually setting ε cannot guarantee fairness among
TIPs.

Thus, we cannot use the HHT method and need to choose another. In next section, we
introduce moving average smoothing which we finally apply. Figure 5.9 shows one example of
results (on one process signal) comparison of HHT and moving average method.
CHAPTER 5. FEATURE OBSERVATION AND EXTRACTION

Analyzing Machine Data for Predictive Maintenance of Electro Chemical Machining Electrodes
CHAPTER 5. FEATURE OBSERVATION AND EXTRACTION

(a) T142116 X_9

(b) EMD results - Intrinsic Mode Functions

(c) Components imf10+imf11+res

(d) Components imf8+imf9+imf10+imf11+res

Figure 5.8: T142116 X_9, applying HHT
We apply HHT on the maintenance pieces 2.0 and 4.0, and moving average ($W_S = 500$) on the piece 3.0. The y-axis Trend_DEV_300 in middle subgraph is the derivative with $W_{S_{dev}} = 300$ (in Formula 5.3) of the trend. The Trend_DEV_500 is the derivative with $W_{S_{dev}} = 500$ of the trend.
5.2.2 Feature Trends Extraction with Moving Average

The moving average smoothing with sliding window is applied as:

\[ y(t) = \frac{\sum_{i=t-WS}^{t} y_o(i)}{WS}, t \geq WS \] (5.1)

where the WS is the window size, the \( y_o \) is the original/input signal while the \( y \) is the output signal. After the moving average filtering, we extract two kinds of features from the process attributes. One is the Feature Offset (\( y_{offset} \) in Formula 5.2), the other is Moving Derivative (\( y_{dev} \) in Formula 5.3).

\[ y_{offset}(t) = y(t) - \frac{\sum_{i=t-WS_{offset}}^{t} y(i)}{WS_{offset}}, t \geq WS_{offset} \] (5.2)

When we apply the Formula 5.2 on process signals of each maintenance piece, the \( y_{offset} \) measure each data point’s offset from the mean value of the piece. In this thesis, the \( WS_{offset} \) equals to each maintenance piece length when we calculate feature offsets.

\[ y_{dev}(t) = \frac{y(t + WS_{dev}/2) - y(t - WS_{dev}/2)}{WS_{dev}}, t \geq WS_{dev}/2 \] (5.3)

5.2.3 Trends Observation on Process Attributes

As we stated in the chapter introduction, we do feature observation to better understand original attributes and possible issues of the data. In this section, we show the visible patterns we found in the original process attributes trends, which can be applied to detect some damages in process control monitoring.

For most of the TIPs in class 3 (one kind of damage piece, see class definition in Table 6.2), we observe some features have common trends before the discarding (end of life). By observing the process signals of TIPs in class 3, we can see the features \( X_1, X_5 \) have level drops while the feature \( X_3 \) has increasing trends at the same time. Figure 5.10 shows some examples of the trends. However, these kinds of trends do not occur in all TIPs of class 3. For other classes in Table 6.2, we find it could be hard to find direct relationships between single feature’s trend and the TIP status. These observations indicate us there could be more invisible patterns that require machine learning models to detect. In Chapter 6, we apply random forest classifiers which can classify different maintenance operation types with the features we extract in Section 5.3.

5.3 Feature Extraction Results

Based on the original attributes, we extract four kinds of features that we apply as predictors in our classification/regression models:

- **Features Offset**

  Inside each maintenance piece, for each process feature \( y \), we first do moving average smoothing as Formula 5.1, with different \( WS \) (e.g., 200, 500) respectively, and then calculate \( y_{offset} \) as Formula 5.2 shows. Since the original signals are noisy, we assume the moving average could improve our prediction models’ performance. To verify this assumption, we also calculate feature offsets without the “moving average” step and compare the classification results in Section 6.4.3. In this thesis, the default \( WS \) for feature offsets is 200 in Formula 5.2.

  The absolute values of original process attributes are incomparable among different machines; after we add this mean offset, they become comparable. From Figure 5.11, we see the offset (in the middle subgraph) could reflect the trend inside each maintenance piece and make pieces comparable.
In each subgraph, the x-axis is $C_2$, the product counter. We see level drops/increases in last products of TIPs in class 3. In this figure, we already removed irregular values when process errors occur.
CHAPTER 5. FEATURE OBSERVATION AND EXTRACTION

• **Moving Derivative**
Inside each maintenance piece, for each process feature \( y \), we first do moving average as Formula 5.1 with \( WS = 500 \), and then calculate moving derivative as Formula 5.3, with \( WS_{DEV} = 300, 500, 1000 \) respectively. The Figure 5.12 shows the results.

• **INDICATOR\(_1\)**

\[
INDICATOR_1 = \frac{X_9}{X_{11}}
\]  
(5.4)

where \( X_9 \) and \( X_{11} \) are two kinds of voltage attributes during production. The indicator is suggested by process engineers.

• **INDICATOR\(_2\)**

\[
INDICATOR_2 = (X_8) - (X_7)
\]  
(5.5)

where \( X_7 \) and \( X_8 \) are two kinds of distance measurements in the machine.

![Image](image_url)

Figure 5.11: Two examples on feature offsets.

The x-axis is \( C_2 \), the product counter. In each figure, the top subgraph is the original data (blue line) and the trend (red line, calculated by Formula 5.1 with \( WS = 200 \)); the middle subgraph is the offset value calculated by Formula 5.2 in which the \( y \) is the red line in the top subgraph; the bottom subgraph is the ERROR\(_{ACCUM} \); the green lines are number of times of maintenance of each TIP.

5.4 **Feature Selection Results and Feature Subsets for Predictors**

• **Tooling data** As we discussed in Section 2.1.1, the tooling data (except for counters) only measure once for each tool, so we do not apply them to predict impending maintenance operations. In Chapter 7, when estimating RUL of tools, we have experiments that involve tooling attributes as predictors.

• **Material data** As Section 4.3 shows, the materials are quite stable and the material data for each product can involve noise since the low sampling rate on the material tests, so we do not use it.
CHAPTER 5. FEATURE OBSERVATION AND EXTRACTION

Figure 5.12: T142112 X_1 moving derivative (WS_{dev} = 500)

The blue line in top subgraph shows the process attribute X_1 on TIP T142112 after removing outliers; the red line in top subgraph shows X_1’s trend calculated by equation 5.1 with WS = 500; the following 3 subgraphs show the moving derivative of the trend (red line) with different sliding window size (WS_{dev} = 300, 500, 1000). From which we can see the three moving derivative curves have similar shape and can reflect significant fluctuants on the attribute.

- **Process data** The process data is one quite important data source that we choose. In all process attributes, some attributes have high missing rates (>99%) or only occurs in several tools records, so we do not choose them as predictors. We finally have 12 process attributes which have both low missing rate and related physical meanings. Figure 5.14 show explanations on the 12 main process features.

- **Quality data** From process engineers’ experience, the quality data is an important source that used by operators to make maintenance decisions. In our dataset, the quality data have high missing rate and we fill them with linear interpolation. Since we do not know if the filling method is reliable, we want to see if it could add information on our prediction. Thus, we choose different feature subsets as predictor variables for our prediction models and check the influences of quality attributes.

The feature subsets that we apply for predictors in classification/regression models of Chapter 6 and Chapter 7 are:

- **a**: INDICATOR_1, INDICATOR_2, feature offsets and moving derivatives of 12 main process attributes and INDICATOR_2, quality attributes
- **b**: INDICATOR_1, INDICATOR_2, feature offsets and moving derivatives of 12 main process attributes and INDICATOR_2
- **c**: C_1, INDICATOR_1, INDICATOR_2, feature offsets and moving derivatives of 12 main process attributes and INDICATOR_2, quality attributes
Figure 5.13: T142112 X_1

The top subgraph is the process feature X_1 on TIP T142112 after remove outliers (blue line) and the trend (red line, calculated by equation 5.1 with WS = 200); the middle subgraph is the offset value calculated by equation 5.2 in which the y(t) is the red line in the top subgraph; and the bottom subgraph is the ‘ERROR ACCUM’; the green lines are maintenance pieces label]

- **d**: C_1, INDICATOR_1, INDICATOR_2, feature offsets and moving derivatives of 12 main process attributes and INDICATOR_2

For feature offsets, the calculation method is as Section 5.2.2 shows, and we apply default sliding window size as \( WS = 200 \) in Formula 5.1. In Section 6.4.3, we have controlled experiments results on choosing other \( WS \) values. For feature moving derivatives, the calculation method is as Section 5.2.2 shows, and we choose default \( WS = 500 \) in formula 5.1, and default \( WS_{DEV} = 300, 500 \) in Formula 5.3.

### 5.5 Other Issues on Process Attributes

Except for the CHANGE after maintenance, there are some other exceptions which might cause some process features jumping such like changing of internal parts. We can find some jumps in certain TIPs (see Figure 5.15). Those exceptions rarely occur, but still requires special treatments in the future.

### 5.6 Conclusion

In this chapter, we first analyze the relations between key errors and tool performance. We find the number and the occurrence frequency of key errors have no direct relations with tool life length (the total number of caps one tool could make). In addition, we observe key errors occurrence accumulation trends of maintenance actions classes and give figures of class 3 and class 8 (see class
As we explained in Section 2.1.2, in the current maintenance system, the key errors are applied as triggers for TOCAP operations, and if the operators indicate there might be tool issues, then the tool will be sent to maintenance inspection. It is a fact that not all key errors always influence the tool geometry. Our observations also conform this fact and we do not choose key errors accumulation information as an indicator for our prediction models on impending maintenance actions and electrode’s RUL. Based on the use case background, we assume that if some errors influence the machine or the tool, then the following process feature trends can be used as indicators to distinguish the influences.

Figure 5.15: Jumps inside one Maintenance Piece
In the second part of this chapter, we show feature observations to understand the dataset. The motivation for feature trend extraction is to distinguish abnormal running patterns with normal running patterns. We give the results of trend extraction based on HHT, however, we did not choose the HHT method for final trend extraction since the component selection could make unfairness among features from different TIPs. We finally choose the moving average method that we showed in Section 5.2.2 and propose our solutions to the calibrating drift problem of the process features from different machines.

Finally, we give feature extraction results and feature selection results. The feature selection is mainly based on the missing rate of the whole dataset. We select all features that we can use and form several features subsets (a, b, c, d in Section 5.4). In model building in Chapter 6 and Chapter 7, we do forward selection by putting all features into the model and make the algorithm select features automatically.
Chapter 6

Classification of Maintenance Types

In this chapter, we first define the maintenance operations (include maintenance trigger and specific action) as 10 classes in Section 6.2 and show classification experiments results in Section 6.3. Based on the classification results, we combine many similar classes into one large class and give a new class definition in Section 6.4.6. We finally perform classification on the whole dataset (except for unclear/special cases) and give the performance analysis.

6.1 Motivation and Target

The motivation for maintenance pieces (MP) classification (distinguish between normally running pieces and abnormal pieces) is to give maintenance suggestions on any running TIPs upfront, and then to minimize downtime of the production machine for tool changeovers as well as reduce costs on unqualified products. In the production process, if there is a clear pattern (on process features) deviating from a normal trend, one might consider this to be “suspicious” and might describe a failing tool. If not, we believe the tool runs normally, and there is no need for a regular inspection. In addition, with normal/abnormal trend classification, we can catch indicators for key errors (which may cause unrepairable damage on TIP) and stop relevant machines in an earlier stage, and reduce the losses.

In Section 4.2, we already split each TIPs data logs into several maintenance pieces according to the maintenance logs information. Each piece has an attribute NO_MAINTENANCE to record how many times the TIP has been sent to the maintenance center. For each TIP, by matching the NO_MAINTENANCE in both data logs and maintenance logs, we could get the specific maintenance operations after each piece. For example, TIP T142116 has one piece the piece 0.0, and after this piece, the T142116 is replaced since it has unrepairable damage. The relevant maintenance log codes are “T_5”, “O_5”, in which the “T_5” means one kind of trigger for the maintenance inspection, and “O_5” is the specific operation. With the maintenance operations information, we could give each maintenance piece a label ([T_5, O_5] in this case) prepared for classification. Table 6.1 shows maintenance operation codes (related to GPE process) in the maintenance logs.

6.2 Maintenance Pieces Classes Definition

One problem for the MP classes definition is that the maintenance operations are complicated, and include many examples which reside in an intermediate zone between normal and abnormal. Table 6.2 shows the MP classes definition that we apply. When defining the MP classes, we do not consider those maintenance records which does not include trigger reasons from GPE process.
CHAPTER 6. CLASSIFICATION OF MAINTENANCE TYPES

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger</td>
<td>T_1</td>
<td>The production run counter has reached the regular inspection interval</td>
</tr>
<tr>
<td>Trigger</td>
<td>T_2</td>
<td>No specific reason</td>
</tr>
<tr>
<td>Trigger</td>
<td>T_3</td>
<td>Quality deviation (outside specifications limits)</td>
</tr>
<tr>
<td>Trigger</td>
<td>T_4</td>
<td>Process errors relate to the electrode</td>
</tr>
<tr>
<td>Trigger</td>
<td>T_5</td>
<td>Visual deviation on products</td>
</tr>
<tr>
<td>Operation</td>
<td>O_1</td>
<td>Clean</td>
</tr>
<tr>
<td>Operation</td>
<td>O_2</td>
<td>Replacement without a reason given</td>
</tr>
<tr>
<td>Operation</td>
<td>O_3</td>
<td>Replacement since geometrical deviation on the tool</td>
</tr>
<tr>
<td>Operation</td>
<td>O_4</td>
<td>Replacement since the TIP reach lifetime limit</td>
</tr>
<tr>
<td>Operation</td>
<td>O_5</td>
<td>Replacement since the TIP is damaged and cannot be repaired</td>
</tr>
<tr>
<td>Operation</td>
<td>O_6</td>
<td>Apply minimal repairs on the electrode</td>
</tr>
<tr>
<td>Operation</td>
<td>O_7</td>
<td>Replacement since it is the second time one electrode is sent to maintenance</td>
</tr>
<tr>
<td>Operation</td>
<td>O_8</td>
<td>Anything is worth mentioning, or no code is available for the reason of change.</td>
</tr>
<tr>
<td>Operation</td>
<td>O_9</td>
<td>Engineering test tool</td>
</tr>
</tbody>
</table>

Table 6.1: Maintenance codes

Based on the experiments results on the classification, we get new combination of maintenance classes and finally define only three classes in Section 6.4.6.

<table>
<thead>
<tr>
<th>Class Label</th>
<th>Class Meaning</th>
<th>#MP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Regular inspection + Clean (no issues on tool)</td>
<td>33</td>
</tr>
<tr>
<td>2</td>
<td>Error Trigger + Discarding (reach life length limit)</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Visual deviation + Discarding (damage on tool)</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Damaged, but repairable</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>Discarding (one kind of damage)</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>Discarding (one kind of damage)</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>Visual deviation + Discarding (reach life length limit)</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>Normal end (discarding (reach life length limit))</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>Cases including additional notes</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>Error Trigger + Clean (no issues on tool)</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 6.2: MP classes

According to the case background, we do the following analysis and then propose possible distinguishable groups (data subsets) that we are interested in for the classification:

- From Table 6.1, we can see class 1 means perfectly (normally) running pieces.
- As long as a tool behaves normally, it will most likely end up somewhere between L-R and L products. Once one TIP’s production counter reach limit L-R, the maintenance operators discard the TIP anyway. For class 2, the TIP was discarded because the “target” life limit was reached, which indicates the tool can be still healthy or somewhat worn out, and we cannot guarantee product quality. Thus, the class 2 means the TIP was replaced with quality/process errors; however, it is not clear if the TIP is replaced with some damages. The class 8 means that the tool was changed without any quality/process errors but might show worn out trends.
• For class 3, the T_5 is one kind of error on which operators send the tool to the maintenance center. At the end of each production cycle, operators are supposed to do a quick visual inspection of products. If they find any deviation, they book the tool to maintenance with this code. The reason of T_5 and O_5 coming together so often is that (large) damages or big worn out on the electrode usually cause visual deviations on products. Class 3 is one main kind of maintenance types that we want to predict as soon as possible.

• For class 4, the small repairs can be many different actions, which means the maintenance department could actually find potential causes of the error. If O_5 is preceded by a quality issue or process error, it means that the tool was damaged, but repairable. After repair, the tool will be put on a machine again. Since we want to catch the TIPs that require repair as soon as possible, class 4 is one kind of maintenance types that we concern.

• For class 5 and class 6, these maintenance actions mean the TIPs are discarded with damages.

• As we explained, once the TIP exceeds production limit, then we discard it anyway, so we do not know the real discarding reason for class 7. Class 9 includes unclear description. The class 7 and class 9 are unclear cases and seldom occur in our dataset.

• For class 10, the triggers mean the tool is sent to maintenance center with certain quality issues or process errors, but the maintenance engineers found the electrodes are healthy. Since there are many kinds of errors, the process features pieces in this class can perform variously trends and we need to find the common patterns.

In conclusion, the class 3, 4, 5, and 6 are issues (TIPs discarded with damage) that we want to distinguish from the normal classes. The pieces in class 3 might show seriously worn out trends at the tail part, the pieces in class 4 might show various kinds of deviating trends, the pieces in class 8 might show very slightly worn out trends, and the class 1, as well as class 10, might show normally running trends. The class 7 and 9 are unclear cases that lack more descriptions from maintenance logs, so we do not consider them in our classification models. The pieces in class 2 are some cases that we do not know if the TIPs are discarded with damage, but we know there are process/quality issues on the pieces.

The final target for maintenance pieces classification is giving any running piece reasonable maintenance suggestions and catch abnormal (TIP damage) pieces in time. However, considering there might be some classes share common patterns (e.g., damage class 3 and 4 might have overlap and are not easily to be distinguished), we first select many interesting groups (class subsets) that we want to check if the classes in each group are distinguishable; besides, considering there are some classes that have very few (lower than 5) pieces in current dataset, we do not include them in our interesting groups for data subset classification. We choose the following interesting groups: \{1, 3\}, \{1, 4\}, \{3, 8\}, \{4, 8\}, \{3, 10\}, \{4, 10\}, \{3, 4\}. We do classification on each data subset, and then apply the classification model to other pieces (rest classes) to see the rest (hard) examples are more likely to be classified to which class. For example, when we do classification on group \{1, 3\}, we filter the pieces in these two classes, and do classification on the selected data subset, and then test the rest classes. The experiments on all interesting groups could give us evidence on the assumptions that we made in this section and could indicate us how to do class combinations for those overlapping classes.

6.3 Experiment Design

We design several controlled experiments, and each experiment applies pieces-fold cross-validation: take the classification on group \{3, 8\} as an example, class 3 has 9 pieces with different numbers of data points (product records) while class 8 has 7 pieces. All of the pieces are used as the test set once to evaluate a classifier that is trained on the remaining data points in the other 15 pieces. The Figure 6.1 shows the idea. After the pieces-fold tests, we calculate average precision, recall, f1-score, accuracy for the whole dataset, based on prediction results on each data points.
In addition, we get prediction probability score for each data point, which can be used to draw Receiver Operating Characteristic (ROC) curve and calculate Area Under Curve (AUC) for the whole dataset. The first motivation for this pieces-fold validation is that including one piece’s information in both training set and test set is unreasonable. Besides, since data points from each piece are set as test set once, we can avoid model variance and check if the built models are over-fitting. Thus, we can compare how robust one model setting is on the dataset.

### 6.3.1 Piece Selection

We select pieces that are longer than 5000 (have long-term trends, made more than 5000 products) and have a stable sampling rate, and then remove “warm-up” records. After that, we take tail 3000, 4000 or 5000 records from each piece for further experiments. The “pick tail” method is based on one assumption: in abnormal pieces (e.g., class 3), it is possible that the TIP first runs normally before some error occurrence leads to following abnormal behaviors. Thus, we pick tail records and compare the classification accuracy with the case that we use the full pieces.

### 6.3.2 Algorithm Selection and Parameters Tuning

The implementation requires python 2.7 runtime environment, and the following python packages: NumPy 1.10.4, scikit-learn 0.17, and pandas 0.17.1. We apply random forests algorithm on the classification experiments and choose Gini’s Diversity Index (gdi, a measure of node impurity) as the criterion for nodes splitting:

\[
gini = 1 - \sum_i p^2(i) \tag{6.1}
\]

where the sum is over the classes \( i \) at the node, and \( p(i) \) is the observed fraction of classes with class \( i \) that reach the node. A node with just one class (a pure node) has gini 0; otherwise the gini is positive. At each node, the algorithm selects the feature that could reduce most Gini impurity to split the node. Except for the split standards, there are some problems that we should notice when applying parameter settings:

#### Deal with Unbalanced Problem

- **class_weight**: in order to keep balance on each tree, we give each class a weight according to its subsample size. For example, the class weight of class “y” is calculated by \((n_{\text{samples}}/(n_{\text{classes}} + \text{np.bincount}(y)))\) for each tree, where the \(n_{\text{samples}}\) is the number of samples in the training set, the \(n_{\text{classes}}\) is the number of classes we have, and the
np.bincount(y) means the number of occurrence of the class "y". In the experiments in Section 6.4, the default setting for class_weight is "balanced_subsamples". When we define class weight in our model, the gini impurity calculation takes the class weight into account:

\[ p(i) = \frac{\text{class_weight}_i \times \text{occurance}_i}{\sum_{k \in \text{classes}} \text{class_weight}_k \times \text{occurance}_k} \] (6.2)

- sampling to make balanced dataset: we can do random sampling on the large group to make balanced training sets. We do experiments on group \{1, 3\}, which has serious unbalance problem. Figure 6.6 shows the results with/without sampling on binary classification \{1, 3\}.

**Avoid Over-fitting**

In random forest algorithm, we can set the minimum number of samples required to split each internal node (min_samples_split) to avoid over-fitting. For example, we can set min_samples_split as 2, 5, 10, 20, 30, 40 etc. Besides, we can define the maximum depth of each decision tree or set the minimum weighted fraction of the input samples required to be at a leaf node (min_weighted_fraction_leaf) to realize tree pruning.

We apply grid search on different parameter settings (see experiments in Section 6.4.2 and detailed results in Appendix A). Considering the target in current stage is not finding best parameter settings but exploring practical methods, so we do not apply grid search on a very large scale.

**6.3.3 Evaluation Approach**

**Precision, Recall, F1-score, on Data Point (sample/instance) Unit**

As Figure 6.2 shows, the results for a binary classifier can be represented in a confusion matrix, where True Positive (TP), False Negative (FN), False Positive (FP) and True Negative (TN) represent the number of examples falling into each possible outcome. Incorrect predictions are clearly broken down into the FN or FP cells.

<table>
<thead>
<tr>
<th>Actual</th>
<th>Prediction</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>TP</td>
<td>FN</td>
<td></td>
</tr>
<tr>
<td>Negative</td>
<td>FP</td>
<td>TN</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.2: Confusion matrix for classifier evaluation

\[ \text{precision} = \frac{TP}{TP + FP} \] (6.3)
\[ \text{recall} = \frac{TP}{TP + FN} \] (6.4)
\[ F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \] (6.5)

A low precision can indicate a large number of False Positives. A low recall indicates many False Negatives. Suppose we want to predict failure only if very confident, then we have more strict requirements on precision than recall. Suppose we want to avoid missing too many failure cases (avoid False Negatives), then we have more strict requirements on recall than precision. We can tune parameters based on practical requirements. The F1 score conveys the balance between the precision and the recall. It ranges from 0 to 1, and the best value is 1.
CHAPTER 6. CLASSIFICATION OF MAINTENANCE TYPES

Average Accuracy (accuracy_weighted, accuracy_unweighted) based on Piece Unit
In the classification models, we predict each sample/instance’s class. However, when we considering the actual use of the classification, we should also evaluate the accuracy of each piece. Each piece includes many samples, and the classification result of each piece is based on voting results of the samples. We have two voting strategies: one is the unweighted voting, which means we give the majority results of the samples as the result of the piece; the other is the weighted voting, in which each sample get a voting weight according to its “C1” value.

The piece prediction accuracy based on majority (unweighted) voting is calculated by

\[
\text{Acc}_{\text{unweighted}} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6.6)
\]

The accuracy based on weighted voting is calculated by

\[
\text{Acc}_{\text{weighted}} = \frac{TP_{\text{weighted}} + TN_{\text{weighted}}}{TP_{\text{weighted}} + TN_{\text{weighted}} + FP_{\text{weighted}} + FN_{\text{weighted}}} \quad (6.7)
\]

in which the confusion matrix for Acc_weighted takes sample voting weights into account.

Receiver Operating Characteristic (ROC)
For binary classifier, we apply the ROC curve as one tool to evaluate the performance. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity/recall. The false-positive rate is also known as the fall-out and can be calculated as (1 - specificity).

Some researchers (Barakat and Bradley (2006) [30]) have proved that ROC curves and AUC could provide a more reliable measure of quality than the accuracy. We apply AUC statistic as one standard for model comparison. The AUC ranges from 0 to 1, and the best value is 1, which means a perfect discriminant ability between the classes.

\[
\text{TPR} = \frac{TP}{P} = \frac{TP}{TP + FN} = \text{sensitivity} = \text{recall} \quad (6.8)
\]

\[
\text{FPR} = \frac{FP}{N} = \frac{FP}{FP + TN} = 1 - \text{specificity} \quad (6.9)
\]

The area under the ROC curve (AUC) is given by (the integral boundaries are reversed as large T has a lower value on the X-axis)

\[
A = \int_{-\infty}^{\infty} \text{TPR}(T) FPR'(T) \, dT = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(T' > T) f_1(T') f_0(T) \, dT' \, dT = P(X_1 > X_0) \quad (6.10)
\]

where \(X_1\) is the score for a positive instance and \(X_0\) is the score for a negative instance.

6.4 Experiments Results and Performance Evaluation
In this section, the default tree bagging method is “bootstrapping” (see explanation in Section 3.2.2), the default feature subset that we applied for predictor variables is group b, the number of features to consider when looking for each node’s best split is \(\sqrt{n_{\text{features}}}\) (the \(n_{\text{features}}\) is the number of features we use as predictors), and the default class weight setting is “balanced_subsamples”. For all tables that show experiments results, we have unified abbreviations for headers, “P” means “feature subset (group)”, “n” means the number of decision trees in each model, “mss” means “min_samples_split”, “mwfl” means “min_weighted_frac_leaf”, “ACC_unw” means “Accuracy based on unweighted voting”, “ACC_w” means “Accuracy based on weighted voting”, “ROC_area” means the “AUC of ROC”, “F1_ave” means the average F1 score of all classes in each model. The ROC positive labels are \{1, 3\}: 3, \{1, 4\}: 4, \{3, 8\}: 3, \{4, 8\}: 4, \{3, 10\}: 3, \{4, 10\}: 4, \{3, 4\}: 3, \{1, 10\}: 1, \{1, 8\}: 1, respectively.
CHAPTER 6. CLASSIFICATION OF MAINTENANCE TYPES

6.4.1 Classification Results on Interesting Groups

Figure 6.3 shows the F1 Score on all interesting groups from 5 experiments with different parameter settings: pick tail 3000 data points from each piece, set min_samples_split as 5, 10, 20, 30, 40, respectively. Table A.1, Table A.5, Table A.2, Table A.3, Table A.4 records the detailed parameter settings and results.

In this section, we use F1-score to see if the classes in each interesting group are distinguishable. In the controlled experiments in next sections, we only do classification on group \{3, 8\}, and we use AUC score to evaluate models. We conclude the findings from the experiments in this section as follows:

- From Figure 6.3, we see the classification performance (F1-score) is not sensitive to the min_samples_split we set. Besides, the groups \{3, 8\}, \{1, 10\}, \{4, 8\}, and \{1, 4\} give statistically promising results. This indicates that class 8 can be used to distinguish damage (class 3) or small damage (class 4). In addition, class 1 and class 10 are distinguishable. In Section 6.4.6, we analyze the results shown in Figure 6.3 in detail and apply them as arguments for our maintenance type (class) combinations.

- As we discussed in Section 6.2, the pieces in class 3 and 4 can have abnormal issues (TIP damages) that we want to distinguish from the normal classes. The class 1, 8, 10 are candidates that we expect to be normal classes that can be applied to distinguish damages. In Figure 6.4 (same experiment setting as in 6.3), we compare the AUC and F1-score of groups \{1, 3\}, \{3, 8\}, \{3, 10\} or groups \{1, 4\}, \{4, 8\}, \{4, 10\}. We see that the class 8 performs best in both AUC and F1 score. Figure 6.5 shows more experiments to check if the classes in interesting groups \{3, 8\}, \{4, 8\}, \{3, 4\} are distinguishable, and the parameter settings are in Table A.2, Table A.3, and Table A.4. From Figure 6.4 and Figure 6.5, we see the \{3, 8\}, \{3, 10\} achieve higher scores than \{4, 8\}, \{4, 10\} in the 12 experiments, which indicates the class 3 and class 8 could have less overlap and are more easily to be distinguished.

- In this project, the operation in class 4 is “O_6”, which can be many different kinds of actions. Thus, the O_6 pieces can perform variously and can include various patterns, which makes it less likely to find common patterns with limited training samples. In addition, the low F1-scores of the experiment on \{3, 4\} (see results in Figure 6.5) offer evidence that the two kinds of damages could have some overlap. We combined them as one common class in Section 6.4.6 for a more coarse-grained maintenance types definition.

- We expected good results on group \{1, 3\}; however, the accuracy is low (from Table A.1 and Figure 6.3). Since class 1 includes 33 pieces while class 3 only has 9 pieces, we consider if the bad results come from the unbalanced sample size. As we stated, the default class weight setting is “balanced_subsamples” which takes samples size into account for Gini’s Diversity Index calculation on each node. For group \{1, 3\}, we also try the sampling method that we mentioned in Section 6.3.2. After we sample class 1 to make a balanced dataset, the performance is still in a low level (see Figure 6.6). The parameter settings are: pick tail 3000 data points from each piece, set min_samples_split as 5, 10, 20 respectively (see detailed settings and results in Table A.8).
CHAPTER 6. CLASSIFICATION OF MAINTENANCE TYPES

Figure 6.3: F1-score on all interesting groups
Feature subset = b, n_estimators = 10, Tail = 3000, min_samples_split = 5, 10, 20, 30, 40 in order.

Figure 6.4: Results on class 3 and class 4
Feature subset = b, n_estimators = 10, Tail = 3000, min_samples_split = 5, 10, 20, 30, 40 in order.
CHAPTER 6. CLASSIFICATION OF MAINTENANCE TYPES

Figure 6.5: Results of groups \{3, 8\}, \{4, 8\}, \{3, 4\}
Feature subset = b, n_estimators = 10. (Tail, min_samples_split) settings in order: (3000, 5), (3000, 10), (3000, 20), (3000, 30), (3000, 40), (4000, 5), (4000, 10), (4000, 20), (4000, 30), (4000, 40), (5000, 30), (5000, 40).

Figure 6.6: Results of group \{1, 3\}
The blue bars are results by applying class weight in each decision tree. The red bars are results by sampling on class 1 to make balanced datasets. Feature subset = b, n_estimators = 10, Tail = 3000, min_samples_split = 5, 10, 20 in order.
6.4.2 Controlled Experiments with Different Feature Subsets

Since group \{3, 8\} get better results than other groups, we choose them as input datasets in our controlled experiments. Figure 6.8 show results of 72 experiments on different parameter settings (see detailed settings and results in Table A.10, Table A.11, Table A.12, Table A.13, and Table A.14). The number of estimators are 3 or 10 (the first 36 experiments are 3), the tail sizes are 2000, 3000, 4000, 5000 in order, the min_samples_split settings are 5, 10, or 20. The min_weighted_fraction_leaf choose different values according to min_samples_split.

From the experiments on the four feature subsets (see feature subsets in Section 5.4) as predictors variables, we find some parameter settings that produce the best results (the main measure we reference is AUC). By extracting common important features from those top models, we get a new feature subset “e” that includes 15 key features (confidential). The experiments with feature subset “e” are also included in the figures and tables. We give the following conclusion from the controlled experiments in this section:

- From Figure 6.7, we see feature subset b reaches the highest F1 score in average, and subset c, d have highest AUC score. Compared with subset b/d, the subset a/c adds quality attributes but does not perform better than b. We have two explanation for this. Firstly, we apply linear interpolate backward methods to fill in the missing records of quality attributes, which might not bring much effective information for our model. Secondly, the quality attributes can be predicted with process features (proved by other Philips researchers’ project), so the adding of quality attributes could be not crucial.

- From Figure 6.8, we can see with each feature subsets, the performance are different with different parameter settings, but there are no clear trends for increasing or decreasing. The AUC on the second half experiments (37-72, n_estimators = 10) reach a higher level than the experiments 1-36 (n_estimators = 3).

![Figure 6.7: Results (average of the 72 experiments) on \{3, 8\}, with different Feature Subsets](image-url)
CHAPTER 6. CLASSIFICATION OF MAINTENANCE TYPES

6.4.3 Controlled Experiments with Different Trend Extraction Strategies

In Section 5.2.2, we proposed moving average smoothing on original signals. Figure 6.9 clearly shows the improvements after applying sliding window smoothing on the original data, and all the experiments are implemented on group \( \{3, 8\} \). Different sliding window size also has slight differences, these experiments show that window size 500 performs a bit better than 200. However, considering the case background, we require the prediction models catch abnormal behaviors as soon as possible, so we choose 200 finally for all other experiments in Section 6.4. The detailed parameter settings in Figure 6.9 are listed in Table A.9. Figure 6.10 show some ROC curves examples from the experiments, in which we can see the performance improvements on ROC area.
CHAPTER 6. CLASSIFICATION OF MAINTENANCE TYPES

(a) AUC and F1 score

(b) Accuracy

Figure 6.9: Results with different trend extractions on predictors, of group \{3, 8\}

(a) original data  (b) sliding window size = 200  (c) sliding window size = 500

Figure 6.10: ROC - compare trend extraction influence on group \{3, 8\}, pick tail size = 4000
6.4.4 Controlled Experiments with Different tail size on “Pick Tail” Strategy

Figure 6.11 and Table A.9 show the results when we choose different tail size of each piece, and all the experiments are implemented on group {3, 8}.

When we choose full pieces as our dataset (for training/test), the results are still acceptable (F1 scores are higher that 0.66, and AUC is higher than 0.7), but are lower than “5000”, “4000” performance. This result indicates that we can choose a proper tail size (e.g., 4000, 5000) in the future classification models. In addition, we see the “2000”, “3000” have lower scores, and one reason could be the shortage of training examples.

![AUC and F1 score - Compare different 'Pick Tail' Strategies](image)

Figure 6.11: AUC and F1 score - Compare different ‘Pick Tail’ Strategies

The dataset size (support) - for tail size 2000: [18000, 14000], for tail size 3000: [27000, 21000], for tail size 4000: [36000, 28000], for tail size 5000: [45000, 35000], for full pieces: confidential.
6.4.5 Conclusions on Top Results we Get of Group \{3, 8\}

Table A.7 shows best (ranked according to AUC) results that we get from experiments in Section 6.4.2. Figure A.1, Figure A.2, Figure A.3, and Figure A.4 show results (confusion matrix, features importance rank, ROC curves) of some examples from top results we get. As we discussed in Section 6.3, in the pieces-fold experiments, each time we choose one piece as the test set and get the features importance rank from the random forest classifier. After testing all pieces, we calculate average features importance score from all the tests and get the final features importance ranks as the “features importance rank” subgraph in the figures. Except for the total average score, we also get the average features importances scores of each class (according to the test piece’s class in each test), and we show the standard differences of different classes as the standard difference indices (the blue lines) in the figures. The results are analyzed in Section 6.4.6 for classes combination.

6.4.6 Maintenance Classes Combination and Classification

In above experiments, we focus on the controlled experiments to distinguish class 3 and class 8. However, we see that there are quite many types (classes in Table 6.2) of maintenance in Philips Drachten department. In the real application, we need to apply classification on the whole dataset, not on a data subset. As we explained in Section 6.2, we have some classes that only have seldom samples in our dataset so we did not set classification models on them. Thus, we propose to combine the classes into fewer groups and do classification on the whole dataset (except for class 7 and class 9, which are not clear cases). The new classes can include “normal, no need for inspection”, “damage, requires replacement or repair”, and even “middle case, hard to say”.

We list some conclusions from experiments in above sections that help us to make the new (combined) class definition:

- First, we already show that the class 2, 5, 6 have limited number of data points, so we just regard them as “rest (hard) examples” in above experiments. The class 7 (we do not know the real discarding reason), and class 9 (we do not know the contents of maintenance notes) are not regular maintenance types so we do not consider them when we combine the regular types.
CHAPTER 6. CLASSIFICATION OF MAINTENANCE TYPES

- **damage, requires replacement or repair** Based on our analysis in Section 6.2, the class 3, 4, 5, and 6 are damages that we want to catch as soon as possible. In the experiments in Section 6.4.1, we see that some classes are not distinguishable, which indicates they might have common patterns. For example, the class 3 (small damage) and class 4 (serious damage) are not distinguished from our experiments results (see Figure 6.5).

In addition, from confusion matrix that we get (see examples in Figure A.1, A.2, A.3, A.4), the class 4, class 5 and class 6 are more likely to be classified as class 3, which indicates the process running trends might have common patterns before damage operations 3, 4, 5, and 6.

Thus, we combine class 3, 4, 5, 6 as damage class.

- **normal, no need for inspection** Based on the analysis in Section 6.2, the class 1, 8, 10 are candidates for normal pieces that can be used to distinguish abnormal (damage) pieces.

From our experiments results in Section 6.4.1, the class 1 and class 10 are distinguishable; the class 10 and class 3 are not very distinguishable (F1-scores are a bit higher than 0.5); class 1 can be used to distinguish class 3 as well as class 4; the class 10 and class 4 are not very distinguishable (F1-scores are a bit higher than 0.5); class 8 get F1-scores higher than 70% , 60% when it is applied to distinguish class 3, and class 4 separately.

Thus, we combine class 1 and 8 as normal classes.

- **middle case, hard to say** Except the above analysis on combining 1 and 8, we left class 2 and class 10. Based on the analysis in Section 6.2, we see that both class 2 and class 10 includes quality/process error codes triggers. From the confusion matrix we get (see examples in Figure A.1, A.2, A.3, A.4), we see class 2 and class 10 do not show preference for class 3 or class 8.

Thus, we combine class 2 and class 10 as middle class. For this combination, we need further analysis.

Thus, we define new maintenance types classes as Table 6.3 shows:

Table 6.3: New maintenance pieces classes define - combined results

<table>
<thead>
<tr>
<th>Class</th>
<th>Original Class Label</th>
<th>#MP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal (N)</td>
<td>1, 8</td>
<td>40</td>
</tr>
<tr>
<td>Damage (E)</td>
<td>3, 4, 5, 6</td>
<td>31</td>
</tr>
<tr>
<td>Middle (M)</td>
<td>2, 10</td>
<td>26</td>
</tr>
</tbody>
</table>

We did classification experiments on the new classes. The experiments also takes pieces-fold design, but not all pieces are selected for test. From each class, we randomly select 10 pieces for model test one by one. The parameter settings and results are shown in Table 6.4, Table 6.5 and the confusion matrix and feature importance ranks of distinguishing on {N, E} are shown in Figure 6.13, Figure 6.14, Figure 6.15, Figure 6.16; the positive label to draw ROC curve is class N.

Table 6.4: Classification results on distinguishing {N, E}

<table>
<thead>
<tr>
<th>P</th>
<th>Precision</th>
<th>Precision</th>
<th>Recall</th>
<th>Recall</th>
<th>F1_N</th>
<th>F1_E</th>
<th>ACC__</th>
<th>ACC_</th>
<th>ROC</th>
<th>F1__</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_N</td>
<td>_E</td>
<td>_N</td>
<td>_E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>0.658</td>
<td>0.761</td>
<td>0.820</td>
<td>0.573</td>
<td>0.730</td>
<td>0.654</td>
<td>0.700</td>
<td>0.700</td>
<td>0.778</td>
<td>0.692</td>
</tr>
<tr>
<td>b</td>
<td>0.656</td>
<td>0.740</td>
<td>0.796</td>
<td>0.583</td>
<td>0.719</td>
<td>0.652</td>
<td>0.700</td>
<td>0.700</td>
<td>0.765</td>
<td>0.686</td>
</tr>
<tr>
<td>c</td>
<td>0.736</td>
<td>0.840</td>
<td>0.869</td>
<td>0.688</td>
<td>0.797</td>
<td>0.756</td>
<td>0.850</td>
<td>0.850</td>
<td>0.861</td>
<td>0.776</td>
</tr>
<tr>
<td>d</td>
<td>0.717</td>
<td>0.748</td>
<td>0.765</td>
<td>0.698</td>
<td>0.740</td>
<td>0.722</td>
<td>0.800</td>
<td>0.800</td>
<td>0.789</td>
<td>0.731</td>
</tr>
</tbody>
</table>

Other parameter settings: the number of trees is 20, the tail size we pick is 3000, the min_samples_split is 30, and the min_weighted_frac_leaf is 0.

50
CHAPTER 6. CLASSIFICATION OF MAINTENANCE TYPES

Table 6.5: Classification results on distinguishing \{N, M, E\}

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>Precision</th>
<th>Precision</th>
<th>Precision</th>
<th>Recall</th>
<th>Recall</th>
<th>Recall</th>
<th>F1_N</th>
<th>F1_M</th>
<th>F1_E</th>
<th>ACC_unw</th>
<th>ACC_w</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N</td>
<td>0.535</td>
<td>0.339</td>
<td>0.460</td>
<td>0.725</td>
<td>0.241</td>
<td>0.429</td>
<td>0.616</td>
<td>0.282</td>
<td>0.444</td>
<td>0.533</td>
<td>0.533</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>0.504</td>
<td>0.461</td>
<td>0.418</td>
<td>0.721</td>
<td>0.305</td>
<td>0.378</td>
<td>0.593</td>
<td>0.307</td>
<td>0.397</td>
<td>0.433</td>
<td>0.433</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>0.645</td>
<td>0.343</td>
<td>0.575</td>
<td>0.744</td>
<td>0.288</td>
<td>0.460</td>
<td>0.691</td>
<td>0.313</td>
<td>0.458</td>
<td>0.567</td>
<td>0.600</td>
</tr>
<tr>
<td></td>
<td>d</td>
<td>0.610</td>
<td>0.364</td>
<td>0.383</td>
<td>0.689</td>
<td>0.305</td>
<td>0.394</td>
<td>0.647</td>
<td>0.332</td>
<td>0.388</td>
<td>0.533</td>
<td>0.533</td>
</tr>
</tbody>
</table>

Other parameter settings: the number of trees is 20, the tail size we pick is 3000, the min_samples_split is 30, and the min_weighted_frac_leaf is 0.

We give the following analysis on the results:

- From Table 6.4, we see when we apply classification on data subset \{N, E\}, the precision on class E is higher than 70%, but the recall is lower than 70%.
  Higher precision indicates a smaller number of False Positives, which means we can catch (predict) most of the positive cases; a low recall indicates many False Negatives, which means we miss many real Positive cases. As we discussed in Section 6.3.3, suppose we want to predict damage only if very confidence, then we have more strict requirements on precision than recall; suppose we want to avoid missing too many damage cases (avoid false negatives), then we have more strict requirements on recall than precision. In our experiments, the precision on class E is higher than 70%, which indicates in the dataset \{N, E\}, when the model predict a test piece as abnormal (class E), then we could say it has 70% possibilities to be a real abnormal piece; the recall on class E is lower than 70%, which indicates in the dataset \{N, E\}, the model could catch only less than 70% abnormal running pieces (class E).
  We see the performance is not satisfying, and one reason could be the limited samples in the current datasets. In the future, we can collect larger datasets, apply other parameter settings, and try different classification algorithms to see if we can improve the model.

- From Table 6.5, we see the middle class (M) is still not distinguishable from the normal cases and the abnormal cases (class M get low F1 scores, low recall as well as low precision); the class E also get low precision and recall scores. This indicates that the class M (2 + 10) might have common patterns with both class N and class E.

- By observing the feature importance rank and comparing with the rank from former experiments on distinguishing class 3 with class 8, we find the offset type of features still play most important roles and the moving derivative type features occupy more percentage.
  In addition, from Table 6.4, the feature subset c, and d perform a bit better than subset a and subset b. The subset c is subset a plus C_1; the subset d is subset b plus C_1. In the feature importance ranks, we find the C_1 rank highest in Figure 6.15 and Figure 6.16.
Figure 6.13: Results on \{N, E\}, with feature subset = a, n_estimators = 20, pick tail size = 3000, min_samples_split =30, min_weighted_frac_leaf = 0
CHAPTER 6. CLASSIFICATION OF MAINTENANCE TYPES

(a) Confusion Matrix

(b) Confusion Matrix on Rest Classes

(c) Feature Importance

(d) ROC

Figure 6.14: Results on \{N, E\}, with feature subset = b, n_estimators = 20, pick tail size = 3000, min_samples_split = 30, min_weighted_frac_leaf = 0
CHAPTER 6. CLASSIFICATION OF MAINTENANCE TYPES

Figure 6.15: Results on \{N, E\}, with feature subset = c, \textit{n estimators} = 20, pick tail size = 3000, \textit{min samples split} = 30, \textit{min weighted frac leaf} = 0
CHAPTER 6. CLASSIFICATION OF MAINTENANCE TYPES

Figure 6.16: Results on \{N, E\}, with feature subset = d, n_estimators = 20, pick tail size = 3000, min_samples_split = 30, min_weighted_frac_leaf = 0

Analyzing Machine Data for Predictive Maintenance of Electro Chemical Machining Electrodes
Chapter 7

RUL Prediction on Tool Status

In this chapter, we give the RUL estimation model of TIPs based on random forest regression algorithm and then conclude the results.

7.1 RUL Estimation Model Building

To predict tool performance, one proposal is building regression models to estimate the tool status for each TIP, which can describe the relationships between the tooling status (degradation), process behavior, and products quality behavior. However, in the current use case, the tool status is defined based on the number of caps each TIP has produced, which means the target variable (RC, or RUL) decreases linearly along the timeline (with stable production rate). In other words, from the Formula 4.2 and 4.1, the RC/RUL are linearly decreasing with the counter ($C_2$). In addition, as we explained in Section 2.1.2, the RC/RUL does not have the ground truth based on the current maintenance strategy. Thus, the remaining useful life prediction on TIPs is a challenging task.

With current datasets, we choose RC or RUL (calculated with Formula 4.2 and 4.1) as the target variables. In Chapter 4 and Chapter 5, we have conducted some data explorations like correcting $C_2$, counting life-length for each history TIP, filling missing data, and feature subsets selection. With those processed data, we built regression models with random forest algorithm. A good prediction model expects small residuals (e.g., with small Mean Absolute Error (MAE) and small Root Mean Squared Error (RMSE)).

7.2 Results

The experiments design is tips-fold cross validation, which is similar to the pieces-fold cross validation as in Chapter 6. Except for the basic feature subsets we get in Section 5.4, we do experiments with/without the product counter ($C_2$) as one predictor variable.

From the experiments, we find that when we involve the counter ($C_2$) in our predictors, the $C_2$ counts more than 90% of the feature importance rank, which means the other features (e.g., process features) do not influence the results at all. When we do not involve the counter, the prediction results are quite worse. These results offer evidence that the prediction on RC/RUL is not practical in the current stage. If we want to estimate how long one tool could still run, we’d better have a reference that represents the real tool degradation over time.
Chapter 8

Conclusions

8.1 Contributions

Our main contribution is to give solutions on the maintenance types classification problem. Additionally, we explore the Remaining Useful Life estimation topic on the shaving electrodes, but the results are limited since we do not have a reference response variable to describe tool status. In this chapter, we conclude the contributions of our classification models on maintenance types, the limitations on RUL estimation, the recommendations for our implementations, and the related future work on the predictive maintenance topic under Philips Shaver Production Plant use case.

8.1.1 Feature Information Extraction

The predictor variables we apply in the classification models are extracted from original attributes after the data visualization and thoroughly analysis on the data sources (see attributes information extraction in Chapter 5). The feature extraction plays quite an important role in the classification models improvements and solves the challenges of calibration drift problem on the original attributes.

8.1.2 Classifications on Maintenance Types

In Chapter 6, we define 10 classes of the maintenance types and do classification on interesting subsets to see the relations among the 10 classes. With all the experiments, we finally combine all maintenance types into 3 classes and give prediction results on the whole datasets.

Solutions to data issues, and empirical findings from classifications experiments

- We give solutions to calibration problem of process features from different machines: instead of using the original data, we calculate features offsets (in Section 5) on each maintenance pieces, and makes each machine’s data comparable.

- We found the following groups (classes subsets from the 10 classes) in Section 6.4.1: the \{3, 8\}, \{1, 10\}, \{4, 8\}, and \{1, 4\} are distinguishable which have high F1 score in all the classification models we set. This indicates that samples in class 8 can be used to distinguish damage maintenance (samples in class 3) or small damage repairs (samples in class 4). In addition, the class 1 and class 10 are distinguishable.

- We found the moving average smooth could improve the performance of the classifiers: in Section 5.2.2 we apply the moving average method on the feature offsets; in Section 6.4.3, we show results of controlled experiments with/without the moving average processing, and these experiments offer evidences that applying moving average on our feature (with proper sliding window size) could improve the classification performance.
CHAPTER 8. CONCLUSIONS

- Pick reasonable tail from each piece: in Section 6.3.1, we give assumptions that tail parts of pieces could be more reliable in training classification models; in Section 6.4.4, we have controlled experiments whose results show that pick reasonable tails (e.g., 4000, 5000 in the distinguishing on \{3, 8\}) could perform better than directly using all the data points in each piece.

- Feature subsets comparison: in Section 6.4.2, we found that after the grid search tests (72 experiments with each feature subsets) on \{3, 8\}, the 5 predictors groups do not have significant performance differences. They all show promising results. The feature subset b has the highest F1 score in average, and feature subset c has highest AUC score in average.

Limitations on the prediction of all maintenance types

As we explained when we define the 10 classes (see Table 6.2), there are some classes (maintenance types) that only have seldom maintenance pieces (less than 5); besides, there might be some classes have common patterns (e.g., class 3 and class 4 we found from experiments); thus, we do not perform the 10-classes classification, but perform classifications on classes subsets.

Based on the results we get from experiments in Section 6.4.1 and Section 6.4.5, we give maintenance types combination: we combine the 10 classes into 3 large classes (N, M, E) and do classification tests. In the classification experiments on data subset \{N, E\}, with feature subsets c or d, the precisions on class E (the abnormal class) are higher than 70%, and the recalls on class E are higher than 65%. However, in the classification on the whole dataset - \{N, M, E\}, the performance are worse and it could rely on two reasons: lack of samples in each class, unclear maintenance records in class M.

We suggest shaving electrode maintenance center add detailed diagnosis on some maintenance records. For example, for the maintenance records like the quality/process errors in class M, we do not know the real discarding reason on TIPs: the trigger is one kind of error, but the operation is only “discarding since reaching limitation on production numbers”, so we do not know if there exists any damage on the TIP. If we could get a diagnosis result of the TIP, then we can arrange the pieces in class M to correct class - normal running or potential damage.

8.1.3 Remaining Useful Life Estimation on ECM Tool (TIP)

We build the estimation models on remaining useful life of the TIP; however, the results show that with the target variable RC/RUL which are calculated based on production counters, we could not get good estimation results. This also confirms that it is not reasonable to apply the production counter, which is roughly linearly increasing with time, to represent the tool status.

8.2 Recommendations

We give the follwing recomendations for the project:

**Apply quality prediction results as one input for our classifiers**

As we mentioned, some researchers in Philips are working on quality predictions (with process features), so we can use their results (quality features) to replace the quality features in each feature subsets and see if there are performance improvements on the classifiers.

**Try other trend extraction strategies, try adaptive derivative**

In Section 5.2.1, we tried HHT as the trend extraction method but did not apply it on the final predictions because of the component selection problem. However, from figures 5.7, we can see that those frequency domain transform method (e.g., HHT) could give general smooth trends. Due to the time constraint, we did not try other methods. If we could solve the component selection problem on HHT, or we could find other applicable trend extraction strategies, then we can apply adaptive derivative (replace the moving derivative we use) on process features, as one part of the predictors.
8.3 Future work - Predict Machine Failures

Machine failures are usually system failures caused by processes issues. Data logs have attribute E_code to record process error code in each history product record. It is interesting to find relationships between error code (machine failures) and process trends. The aim for process error prediction is on revealing rules that lead to process errors. In addition, tool wear could sometimes lead to abnormal process circumstances which could cause machine failures. Thus, if we have the tool degradation data (e.g., by scheduling geometric parameters checking of tools on one pilot production line), then it could be interesting to build tool status estimation model, from which we could arrange timely inspections on tools and reduce possible machine failures in turn, in this case.
Bibliography


[30] Barakat, Nahla, and Andrew P. Bradley. “Rule extraction from support vector machines: measuring the explanation capability using the area under the ROC curve.” 18th International Conference on Pattern Recognition (ICPR’06). Vol. 2. IEEE, 2006. 41
Appendix A

Results of experiments in Chapter 6

A.1 Examples of Best Results we Get on Classification of Group \{3, 8\}

![Confusion Matrix and ROC Curve](image)

Figure A.1: Results on \{3, 8\}, with feature subset = a, n_estimators = 10, pick tail size = 5000, min_samples_split = 10, min_weighted_frac_leaf = 0
APPENDIX A. RESULTS OF EXPERIMENTS IN CHAPTER 6

(a) Confusion Matrix on Rest Classes

(b) ROC

Figure A.2: Results on \{3, 8\}, with feature subset = b, n_estimators = 10, pick tail size = 5000, min_samples_split = 20, min_weighted_frac_leaf = 0

(a) Confusion Matrix on Rest Classes

(b) ROC

Figure A.3: Results on \{3, 8\}, with feature subset = c, n_estimators = 10, pick tail size = 5000, min_samples_split = 20, min_weighted_frac_leaf = 0

(a) Confusion Matrix on Rest Classes

(b) ROC

Figure A.4: Results on \{3, 8\}, with feature subset = d, n_estimators = 10, pick tail size = 5000, min_samples_split = 10, min_weighted_frac_leaf = 0
APPENDIX A. RESULTS OF EXPERIMENTS IN CHAPTER 6

(a) Confusion Matrix on Rest Classes

(b) ROC

Figure A.5: Results on \{3, 8\}, with feature subset = e, n_estimators = 3, pick tail size = 4000, min_samples_split = 5, min_weighted_frac_leaf = 0
A.2 Detailed Results of All Experiments in Chapter 6

For all Tables that show experiments results, we have unified abbreviations for headers, “P” means “feature subset (group)”, “n” means the number of decision trees in each model, “mss” means “min_samples_split”, “mwfl” means “min_weighted_frac_leaf”, “ACC_unw” means “Accuracy based on unweighted voting”, “ACC_w” means “Accuracy based on weighted voting”, “ROC_area” means the “AUC of ROC”, “F1_ave” means the average F1 score of all classes in each model.

Table A.1: Classification results on interesting groups \{1, 3\}, \{1, 4\}, \{1, 8\}, \{1, 10\}, \{3, 10\}, \{4, 10\}

<table>
<thead>
<tr>
<th>P</th>
<th>n</th>
<th>Tail</th>
<th>m</th>
<th>mw</th>
<th>Precis</th>
<th>Precis</th>
<th>Recall</th>
<th>Recall</th>
<th>F1_1</th>
<th>F1_3</th>
<th>ACC_unw</th>
<th>ACC_w</th>
<th>ROC</th>
<th>F1_ave</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>n</td>
<td>n</td>
<td>0.806</td>
<td>0.848</td>
<td>0.933</td>
<td>0.136</td>
<td>0.874</td>
<td>0.296</td>
<td>0.450</td>
<td>0.450</td>
<td>0.294</td>
<td>0.555</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>10</td>
<td>0.816</td>
<td>0.816</td>
<td>0.917</td>
<td>0.226</td>
<td>0.888</td>
<td>0.342</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>20</td>
<td>0.816</td>
<td>0.816</td>
<td>0.917</td>
<td>0.226</td>
<td>0.888</td>
<td>0.342</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>30</td>
<td>0.816</td>
<td>0.816</td>
<td>0.917</td>
<td>0.226</td>
<td>0.888</td>
<td>0.342</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>40</td>
<td>0.816</td>
<td>0.816</td>
<td>0.917</td>
<td>0.226</td>
<td>0.888</td>
<td>0.342</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
</tr>
</tbody>
</table>

Table A.2: Classification results on interesting group \{3, 4\}

<table>
<thead>
<tr>
<th>P</th>
<th>n</th>
<th>Tail</th>
<th>m</th>
<th>mw</th>
<th>Precis</th>
<th>Precis</th>
<th>Recall</th>
<th>Recall</th>
<th>F1_1</th>
<th>F1_3</th>
<th>ACC_unw</th>
<th>ACC_w</th>
<th>ROC</th>
<th>F1_ave</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>n</td>
<td>n</td>
<td>0.685</td>
<td>0.768</td>
<td>0.717</td>
<td>0.475</td>
<td>0.731</td>
<td>0.644</td>
<td>0.644</td>
<td>0.655</td>
<td>0.628</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>10</td>
<td>0.725</td>
<td>0.819</td>
<td>0.797</td>
<td>0.506</td>
<td>0.771</td>
<td>0.661</td>
<td>0.661</td>
<td>0.661</td>
<td>0.627</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>20</td>
<td>0.725</td>
<td>0.819</td>
<td>0.797</td>
<td>0.506</td>
<td>0.771</td>
<td>0.661</td>
<td>0.661</td>
<td>0.661</td>
<td>0.627</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>30</td>
<td>0.725</td>
<td>0.819</td>
<td>0.797</td>
<td>0.506</td>
<td>0.771</td>
<td>0.661</td>
<td>0.661</td>
<td>0.661</td>
<td>0.627</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>40</td>
<td>0.725</td>
<td>0.819</td>
<td>0.797</td>
<td>0.506</td>
<td>0.771</td>
<td>0.661</td>
<td>0.661</td>
<td>0.661</td>
<td>0.627</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### APPENDIX A. RESULTS OF EXPERIMENTS IN CHAPTER 6

Table A.3: Classification results on interesting group \{3, 8\}

<table>
<thead>
<tr>
<th>P</th>
<th>n</th>
<th>Tail</th>
<th>m</th>
<th>mw</th>
<th>Precis</th>
<th>Precis</th>
<th>Recall</th>
<th>Recall</th>
<th>F1_3</th>
<th>F1_8</th>
<th>ACC_3</th>
<th>ACC_8</th>
<th>ROC_F1</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>5</td>
<td>0</td>
<td>0.724</td>
<td>0.667</td>
<td>0.756</td>
<td>0.629</td>
<td>0.74</td>
<td>0.648</td>
<td>0.857</td>
<td>0.851</td>
<td>0.725</td>
<td>0.694</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>20</td>
<td>0</td>
<td>0.745</td>
<td>0.717</td>
<td>0.802</td>
<td>0.743</td>
<td>0.75</td>
<td>0.684</td>
<td>0.857</td>
<td>0.875</td>
<td>0.742</td>
<td>0.729</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>30</td>
<td>0</td>
<td>0.785</td>
<td>0.695</td>
<td>0.781</td>
<td>0.641</td>
<td>0.758</td>
<td>0.667</td>
<td>0.802</td>
<td>0.802</td>
<td>0.723</td>
<td>0.712</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>40</td>
<td>0</td>
<td>0.745</td>
<td>0.67</td>
<td>0.749</td>
<td>0.679</td>
<td>0.744</td>
<td>0.715</td>
<td>0.746</td>
<td>0.747</td>
<td>0.717</td>
<td>0.709</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>4000</td>
<td>5</td>
<td>0</td>
<td>0.742</td>
<td>0.715</td>
<td>0.801</td>
<td>0.642</td>
<td>0.771</td>
<td>0.697</td>
<td>0.786</td>
<td>0.786</td>
<td>0.766</td>
<td>0.772</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>4000</td>
<td>10</td>
<td>0</td>
<td>0.742</td>
<td>0.807</td>
<td>0.888</td>
<td>0.809</td>
<td>0.809</td>
<td>0.786</td>
<td>0.786</td>
<td>0.779</td>
<td>0.786</td>
<td>0.75</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>4000</td>
<td>20</td>
<td>0</td>
<td>0.786</td>
<td>0.834</td>
<td>0.905</td>
<td>0.822</td>
<td>0.712</td>
<td>0.786</td>
<td>0.786</td>
<td>0.786</td>
<td>0.786</td>
<td>0.767</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>4000</td>
<td>30</td>
<td>0</td>
<td>0.75</td>
<td>0.831</td>
<td>0.963</td>
<td>0.819</td>
<td>0.705</td>
<td>0.786</td>
<td>0.786</td>
<td>0.786</td>
<td>0.786</td>
<td>0.762</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>4000</td>
<td>40</td>
<td>0</td>
<td>0.768</td>
<td>0.821</td>
<td>0.964</td>
<td>0.754</td>
<td>0.726</td>
<td>0.786</td>
<td>0.786</td>
<td>0.798</td>
<td>0.786</td>
<td>0.77</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>5000</td>
<td>30</td>
<td>0</td>
<td>0.75</td>
<td>0.854</td>
<td>0.92</td>
<td>0.765</td>
<td>0.728</td>
<td>0.786</td>
<td>0.786</td>
<td>0.829</td>
<td>0.786</td>
<td>0.829</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>5000</td>
<td>40</td>
<td>0</td>
<td>0.729</td>
<td>0.825</td>
<td>0.907</td>
<td>0.808</td>
<td>0.672</td>
<td>0.786</td>
<td>0.786</td>
<td>0.84</td>
<td>0.84</td>
<td>0.74</td>
</tr>
</tbody>
</table>

Table A.4: Classification results on interesting group \{4, 8\}

<table>
<thead>
<tr>
<th>P</th>
<th>n</th>
<th>Tail</th>
<th>m</th>
<th>mw</th>
<th>Precis</th>
<th>Precis</th>
<th>Recall</th>
<th>Recall</th>
<th>F1_4</th>
<th>F1_8</th>
<th>ACC_4</th>
<th>ACC_8</th>
<th>ROC_F1</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>5</td>
<td>0</td>
<td>0.774</td>
<td>0.536</td>
<td>0.912</td>
<td>0.276</td>
<td>0.837</td>
<td>0.364</td>
<td>0.545</td>
<td>0.545</td>
<td>0.655</td>
<td>0.601</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>10</td>
<td>0</td>
<td>0.772</td>
<td>0.543</td>
<td>0.918</td>
<td>0.264</td>
<td>0.839</td>
<td>0.355</td>
<td>0.617</td>
<td>0.617</td>
<td>0.681</td>
<td>0.597</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>20</td>
<td>0</td>
<td>0.781</td>
<td>0.59</td>
<td>0.923</td>
<td>0.298</td>
<td>0.846</td>
<td>0.396</td>
<td>0.744</td>
<td>0.744</td>
<td>0.694</td>
<td>0.621</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>30</td>
<td>0</td>
<td>0.788</td>
<td>0.642</td>
<td>0.936</td>
<td>0.329</td>
<td>0.854</td>
<td>0.433</td>
<td>0.617</td>
<td>0.617</td>
<td>0.682</td>
<td>0.644</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>3000</td>
<td>40</td>
<td>0</td>
<td>0.782</td>
<td>0.597</td>
<td>0.926</td>
<td>0.298</td>
<td>0.848</td>
<td>0.397</td>
<td>0.617</td>
<td>0.617</td>
<td>0.711</td>
<td>0.622</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>4000</td>
<td>5</td>
<td>0</td>
<td>0.794</td>
<td>0.649</td>
<td>0.927</td>
<td>0.344</td>
<td>0.855</td>
<td>0.446</td>
<td>0.688</td>
<td>0.688</td>
<td>0.743</td>
<td>0.651</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>4000</td>
<td>10</td>
<td>0</td>
<td>0.805</td>
<td>0.74</td>
<td>0.952</td>
<td>0.374</td>
<td>0.872</td>
<td>0.497</td>
<td>0.744</td>
<td>0.744</td>
<td>0.787</td>
<td>0.685</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>4000</td>
<td>20</td>
<td>0</td>
<td>0.809</td>
<td>0.702</td>
<td>0.937</td>
<td>0.34</td>
<td>0.809</td>
<td>0.51</td>
<td>0.688</td>
<td>0.688</td>
<td>0.744</td>
<td>0.689</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>4000</td>
<td>30</td>
<td>0</td>
<td>0.802</td>
<td>0.647</td>
<td>0.923</td>
<td>0.381</td>
<td>0.858</td>
<td>0.479</td>
<td>0.617</td>
<td>0.617</td>
<td>0.77</td>
<td>0.669</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
<td>4000</td>
<td>40</td>
<td>0</td>
<td>0.801</td>
<td>0.679</td>
<td>0.932</td>
<td>0.421</td>
<td>0.867</td>
<td>0.525</td>
<td>0.759</td>
<td>0.759</td>
<td>0.741</td>
<td>0.696</td>
</tr>
</tbody>
</table>

Table A.5: Classification results on interesting group \{1, 3, 4\}

<table>
<thead>
<tr>
<th>m</th>
<th>Precis</th>
<th>Precis</th>
<th>Precis</th>
<th>Recall</th>
<th>Recall</th>
<th>Recall</th>
<th>Recall</th>
<th>F1_1</th>
<th>F1_3</th>
<th>F1_4</th>
<th>ACC_1</th>
<th>ACC_3</th>
<th>ACC_4</th>
<th>ROC_F1</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.652</td>
<td>0.279</td>
<td>0.455</td>
<td>0.704</td>
<td>0.144</td>
<td>0.401</td>
<td>0.719</td>
<td>0.181</td>
<td>0.426</td>
<td>0.426</td>
<td>0.426</td>
<td>0.426</td>
<td>0.426</td>
<td>0.426</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.62</td>
<td>0.266</td>
<td>0.409</td>
<td>0.751</td>
<td>0.126</td>
<td>0.353</td>
<td>0.68</td>
<td>0.171</td>
<td>0.375</td>
<td>0.364</td>
<td>0.381</td>
<td>0.408</td>
<td>0.408</td>
<td>0.408</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.62</td>
<td>0.253</td>
<td>0.424</td>
<td>0.754</td>
<td>0.147</td>
<td>0.366</td>
<td>0.686</td>
<td>0.189</td>
<td>0.393</td>
<td>0.381</td>
<td>0.371</td>
<td>0.423</td>
<td>0.423</td>
<td>0.423</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>0.64</td>
<td>0.256</td>
<td>0.416</td>
<td>0.768</td>
<td>0.134</td>
<td>0.351</td>
<td>0.698</td>
<td>0.157</td>
<td>0.398</td>
<td>0.398</td>
<td>0.408</td>
<td>0.418</td>
<td>0.418</td>
<td>0.418</td>
<td></td>
</tr>
</tbody>
</table>

Other parameter settings: the feature subset is b, the number of trees is 10, the tail size we pick is 3000, and the min_weighted_frac_leaf is 0.
## APPENDIX A. RESULTS OF EXPERIMENTS IN CHAPTER 6

### Table A.6: Classification results on interesting group \{3, 8\}, with/without moving average

<table>
<thead>
<tr>
<th>P</th>
<th>Tau</th>
<th>m</th>
<th>ss</th>
<th>Prec</th>
<th>Reca</th>
<th>F1</th>
<th>ACC</th>
<th>ROC</th>
<th>F1</th>
<th>ACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3000</td>
<td>5</td>
<td>0</td>
<td>0.661</td>
<td>0.594</td>
<td>0.723</td>
<td>0.529</td>
<td>0.690</td>
<td>0.558</td>
<td>0.675</td>
</tr>
<tr>
<td>3</td>
<td>3000</td>
<td>5</td>
<td>0.0002</td>
<td>0.653</td>
<td>0.570</td>
<td>0.689</td>
<td>0.529</td>
<td>0.670</td>
<td>0.549</td>
<td>0.675</td>
</tr>
<tr>
<td>3</td>
<td>3000</td>
<td>10</td>
<td>0</td>
<td>0.666</td>
<td>0.582</td>
<td>0.690</td>
<td>0.556</td>
<td>0.678</td>
<td>0.569</td>
<td>0.675</td>
</tr>
<tr>
<td>3</td>
<td>3000</td>
<td>10</td>
<td>0.0004</td>
<td>0.671</td>
<td>0.575</td>
<td>0.667</td>
<td>0.580</td>
<td>0.669</td>
<td>0.578</td>
<td>0.619</td>
</tr>
<tr>
<td>3</td>
<td>3000</td>
<td>20</td>
<td>0</td>
<td>0.691</td>
<td>0.539</td>
<td>0.610</td>
<td>0.538</td>
<td>0.601</td>
<td>0.548</td>
<td>0.672</td>
</tr>
<tr>
<td>3</td>
<td>3000</td>
<td>20</td>
<td>0.0008</td>
<td>0.676</td>
<td>0.546</td>
<td>0.662</td>
<td>0.592</td>
<td>0.669</td>
<td>0.584</td>
<td>0.663</td>
</tr>
<tr>
<td>4</td>
<td>3000</td>
<td>5</td>
<td>0</td>
<td>0.645</td>
<td>0.619</td>
<td>0.787</td>
<td>0.444</td>
<td>0.709</td>
<td>0.517</td>
<td>0.659</td>
</tr>
<tr>
<td>4</td>
<td>3000</td>
<td>5</td>
<td>0.0002</td>
<td>0.696</td>
<td>0.651</td>
<td>0.761</td>
<td>0.572</td>
<td>0.727</td>
<td>0.609</td>
<td>0.675</td>
</tr>
<tr>
<td>4</td>
<td>3000</td>
<td>10</td>
<td>0</td>
<td>0.652</td>
<td>0.630</td>
<td>0.803</td>
<td>0.449</td>
<td>0.719</td>
<td>0.527</td>
<td>0.730</td>
</tr>
<tr>
<td>4</td>
<td>3000</td>
<td>10</td>
<td>0.0004</td>
<td>0.684</td>
<td>0.650</td>
<td>0.774</td>
<td>0.540</td>
<td>0.726</td>
<td>0.590</td>
<td>0.675</td>
</tr>
<tr>
<td>5</td>
<td>3000</td>
<td>5</td>
<td>0</td>
<td>0.672</td>
<td>0.622</td>
<td>0.749</td>
<td>0.530</td>
<td>0.708</td>
<td>0.572</td>
<td>0.675</td>
</tr>
<tr>
<td>5</td>
<td>3000</td>
<td>5</td>
<td>0.0008</td>
<td>0.672</td>
<td>0.667</td>
<td>0.765</td>
<td>0.629</td>
<td>0.640</td>
<td>0.648</td>
<td>0.557</td>
</tr>
<tr>
<td>5</td>
<td>3000</td>
<td>10</td>
<td>0</td>
<td>0.747</td>
<td>0.722</td>
<td>0.806</td>
<td>0.649</td>
<td>0.775</td>
<td>0.683</td>
<td>0.857</td>
</tr>
<tr>
<td>5</td>
<td>3000</td>
<td>10</td>
<td>0.0004</td>
<td>0.742</td>
<td>0.709</td>
<td>0.771</td>
<td>0.624</td>
<td>0.741</td>
<td>0.668</td>
<td>0.802</td>
</tr>
<tr>
<td>5</td>
<td>3000</td>
<td>20</td>
<td>0</td>
<td>0.735</td>
<td>0.760</td>
<td>0.811</td>
<td>0.624</td>
<td>0.771</td>
<td>0.668</td>
<td>0.802</td>
</tr>
<tr>
<td>6</td>
<td>3000</td>
<td>5</td>
<td>0</td>
<td>0.694</td>
<td>0.725</td>
<td>0.761</td>
<td>0.592</td>
<td>0.672</td>
<td>0.562</td>
<td>0.675</td>
</tr>
<tr>
<td>6</td>
<td>3000</td>
<td>5</td>
<td>0.0004</td>
<td>0.742</td>
<td>0.741</td>
<td>0.824</td>
<td>0.742</td>
<td>0.648</td>
<td>0.675</td>
<td>0.725</td>
</tr>
<tr>
<td>6</td>
<td>3000</td>
<td>10</td>
<td>0</td>
<td>0.742</td>
<td>0.715</td>
<td>0.801</td>
<td>0.612</td>
<td>0.740</td>
<td>0.727</td>
<td>0.766</td>
</tr>
<tr>
<td>6</td>
<td>3000</td>
<td>10</td>
<td>0.0004</td>
<td>0.758</td>
<td>0.745</td>
<td>0.824</td>
<td>0.662</td>
<td>0.790</td>
<td>0.701</td>
<td>0.786</td>
</tr>
<tr>
<td>6</td>
<td>3000</td>
<td>20</td>
<td>0</td>
<td>0.742</td>
<td>0.807</td>
<td>0.884</td>
<td>0.604</td>
<td>0.809</td>
<td>0.730</td>
<td>0.786</td>
</tr>
<tr>
<td>6</td>
<td>3000</td>
<td>20</td>
<td>0.0008</td>
<td>0.738</td>
<td>0.767</td>
<td>0.836</td>
<td>0.669</td>
<td>0.794</td>
<td>0.769</td>
<td>0.786</td>
</tr>
<tr>
<td>7</td>
<td>3000</td>
<td>5</td>
<td>0</td>
<td>0.754</td>
<td>0.835</td>
<td>0.905</td>
<td>0.620</td>
<td>0.822</td>
<td>0.712</td>
<td>0.786</td>
</tr>
<tr>
<td>7</td>
<td>3000</td>
<td>5</td>
<td>0.0008</td>
<td>0.753</td>
<td>0.784</td>
<td>0.864</td>
<td>0.635</td>
<td>0.804</td>
<td>0.702</td>
<td>0.786</td>
</tr>
<tr>
<td>7</td>
<td>3000</td>
<td>10</td>
<td>0</td>
<td>0.801</td>
<td>0.756</td>
<td>0.815</td>
<td>0.748</td>
<td>0.780</td>
<td>0.748</td>
<td>0.786</td>
</tr>
<tr>
<td>7</td>
<td>3000</td>
<td>10</td>
<td>0.0002</td>
<td>0.793</td>
<td>0.774</td>
<td>0.837</td>
<td>0.720</td>
<td>0.814</td>
<td>0.746</td>
<td>0.857</td>
</tr>
<tr>
<td>7</td>
<td>3000</td>
<td>20</td>
<td>0</td>
<td>0.825</td>
<td>0.818</td>
<td>0.868</td>
<td>0.763</td>
<td>0.846</td>
<td>0.790</td>
<td>0.873</td>
</tr>
<tr>
<td>7</td>
<td>3000</td>
<td>20</td>
<td>0.0004</td>
<td>0.796</td>
<td>0.784</td>
<td>0.838</td>
<td>0.795</td>
<td>0.824</td>
<td>0.774</td>
<td>0.873</td>
</tr>
<tr>
<td>8</td>
<td>3000</td>
<td>5</td>
<td>0</td>
<td>0.815</td>
<td>0.779</td>
<td>0.833</td>
<td>0.751</td>
<td>0.824</td>
<td>0.768</td>
<td>0.929</td>
</tr>
<tr>
<td>8</td>
<td>3000</td>
<td>5</td>
<td>0.0008</td>
<td>0.801</td>
<td>0.766</td>
<td>0.824</td>
<td>0.738</td>
<td>0.812</td>
<td>0.751</td>
<td>0.929</td>
</tr>
<tr>
<td>8</td>
<td>3000</td>
<td>10</td>
<td>0</td>
<td>0.799</td>
<td>0.812</td>
<td>0.871</td>
<td>0.719</td>
<td>0.833</td>
<td>0.763</td>
<td>0.857</td>
</tr>
<tr>
<td>8</td>
<td>3000</td>
<td>10</td>
<td>0.0004</td>
<td>0.801</td>
<td>0.791</td>
<td>0.859</td>
<td>0.732</td>
<td>0.826</td>
<td>0.760</td>
<td>0.857</td>
</tr>
<tr>
<td>8</td>
<td>3000</td>
<td>20</td>
<td>0</td>
<td>0.770</td>
<td>0.739</td>
<td>0.811</td>
<td>0.788</td>
<td>0.790</td>
<td>0.713</td>
<td>0.786</td>
</tr>
<tr>
<td>8</td>
<td>3000</td>
<td>20</td>
<td>0.0008</td>
<td>0.783</td>
<td>0.794</td>
<td>0.860</td>
<td>0.693</td>
<td>0.820</td>
<td>0.740</td>
<td>0.786</td>
</tr>
<tr>
<td>9</td>
<td>3000</td>
<td>5</td>
<td>0</td>
<td>0.775</td>
<td>0.854</td>
<td>0.912</td>
<td>0.660</td>
<td>0.808</td>
<td>0.745</td>
<td>0.857</td>
</tr>
<tr>
<td>9</td>
<td>3000</td>
<td>5</td>
<td>0.0008</td>
<td>0.804</td>
<td>0.818</td>
<td>0.874</td>
<td>0.811</td>
<td>0.796</td>
<td>0.857</td>
<td></td>
</tr>
</tbody>
</table>

Other parameter settings: the number of trees is 10.
APPENDIX A. RESULTS OF EXPERIMENTS IN CHAPTER 6

Table A.7: Top classification results on interesting group \{3, 8\}, when choose different predictors

<table>
<thead>
<tr>
<th>( n )</th>
<th>( m )</th>
<th>( m )</th>
<th>( \text{Precis} )</th>
<th>( \text{Precis} )</th>
<th>( \text{Recall} )</th>
<th>( \text{Recall} )</th>
<th>( \text{P1}_{\text{m}} )</th>
<th>( \text{P1}_{\text{m}} )</th>
<th>( \text{ACC}_{\text{m}} )</th>
<th>( \text{ACC}_{\text{m}} )</th>
<th>( \text{ROC} )</th>
<th>( \text{ROC} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>10</td>
<td>1000</td>
<td>0.000100446</td>
<td>0.778</td>
<td>0.835</td>
<td>0.937</td>
<td>0.667</td>
<td>0.833</td>
<td>0.743</td>
<td>0.786</td>
<td>0.786</td>
<td>0.816</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>8.04E-05</td>
<td>0.778</td>
<td>0.835</td>
<td>0.937</td>
<td>0.667</td>
<td>0.833</td>
<td>0.743</td>
<td>0.786</td>
<td>0.786</td>
<td>0.816</td>
<td>0.788</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.0002</td>
<td>0.706</td>
<td>0.775</td>
<td>0.849</td>
<td>0.667</td>
<td>0.833</td>
<td>0.743</td>
<td>0.786</td>
<td>0.786</td>
<td>0.816</td>
<td>0.788</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.0002</td>
<td>0.706</td>
<td>0.775</td>
<td>0.849</td>
<td>0.667</td>
<td>0.833</td>
<td>0.743</td>
<td>0.786</td>
<td>0.786</td>
<td>0.816</td>
<td>0.788</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.0002</td>
<td>0.706</td>
<td>0.775</td>
<td>0.849</td>
<td>0.667</td>
<td>0.833</td>
<td>0.743</td>
<td>0.786</td>
<td>0.786</td>
<td>0.816</td>
<td>0.788</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.0002</td>
<td>0.706</td>
<td>0.775</td>
<td>0.849</td>
<td>0.667</td>
<td>0.833</td>
<td>0.743</td>
<td>0.786</td>
<td>0.786</td>
<td>0.816</td>
<td>0.788</td>
</tr>
</tbody>
</table>

Table A.8: Classification results on interesting group \{1, 3\}, apply sampling to make balance dataset

| \( n \) | \( m \) | \( \text{Precis} \) | \( \text{Precis} \) | \( \text{Recall} \) | \( \text{Recall} \) | \( \text{P1}_{\text{m}} \) | \( \text{P1}_{\text{m}} \) | \( \text{ACC}_{\text{m}} \) | \( \text{ACC}_{\text{m}} \) | \( \text{ROC} \) | \( \text{ROC} \) | \( \text{ave} \) | \( \text{ave} \) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 10 | 0.000100446 | 0.778 | 0.835 | 0.937 | 0.667 | 0.833 | 0.743 | 0.786 | 0.786 | 0.816 | 0.788 |
| 10 | 10 | 8.04E-05 | 0.778 | 0.835 | 0.937 | 0.667 | 0.833 | 0.743 | 0.786 | 0.786 | 0.816 | 0.788 |
| 3 | 3 | 0.0002 | 0.706 | 0.775 | 0.849 | 0.667 | 0.833 | 0.743 | 0.786 | 0.786 | 0.816 | 0.788 |
| 10 | 3 | 0.0002 | 0.706 | 0.775 | 0.849 | 0.667 | 0.833 | 0.743 | 0.786 | 0.786 | 0.816 | 0.788 |
| 3 | 5 | 0.0002 | 0.706 | 0.775 | 0.849 | 0.667 | 0.833 | 0.743 | 0.786 | 0.786 | 0.816 | 0.788 |
| 10 | 5 | 0.0002 | 0.706 | 0.775 | 0.849 | 0.667 | 0.833 | 0.743 | 0.786 | 0.786 | 0.816 | 0.788 |

Other parameter settings: the feature subset is \( b \), the number of trees is 10, the tail size we pick is 3000, and the \( \text{min}_\text{weight}_\text{frac}_\text{leaf} \) is 0.
APPENDIX A. RESULTS OF EXPERIMENTS IN CHAPTER 6

Table A.9: Classification results on interesting group \{3, 8\}, when picking different tail size

<table>
<thead>
<tr>
<th>P</th>
<th>n</th>
<th>Tail</th>
<th>m</th>
<th>Precis</th>
<th>Precis</th>
<th>Recall</th>
<th>Recall</th>
<th>F1</th>
<th>F1</th>
<th>ACC</th>
<th>ACC</th>
<th>ROC</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Analyzing Machine Data for Predictive Maintenance of Electro Chemical Machining Electrodes 69
## APPENDIX A. RESULTS OF EXPERIMENTS IN CHAPTER 6

**Table A.10: Classification results of experiments in Section 6.4.2 on interesting group \( \{3, 8\} \)**

<table>
<thead>
<tr>
<th>( k )</th>
<th>( n )</th>
<th>Test</th>
<th>( m )</th>
<th>( n_{in} )</th>
<th>( n_{out} )</th>
<th>( \text{Prec} )</th>
<th>( \text{Pres} )</th>
<th>( \text{Recall} )</th>
<th>( \text{Recall} )</th>
<th>( F_{1} )</th>
<th>( F_{1} )</th>
<th>( \text{ACC.} )</th>
<th>( \text{ACC.} )</th>
<th>( \text{RUS.} )</th>
<th>( \text{RUS.} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
<td>0.499</td>
<td>0.666</td>
</tr>
</tbody>
</table>

---

**Electro Chemical Machining Electrodes**

---

**Analyzing Machine Data for Predictive Maintenance of Electro Chemical Machining Electrodes**

---

70
## APPENDIX A. RESULTS OF EXPERIMENTS IN CHAPTER 6

<table>
<thead>
<tr>
<th>P</th>
<th>n</th>
<th>Taxl m</th>
<th>mow</th>
<th>Precise</th>
<th>Precise</th>
<th>Recall</th>
<th>Recall</th>
<th>P1,2</th>
<th>P1,2</th>
<th>ACCm</th>
<th>ACCm</th>
<th>ROC</th>
<th>P1,2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>0.0002</td>
<td>0.752</td>
<td>0.753</td>
<td>0.754</td>
<td>0.755</td>
<td>0.756</td>
<td>0.757</td>
<td>0.758</td>
<td>0.759</td>
<td>0.760</td>
<td>0.761</td>
<td>0.762</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>0.0004</td>
<td>0.747</td>
<td>0.748</td>
<td>0.749</td>
<td>0.750</td>
<td>0.751</td>
<td>0.752</td>
<td>0.753</td>
<td>0.754</td>
<td>0.755</td>
<td>0.756</td>
<td>0.757</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>0.01</td>
<td>0.740</td>
<td>0.741</td>
<td>0.742</td>
<td>0.743</td>
<td>0.744</td>
<td>0.745</td>
<td>0.746</td>
<td>0.747</td>
<td>0.748</td>
<td>0.749</td>
<td>0.750</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>0.0001</td>
<td>0.737</td>
<td>0.738</td>
<td>0.739</td>
<td>0.740</td>
<td>0.741</td>
<td>0.742</td>
<td>0.743</td>
<td>0.744</td>
<td>0.745</td>
<td>0.746</td>
<td>0.747</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>0.0008</td>
<td>0.736</td>
<td>0.737</td>
<td>0.738</td>
<td>0.739</td>
<td>0.740</td>
<td>0.741</td>
<td>0.742</td>
<td>0.743</td>
<td>0.744</td>
<td>0.745</td>
<td>0.746</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>0.0002</td>
<td>0.733</td>
<td>0.734</td>
<td>0.735</td>
<td>0.736</td>
<td>0.737</td>
<td>0.738</td>
<td>0.739</td>
<td>0.740</td>
<td>0.741</td>
<td>0.742</td>
<td>0.743</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>0.0008</td>
<td>0.728</td>
<td>0.729</td>
<td>0.730</td>
<td>0.731</td>
<td>0.732</td>
<td>0.733</td>
<td>0.734</td>
<td>0.735</td>
<td>0.736</td>
<td>0.737</td>
<td>0.738</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>0.0002</td>
<td>0.724</td>
<td>0.725</td>
<td>0.726</td>
<td>0.727</td>
<td>0.728</td>
<td>0.729</td>
<td>0.730</td>
<td>0.731</td>
<td>0.732</td>
<td>0.733</td>
<td>0.734</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>0.0008</td>
<td>0.721</td>
<td>0.722</td>
<td>0.723</td>
<td>0.724</td>
<td>0.725</td>
<td>0.726</td>
<td>0.727</td>
<td>0.728</td>
<td>0.729</td>
<td>0.730</td>
<td>0.731</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>0.0004</td>
<td>0.717</td>
<td>0.718</td>
<td>0.719</td>
<td>0.720</td>
<td>0.721</td>
<td>0.722</td>
<td>0.723</td>
<td>0.724</td>
<td>0.725</td>
<td>0.726</td>
<td>0.727</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>0.0002</td>
<td>0.711</td>
<td>0.712</td>
<td>0.713</td>
<td>0.714</td>
<td>0.715</td>
<td>0.716</td>
<td>0.717</td>
<td>0.718</td>
<td>0.719</td>
<td>0.720</td>
<td>0.721</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>0.0002</td>
<td>0.706</td>
<td>0.707</td>
<td>0.708</td>
<td>0.709</td>
<td>0.710</td>
<td>0.711</td>
<td>0.712</td>
<td>0.713</td>
<td>0.714</td>
<td>0.715</td>
<td>0.716</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>0.0008</td>
<td>0.701</td>
<td>0.702</td>
<td>0.703</td>
<td>0.704</td>
<td>0.705</td>
<td>0.706</td>
<td>0.707</td>
<td>0.708</td>
<td>0.709</td>
<td>0.710</td>
<td>0.711</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>0.0004</td>
<td>0.697</td>
<td>0.698</td>
<td>0.699</td>
<td>0.700</td>
<td>0.701</td>
<td>0.702</td>
<td>0.703</td>
<td>0.704</td>
<td>0.705</td>
<td>0.706</td>
<td>0.707</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>0.0002</td>
<td>0.691</td>
<td>0.692</td>
<td>0.693</td>
<td>0.694</td>
<td>0.695</td>
<td>0.696</td>
<td>0.697</td>
<td>0.698</td>
<td>0.699</td>
<td>0.700</td>
<td>0.701</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>0.0002</td>
<td>0.687</td>
<td>0.688</td>
<td>0.689</td>
<td>0.690</td>
<td>0.691</td>
<td>0.692</td>
<td>0.693</td>
<td>0.694</td>
<td>0.695</td>
<td>0.696</td>
<td>0.697</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>0.0008</td>
<td>0.682</td>
<td>0.683</td>
<td>0.684</td>
<td>0.685</td>
<td>0.686</td>
<td>0.687</td>
<td>0.688</td>
<td>0.689</td>
<td>0.690</td>
<td>0.691</td>
<td>0.692</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>0.0004</td>
<td>0.678</td>
<td>0.679</td>
<td>0.680</td>
<td>0.681</td>
<td>0.682</td>
<td>0.683</td>
<td>0.684</td>
<td>0.685</td>
<td>0.686</td>
<td>0.687</td>
<td>0.688</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>0.0002</td>
<td>0.672</td>
<td>0.673</td>
<td>0.674</td>
<td>0.675</td>
<td>0.676</td>
<td>0.677</td>
<td>0.678</td>
<td>0.679</td>
<td>0.680</td>
<td>0.681</td>
<td>0.682</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
<td>0.0002</td>
<td>0.667</td>
<td>0.668</td>
<td>0.669</td>
<td>0.670</td>
<td>0.671</td>
<td>0.672</td>
<td>0.673</td>
<td>0.674</td>
<td>0.675</td>
<td>0.676</td>
<td>0.677</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
<td>0.0008</td>
<td>0.662</td>
<td>0.663</td>
<td>0.664</td>
<td>0.665</td>
<td>0.666</td>
<td>0.667</td>
<td>0.668</td>
<td>0.669</td>
<td>0.670</td>
<td>0.671</td>
<td>0.672</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
<td>0.0004</td>
<td>0.658</td>
<td>0.659</td>
<td>0.660</td>
<td>0.661</td>
<td>0.662</td>
<td>0.663</td>
<td>0.664</td>
<td>0.665</td>
<td>0.666</td>
<td>0.667</td>
<td>0.668</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
<td>0.0002</td>
<td>0.652</td>
<td>0.653</td>
<td>0.654</td>
<td>0.655</td>
<td>0.656</td>
<td>0.657</td>
<td>0.658</td>
<td>0.659</td>
<td>0.660</td>
<td>0.661</td>
<td>0.662</td>
</tr>
</tbody>
</table>

### Table A.11: Classification Results of Experiments in Section 6.4.2 on Interesting Group \{3, 8\}

Analyzing Machine Data for Predictive Maintenance of Electro Chemical Machining Electrodes
Table A.12: Classification Results of Experiments in Section 6.4.2 on Interesting Group \{3, 8\}

| c  | 2000 | 5  | 0.000201853 | 0.711 | 0.611 | 0.682 | 0.663 | 0.696 | 0.627 | 0.602 | 0.602 | 0.685 | 0.661 |
| c  | 3  | 2000 | 5  | 0.00001 | 0.744 | 0.669 | 0.814 | 0.699 | 0.818 | 0.746 | 0.762 | 0.762 | 0.797 | 0.799 |
| c  | 3  | 2000 | 10 | 0.00001 | 0.744 | 0.669 | 0.814 | 0.699 | 0.818 | 0.746 | 0.762 | 0.762 | 0.797 | 0.799 |
| c  | 3  | 3000 | 10 | 0.00001 | 0.744 | 0.669 | 0.814 | 0.699 | 0.818 | 0.746 | 0.762 | 0.762 | 0.797 | 0.799 |
| c  | 3  | 3000 | 5  | 0.00001 | 0.744 | 0.669 | 0.814 | 0.699 | 0.818 | 0.746 | 0.762 | 0.762 | 0.797 | 0.799 |
| c  | 3  | 2000 | 5  | 0.00001 | 0.744 | 0.669 | 0.814 | 0.699 | 0.818 | 0.746 | 0.762 | 0.762 | 0.797 | 0.799 |
| c  | 3  | 2000 | 10 | 0.00001 | 0.744 | 0.669 | 0.814 | 0.699 | 0.818 | 0.746 | 0.762 | 0.762 | 0.797 | 0.799 |
| c  | 3  | 2000 | 5  | 0.00001 | 0.744 | 0.669 | 0.814 | 0.699 | 0.818 | 0.746 | 0.762 | 0.762 | 0.797 | 0.799 |
| c  | 3  | 2000 | 10 | 0.00001 | 0.744 | 0.669 | 0.814 | 0.699 | 0.818 | 0.746 | 0.762 | 0.762 | 0.797 | 0.799 |
| c  | 3  | 2000 | 5  | 0.00001 | 0.744 | 0.669 | 0.814 | 0.699 | 0.818 | 0.746 | 0.762 | 0.762 | 0.797 | 0.799 |
| c  | 3  | 2000 | 10 | 0.00001 | 0.744 | 0.669 | 0.814 | 0.699 | 0.818 | 0.746 | 0.762 | 0.762 | 0.797 | 0.799 |

APPENDIX A. RESULTS OF EXPERIMENTS IN CHAPTER 6

Analyzing Machine Data for Predictive Maintenance of Electro Chemical Machining Electrodes
### APPENDIX A: RESULTS OF EXPERIMENTS IN CHAPTER 6

**Table A.13: Classification Results of Experiments in Section 6.4.2 on Interesting Group \{3, 8\}**

| F | n | tail | d 1 | d 2 | d 3 | d 4 | d 5 | d 6 | d 7 | d 8 | d 9 | d 10 | d 11 | d 12 | d 13 | d 14 | d 15 | d 16 | d 17 | d 18 | d 19 | d 20 | d 21 | d 22 | d 23 | d 24 | d 25 | d 26 | d 27 | d 28 | d 29 | d 30 |
|---|---|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 3 | 1 | 0.001 | 0.002 | 0.003 | 0.004 | 0.005 | 0.006 | 0.007 | 0.008 | 0.009 | 0.010 | 0.011 | 0.012 | 0.013 | 0.014 | 0.015 | 0.016 | 0.017 | 0.018 | 0.019 | 0.020 | 0.021 | 0.022 | 0.023 | 0.024 | 0.025 | 0.026 | 0.027 |
| 3 | 2 | 0.002 | 0.003 | 0.004 | 0.005 | 0.006 | 0.007 | 0.008 | 0.009 | 0.010 | 0.011 | 0.012 | 0.013 | 0.014 | 0.015 | 0.016 | 0.017 | 0.018 | 0.019 | 0.020 | 0.021 | 0.022 | 0.023 | 0.024 | 0.025 | 0.026 | 0.027 | 0.028 |
| 3 | 3 | 0.003 | 0.004 | 0.005 | 0.006 | 0.007 | 0.008 | 0.009 | 0.010 | 0.011 | 0.012 | 0.013 | 0.014 | 0.015 | 0.016 | 0.017 | 0.018 | 0.019 | 0.020 | 0.021 | 0.022 | 0.023 | 0.024 | 0.025 | 0.026 | 0.027 | 0.028 | 0.029 |
| 3 | 4 | 0.004 | 0.005 | 0.006 | 0.007 | 0.008 | 0.009 | 0.010 | 0.011 | 0.012 | 0.013 | 0.014 | 0.015 | 0.016 | 0.017 | 0.018 | 0.019 | 0.020 | 0.021 | 0.022 | 0.023 | 0.024 | 0.025 | 0.026 | 0.027 | 0.028 | 0.029 | 0.030 |
| 3 | 5 | 0.005 | 0.006 | 0.007 | 0.008 | 0.009 | 0.010 | 0.011 | 0.012 | 0.013 | 0.014 | 0.015 | 0.016 | 0.017 | 0.018 | 0.019 | 0.020 | 0.021 | 0.022 | 0.023 | 0.024 | 0.025 | 0.026 | 0.027 | 0.028 | 0.029 | 0.030 | 0.031 |
| 3 | 6 | 0.006 | 0.007 | 0.008 | 0.009 | 0.010 | 0.011 | 0.012 | 0.013 | 0.014 | 0.015 | 0.016 | 0.017 | 0.018 | 0.019 | 0.020 | 0.021 | 0.022 | 0.023 | 0.024 | 0.025 | 0.026 | 0.027 | 0.028 | 0.029 | 0.030 | 0.031 | 0.032 |
| 3 | 7 | 0.007 | 0.008 | 0.009 | 0.010 | 0.011 | 0.012 | 0.013 | 0.014 | 0.015 | 0.016 | 0.017 | 0.018 | 0.019 | 0.020 | 0.021 | 0.022 | 0.023 | 0.024 | 0.025 | 0.026 | 0.027 | 0.028 | 0.029 | 0.030 | 0.031 | 0.032 | 0.033 |
| 3 | 8 | 0.008 | 0.009 | 0.010 | 0.011 | 0.012 | 0.013 | 0.014 | 0.015 | 0.016 | 0.017 | 0.018 | 0.019 | 0.020 | 0.021 | 0.022 | 0.023 | 0.024 | 0.025 | 0.026 | 0.027 | 0.028 | 0.029 | 0.030 | 0.031 | 0.032 | 0.033 | 0.034 |

**Analyzing Machine Data for Predictive Maintenance of**

**Electro Chemical Machining Electrodes**

73
Table A.14: Classification Results of Experiments in Section 6.4.2 on Interesting Group \{3, 8\}

<table>
<thead>
<tr>
<th>n</th>
<th>Test</th>
<th>(n = 5)</th>
<th>(n = 10)</th>
<th>(n = 20)</th>
<th>(n = 3)</th>
<th>(n = 30)</th>
<th>(n = 2000)</th>
<th>(n = 3000)</th>
<th>(n = 4000)</th>
<th>(n = 5000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>3</td>
<td>0.743</td>
<td>0.754</td>
<td>0.753</td>
<td>0.759</td>
<td>0.766</td>
<td>0.766</td>
<td>0.777</td>
<td>0.783</td>
<td>0.782</td>
</tr>
<tr>
<td>4</td>
<td>0.748</td>
<td>0.758</td>
<td>0.763</td>
<td>0.770</td>
<td>0.777</td>
<td>0.779</td>
<td>0.787</td>
<td>0.790</td>
<td>0.800</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.753</td>
<td>0.763</td>
<td>0.765</td>
<td>0.771</td>
<td>0.778</td>
<td>0.780</td>
<td>0.787</td>
<td>0.790</td>
<td>0.800</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.758</td>
<td>0.767</td>
<td>0.771</td>
<td>0.777</td>
<td>0.784</td>
<td>0.786</td>
<td>0.792</td>
<td>0.795</td>
<td>0.800</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.763</td>
<td>0.772</td>
<td>0.775</td>
<td>0.781</td>
<td>0.788</td>
<td>0.790</td>
<td>0.796</td>
<td>0.799</td>
<td>0.800</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.768</td>
<td>0.777</td>
<td>0.780</td>
<td>0.786</td>
<td>0.793</td>
<td>0.795</td>
<td>0.801</td>
<td>0.804</td>
<td>0.800</td>
<td></td>
</tr>
</tbody>
</table>

APPENDIX A. RESULTS OF EXPERIMENTS IN CHAPTER 6

Electro Chemical Machining Electrodes

Analyzing Machine Data for Predictive Maintenance of Electro Chemical Machining Electrodes