MASTER

Data driven decision support for failure prediction

Ganesan, A.

Award date:
2012

Link to publication

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
Data Driven Decision Support for Failure Prediction

Master Thesis

A.Ganesan

Supervisor:
dr. M. Pechenizkiy

Eindhoven, July 2012
Abstract

Failure prediction and diagnosis is one of the principal areas in reliability engineering and system safety. Recent developments in the field of data mining opened new research orientation to failure prediction and diagnosis. In this thesis, the feasibility of data-driven decision support for failure prediction of the TWINSCAN machines of ASML is studied. Data-driven approaches focus on learning and classifying occurring faults from historical data without assuming a priori model ahead of time. Machine learning techniques such as classification, for instance are used to predict failures for which the dependent variables are unknown. One of the main tasks is to employ data mining technology and existing predictive modelling techniques on existing ASML event logs generated by TWINSCAN machines. The problems addressed in this project are related to several areas studied in data mining, including but not limited to mining data streams, predictive analytics, time-series prediction.

An ensemble data-driven decision support system is proposed for predictive maintenance of ASML TWINSCAN machines. Several machine learning and statistical techniques are compared in terms of their predictive ability to find the best models. An ensemble decision support system is proposed by incorporating the best models based on the findings. The model encompasses existing and novel fault prediction techniques to predict future machine failures. The results showed that machine characteristics are instantaneous and the machine do not show early signs of failures. The results obtained by pattern mining suggests that failures could be predicted at least 15 minutes before their occurrence by employing rules with a confidence value of at least 0.25. The classification results showed that certain failures could be predicted on an average with a precision greater than 0.75 and a recall value of 0.39 with a prediction time that ranges from 1 hour to 24 hours before their occurrence.

Keywords: Failure Prediction and Diagnostics, TWINSCAN, Ensemble System, Decision Support System, Predictive Data Mining.
Preface

This thesis is a result of the graduation project as part of the curriculum for the degree MSc in Computer Science and Engineering at TU/e. The project was carried out within the D&H group of Mathematics and Computer Science department of TU/e in collaboration with D&E Applications Group of ASML. The project duration was from February 2012 to July 2012 at ASML, Netherlands B.V.

I would like to acknowledge all who supported me during this period. In particular, I would like to thank my supervisor from TU/e, Prof. Mykola Pechenizkiy, for his assistance, valuable comments and advice. His assistance helped me improve my work immensely. He has been kind enough to find time and accept my innumerable meeting requests.

I would like to thank Pieter Knelissen, my supervisor from ASML for his valuable inputs and guidance. His knowledge and domain expertise helped me in many ways. I extend my sincere gratitude to Durk van der Ploeg for giving me the opportunity to work on this project. I would also like to thank members of the Development and Engineering, Applications Group Department at ASML for their knowledge sharing.

I would like to thank all my friends in Eindhoven who kept my recreational quotient alive during this period. At this juncture, I would like to remember and thank two most important people who travelled along. My father, for he has been a great inspiration all through my life and is not only the driver behind my wheels but the wheels itself. My mother, for her perpetual encouragement and motivation.

Ajith Ganesan
July 2012
# Contents

List of Figures v

List of Tables ix

List of Acronyms xi

1 Introduction 1

1.1 Thesis Context 1

1.2 Problem Description 5

1.3 Business Motivation 6

1.4 Research Questions 7

1.5 Research Approach 9

1.6 Results 12

2 Data Understanding 15

2.1 Log File Data Structure 16

2.2 Data Preprocessing 17

2.3 Message Code Analysis 19

2.4 Failure Points Analysis 22

2.5 Outliers Analysis 24

2.6 Raw Log Data Analysis 25

2.7 Conclusion 26

3 Preliminaries and Related Works 27

3.1 Fault Prediction Systems 27

3.2 Employed Strategies 30

3.2.1 Pattern Mining 31

3.2.2 Classification 40

3.2.3 Event Frequency Statistical Modelling 44

3.3 Related Existing Works 47
4 Evaluation of Techniques
   4.1 Pattern Mining Evaluation .................................. 54
   4.2 Classification Evaluation ................................... 55
   4.3 Event Frequency Statistical Modelling Evaluation ........ 57
   4.4 Evaluation metrics in operational settings ............... 58

5 Experiments
   5.1 Pattern Mining ................................................. 61
      5.1.1 Rule Mining ................................................. 61
      5.1.2 Batch Rule Mining ......................................... 71
      5.1.3 Comparison of RM and BRM ................................. 75
      5.1.4 Rules State Space .......................................... 76
   5.2 Classification ................................................ 79
      5.2.1 Frequency Based Classification .......................... 79
      5.2.2 Rules Based Classification ................................ 89
   5.3 Event Frequency Statistical Modelling ....................... 92
   5.4 Decision Support System ..................................... 93
   5.5 Conclusion .................................................... 96

6 Conclusions
   6.1 Main Contributions and Results ............................... 99
   6.2 Implication .................................................... 100
   6.3 Limitations and Future Works ................................. 101

References .......................................................... 105

Appendix A Log File Analysis .................................... 106
Appendix B Rule Mining ............................................. 110
Appendix C Batch Rule Mining ..................................... 113
Appendix D Classification .......................................... 117
Appendix E Rules State Space ..................................... 128
# List of Figures

1.1 TWINSCAN Machine .............................................. 2
1.2 Example Instance - Log data event stream ....................... 3
1.3 Example Instance of Strategies - Failure Prediction .......... 3
1.4 Example Instance - Log data event stream ....................... 4
1.5 Project Framework .................................................. 6
1.6 Example Instance - Predictive ability over time ............... 8
1.7 Example Instance - Distinct characteristics of root causes .... 8
1.8 Example Instance - Failure characteristic changes .......... 9
1.9 Example Instance - Ensemble Model .............................. 10
1.10 Project Life Cycle .................................................. 11

2.1 Data Exploration Phase ............................................. 16
2.2 Frequency of message codes per day ............................. 20
2.3 Stacked column - Frequency of total message codes per day ... 20
2.4 Stacked percentage column - Distribution based on message types ........................................... 21
2.5 Boxplot - Distribution based on message types ............... 22
2.6 Frequency of failures ............................................... 23
2.7 Distribution of failures .............................................. 23
2.8 Analysed Log Data - Distribution of Events .................... 24

3.1 Model Development Phase ......................................... 28
3.2 Failure Prediction - Example Instance 1 ......................... 29
3.3 Failure Prediction - Example Instance 2 ........................ 30
3.4 Failure Prediction - Example Instance 3 ........................ 30
3.5 Rule Mining .......................................................... 32
3.6 Example Instance - Rule Mining ................................... 34
3.7 Example Instance - Single Event(E) Stream ...................... 35
3.8 Example Instance - Batched Single Event(E) Stream .......... 36
3.9 Example Instance - Event Steam ................................... 37
3.10 Example Instance - Batch Rule Mining ........................... 37
<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.11</td>
<td>Example Instance - Batch Rule Mining Limitation</td>
<td>38</td>
</tr>
<tr>
<td>3.12</td>
<td>Example Instance - Rules State Space</td>
<td>39</td>
</tr>
<tr>
<td>3.13</td>
<td>Classification - Failure next window</td>
<td>43</td>
</tr>
<tr>
<td>3.14</td>
<td>Classification - Failure one window earlier</td>
<td>43</td>
</tr>
<tr>
<td>3.15</td>
<td>Feature Instance Compression</td>
<td>44</td>
</tr>
<tr>
<td>3.16</td>
<td>Example Instance - Event Frequency Statistical Modelling</td>
<td>45</td>
</tr>
<tr>
<td>3.17</td>
<td>Calculation of IAT</td>
<td>45</td>
</tr>
<tr>
<td>3.18</td>
<td>Example Instance - Increasing Frequency of Events</td>
<td>46</td>
</tr>
<tr>
<td>3.19</td>
<td>Employed Strategies with respective features</td>
<td>47</td>
</tr>
<tr>
<td>4.1</td>
<td>Example Instance - Calculation of TP, FP, TN and FN in streaming settings</td>
<td>57</td>
</tr>
<tr>
<td>5.1</td>
<td>Example Instance - Support, Recall relation in RM</td>
<td>63</td>
</tr>
<tr>
<td>5.2</td>
<td>RM - Number of rules obtained for each failure for varied lead time (\Delta l_t) values with a constant time window, (\Delta t_w) of 180 minutes, support level, (s) of 0.10 and confidence, (c) of 0.25.</td>
<td>63</td>
</tr>
<tr>
<td>5.3</td>
<td>RM - Frequency of rules generated across confidence levels obtained for each failure for a lead time, (\Delta l_t) of 5 minutes, time window, (\Delta t_w) of 180 minutes, support, (s) of 0.10 and a confidence, (c) of 0.25.</td>
<td>64</td>
</tr>
<tr>
<td>5.4</td>
<td>RM - Frequency of rules generated across confidence levels obtained for each failure for a lead time, (\Delta l_t) of 15 minutes, time window, (\Delta t_w) of 180 minutes, support, (s) of 0.10 and a confidence, (c) of 0.25.</td>
<td>65</td>
</tr>
<tr>
<td>5.5</td>
<td>RM - Frequency of MTTF of rules grouped to respective bins obtained for each failure for a lead time, (\Delta l_t) of 5 minutes, time window, (\Delta t_w) of 180 minutes, support, (s) of 0.10 and a confidence, (c) of 0.25.</td>
<td>67</td>
</tr>
<tr>
<td>5.6</td>
<td>Failure Prediction using RM under operational settings</td>
<td>67</td>
</tr>
<tr>
<td>5.7</td>
<td>Improving Recall values in RM</td>
<td>70</td>
</tr>
<tr>
<td>5.8</td>
<td>BRM - Comparison of number of rules obtained for each failure for varied lead time, (\Delta l_t) with a constant time window, (\Delta t_w) of 180 minutes, support level, (s) of 0.10, confidence, (c_r) of 0.50, distance separation (\Delta d) of 24 hour and (c_b) of 0.25.</td>
<td>72</td>
</tr>
<tr>
<td>5.9</td>
<td>BRM - Frequency of rules generated across batch confidence levels, (c_b) obtained for each failure for a (\Delta l_t) of 5 minutes, (\Delta t_w) of 180 minutes, support, (s) of 0.10, fixed confidence, (c_r) of 0.50, distance separation (\Delta d) of 24 hour and fixed (c_b) of 0.25.</td>
<td>73</td>
</tr>
</tbody>
</table>
5.10 BRM - Frequency of rules generated across confidence levels, 
$c_b$ obtained for each failure for a $\Delta l t$ of 15 minutes, $\Delta t w$ of 180 minutes, support, $s$ of 0.10, confidence, $c_r$ of 0.50, distance separation $\Delta d$ of 24 hour and $c_b$ of 0.25. .......................... 74
5.11 Distribution of Failure Probability for a state space generated
for each failure with a base $\Delta l t$ of 5 minutes .............................. 78
5.12 Classification Phases ................................................................. 80
5.13 Failure prediction using classifying models at operational site . 83
5.14 False Alarms Acceptable Trade-off ............................... 84
5.15 Average Precision, Recall for positive class of top three classifiers with 12 Hour time window for failure prediction of $tw_{i+2}$ performed with original data. ................................. 86
5.16 Average Precision, Recall for positive class of top three classifiers with 1 Hour time window for failure prediction of $tw_{i+2}$ performed with original data. ................................. 87
5.17 Average Precision, Recall of top three classifiers with 1 Hour time window for failure prediction of $tw_{i+1}$ performed with original data. ................................. 88
5.18 Failures with comparative better precision and recall for positive classes across all parameters. Average of top three classifiers per length $l$ value is shown. ................................. 88
5.19 Average Precision, Recall of top three classifiers performed with original data for any parameter setting. ................................. 89
5.20 Average Precision, Recall of top three classifiers with 24 Hour time window for failure prediction of $tw_{i+1}$ performed for compression factor of 1 ............................. 90
5.21 Average Precision, Recall of top three classifiers with 24 Hour time window for failure prediction of $tw_{i+1}$ performed for compression factor of 2 ............................. 91
5.22 Decision Support System proposal ................................. 94
A.1 Example Instance - Log File .......................................................... 106
B.1 RM - Frequency of rules generated across confidence levels
obtained for each failure for a $\Delta l t$ of 8 minutes, $\Delta t w$ of 180 minutes, support, $s$ of 0.10 and a confidence, $c$ of 0.25. .......... 110
B.2 RM - Frequency of rules generated across confidence levels
obtained for each failure for a $\Delta l t$ of 10 minutes, $\Delta t w$ of 180 minutes, support, $s$ of 0.10 and a confidence, $c$ of 0.25. .......... 111
LIST OF FIGURES

B.3 RM - Frequency of rules generated across confidence levels obtained for each failure for a $\Delta lt$ of 12 minutes, $\Delta tw$ of 180 minutes, support, $s$ of 0.10 and a confidence, $c$ of 0.25. . . . . 111

B.4 RM - Frequency of rules generated across confidence levels obtained for each failure for a $\Delta lt$ of 30 minutes, $\Delta tw$ of 180 minutes, support, $s$ of 0.10 and a confidence, $c$ of 0.25. . . . . 112

C.1 BRM - Frequency of rules generated across confidence levels obtained for each failure for a $\Delta lt$ of 8 minutes, $\Delta tw$ of 180 minutes, support, $s$ of 0.10 , confidence, $c_r$ of 0.50, distance separation $\Delta d$ of 24 hour and $c_b$ of 0.25. . . . . . . . . . . . . 113

C.2 BRM - Frequency of rules generated across confidence levels obtained for each failure for a $\Delta lt$ of 10 minutes, $\Delta tw$ of 180 minutes, support, $s$ of 0.10 , confidence, $c_r$ of 0.50, distance separation $\Delta d$ of 24 hour and $c_b$ of 0.25. . . . . . . . . . . . . 114

C.3 BRM - Frequency of rules generated across confidence levels obtained for each failure for a $\Delta lt$ of 12 minutes, $\Delta tw$ of 180 minutes, support, $s$ of 0.10 , confidence, $c_r$ of 0.50, distance separation $\Delta d$ of 24 hour and $c_b$ of 0.25. . . . . . . . . . . . . 114

C.4 BRM - Frequency of rules generated across confidence levels obtained for each failure for a $\Delta lt$ of 30 minutes, $\Delta tw$ of 180 minutes, support, $s$ of 0.10 , confidence, $c_r$ of 0.50, distance separation $\Delta d$ of 24 hour and $c_b$ of 0.25. . . . . . . . . . . . . 115

E.1 Example Instance - XML file generated for RSS for the failure LO-8422 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 128
# List of Tables

2.1 Log Data - Example Instance .............................. 17  
2.2 Homogeneous Records - Example Instance .................. 18  
2.3 Boxplot Statistics - Distribution based on message types .... 22  
2.4 Summary of total occurrence of message codes ............... 25  
2.5 Summary of total occurrence of message codes in raw log data 25  

4.1 Confusion Matrix ........................................ 56  
4.2 Evaluation of Techniques and their relevance in operational settings ........................................ 60  

5.1 Comparison of RM and BRM. U-Union of rules generated from RM and BRM. RM-Relative complement of BRM with respect to RM. BRM-Relative complement of RM with respect to BRM. 76  
5.2 Rules State Space generated with a base $\Delta lt$ of 5 minutes. ... 78  

A.1 Analysed Log Data Statistics after temporal compression ... 107  
A.2 Analysed Raw Log Data Statistics .......................... 108  
A.3 Summary of total occurrence of message codes per type after temporal compression ................................. 108  
A.4 Summary of total occurrence of message codes per type in raw log data ............................................ 109  

C.1 Comparison of RM and BRM. U-Union of rules generated from RM and BRM. RM-Relative complement of BRM with respect to RM. BRM-Relative complement of RM with respect to BRM. 116  

D.1 TP, FP, Precision and Recall for positive class of top three classifiers with 12 Hour time window for failure prediction of $tw_{i+2}$ performed with original data. ................................. 119  
D.2 TP, FP, Precision and Recall for positive class of top three classifiers with 1 Hour time window for failure prediction of $tw_{i+2}$ performed with original data. ................................. 121
D.3 TP, FP, Precision and Recall for positive class of top three classifiers with 1 Hour time window for failure prediction of \( tw_{i+1} \) performed with original data. 

D.4 TP, FP, Precision and Recall for positive class of top three classifiers with respective parameters for failure prediction of LO-0052 performed with original data. 

D.5 TP, FP, Precision and Recall for positive class of top three classifiers with respective parameters for failure prediction of LO-0058 performed with original data. 

D.6 TP, FP, Precision and Recall for positive class of top three classifiers with respective parameters for failure prediction of LO-848F performed with original data. 

D.7 TP, FP, Precision and Recall for positive class of top three classifiers for RBC with compression factor of 2 and prediction on \( tw_{i+2} \).
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>Anderson-Darling test</td>
</tr>
<tr>
<td>DSS</td>
<td>Decision Support System</td>
</tr>
<tr>
<td>BRM</td>
<td>Extended Rule Mining</td>
</tr>
<tr>
<td>EFSM</td>
<td>Event Frequency Statistical Modelling</td>
</tr>
<tr>
<td>FBC</td>
<td>Frequency Based Classification</td>
</tr>
<tr>
<td>FN</td>
<td>False Negatives</td>
</tr>
<tr>
<td>FP</td>
<td>False Positives</td>
</tr>
<tr>
<td>FP</td>
<td>Failure Probability of states in Rules State Space</td>
</tr>
<tr>
<td>FT</td>
<td>Functional Trees</td>
</tr>
<tr>
<td>IAT</td>
<td>Inter Arrival Times</td>
</tr>
<tr>
<td>LO</td>
<td>Lot Abortions</td>
</tr>
<tr>
<td>MTTF</td>
<td>Mean Time To Failure</td>
</tr>
<tr>
<td>MTNS</td>
<td>Mean Time to Next State</td>
</tr>
<tr>
<td>NSP</td>
<td>Next State Probability</td>
</tr>
<tr>
<td>RSS</td>
<td>Rules State Space</td>
</tr>
<tr>
<td>RBC</td>
<td>Rules Based Classification</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>RM</td>
<td>Rule Mining</td>
</tr>
<tr>
<td>TP</td>
<td>True Positives</td>
</tr>
<tr>
<td>TN</td>
<td>True Negatives</td>
</tr>
<tr>
<td>VFI</td>
<td>Voting Feature Intervals</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This thesis is a result of the graduation project as part of the curriculum for the degree MSc in Computer Science and Engineering at TU/e. The project was carried out within the D&H group of the Mathematics and Computer Science department of TU/e in collaboration with D&E Applications Group, ASML, Eindhoven.

The deployed TWINSCAN machines of ASML shown in Figure 1.1 generate log file data that includes events, warnings and errors that occur in the machines. A subset of these errors are critical and can lead to machine failures. The objective of this thesis is to study the feasibility and to facilitate data-driven decision support for failure prediction of the TWINSCAN machines of ASML.

This chapter emphasizes the aforementioned problem in detail. Section 1.1 discusses the thesis context followed by elaborate problem description in Section 1.2. Section 1.3 discusses business motivation followed by research questions in Section 1.4. The research approach and the results are outlined in Sections 1.5 and 1.6 respectively.

1.1 Thesis Context

The TWINSCAN machines (Figure 1.1) of ASML are deployed at various client locations. Considering the inherent potential associated with these machines, hours of stoppage of production could result in massive loss of money. The machines have inter dependencies between various modules to
Introduction

perform complex time critical and real time operations. Each of these machines which are in production generates log data. The log data is recorded with the list of events that occur in the machine and warning and error messages that are generated by the various modules deployed in the machine. An error message is logged by the machine when there is a machine deviation from its intended behaviour. The detailed description of log file structure and analysis among other information is given in Chapter 2.

![TWINSCAN Machine](image)

The errors encountered in the machine vary in their intensity of damage. A series of events, warnings and non-critical errors could result in a fatal critical error which could lead to the stoppage of production. The log file is also recorded with these critical errors which cause machine failures. This thesis deals with analyzing existing log data generated by TWINSCAN machines to predict the occurrence of these fatal critical errors. For instance, Figure 1.2 shows an example instance of the log data structure of 4 days. The list of events, warnings and errors that happened at various time points in the machine are marked as A, B, C, D etc. The critical errors that occurred during this period are marked as failures.

The aim of this thesis is to analyze the log file data as shown in Figure 1.2 to predict the future occurrences of failures. This is performed by proposing strategies that captures machine behaviour changes before failure occurrences and triggers failure alarm whenever the captured behaviours are observed. Typically, this is performed by analyzing machine behaviour by defining a time window of \( n \) time units before the occurrence of each failure. The time duration between the end of the time window and the failure occurrence is
Introduction

Figure 1.2: Example Instance - Log data event stream

considered as lead time and the length of the lead time is equal to $n$ time units. Figure 1.3 shows this scenario. If the machine behaviour change is encountered frequently in earlier failure occurrences, the observed behaviour is considered to be related to failure and a failure alarm is raised in the future whenever similar behaviour is encountered. Figure 1.3 explains this scenario in which a failure alarm is raised inside the time window during which the observed machine behaviour related to failure is encountered. The failure due time in the figure represents the time duration between the triggering of alarm and failure occurrence.

The proximity of the time window to the failure affects the time before which failure prediction can be made. A failure alarm is raised if the observed behaviour during the time window is encountered in the future. One of the main tasks of this project is to employ such data mining technology and existing predictive modelling techniques on event logs of TWINSCAN machines.

Figure 1.3: Example Instance of Strategies - Failure Prediction
Failure Points

To facilitate prediction of machine failures, it is important to understand the cause of machine failures. Ideally, every error logged is an aberration in machine characteristics. But in view of the fact that there are hundreds of distinct errors that have occurred in the analysed log (Section 2.3) and some of them have occurred multiple times (in the order of 100 times per day) in a very limited time, they could not have triggered machine failures. To this regard, it is paramount to find the set of critical errors that trigger machine failures.

The project focuses on the failure prediction for a single module. This module named LO is the crux of the system. It has been observed from the past by machine administrators and domain experts of ASML that errors logged from this module are critical for machine operations. Therefore module LO is a substantial component in order to keep machine in production. The characteristics and distribution of critical errors logged from LO module are detailed in Section 2.4. Each LO type error occurrence is considered as individual failure point. Domain experts opined that there are other modules that have similar characteristics to LO and thus the strategies that are used to predict the machine failures caused by LO could also be applicable to other modules.

![Figure 1.4: Example Instance - Log data event stream](image-url)

For instance, Figure 1.4 shows a basic instance of log data structure with respective time frame. It shows individual occurrences of events (A, B, etc.) at respective time frames with three critical failures of two distinct types (LO-xxx1 and LO-xxx2) happened in an interval of 2 days. The objective is to predict these individual failure points.
1.2 Problem Description

The objective of the project is to analyse, apply and experiment with existing methodologies on TWINSCAN log data to propose a decision support system to predict future machine failures. The possibility of applying novel methodologies when existing methods would not produce optimal results should also be considered. This is analogous to making time-critical detections in a continuous measurements of time series data in order to capture their time-evolving trends and patterns. The objective should be realized by considering the possibilities of all the research questions discussed in Section 1.4.

The motivation behind choosing a decision support system is the fact that the failure alarms triggered from the system have to be examined by domain experts before taking necessary action decisions. Although failure occurrence is detected in advance, the system administrator may need to diagnose the problem and figure out the best sequence of remedial actions. Therefore decisions are not completely automated and the system aims at alerting the domain experts about the possibility of future failures which then have to be scrutinized by domain experts for false positives.

Diagnosis of failure alarms involves understanding the attributes associated with failure alarms. A failure alarm raised for a certain failure with no other causal information can not afford to have many false alarms. In other words, in certain cases where failure alarms contain no associated causal information for the failure, it is as important not to raise a false failure alarm as deducting a failure. This is because, a domain expert has to manually decipher the alarm to understand whether it is a genuine alarm or a false alarm with no additional information regarding the cause of a failure. The tolerance level for the amount of false alarms vary based on whether explicit patterns or rules that include the cause of the failure is established. In the case of black box model, little or no information is attached with the failure alarm in which case, the analysis of the failure alarm takes longer time which subsequently reduces the tolerance level for the amount of false alarms raised. The precise acceptable and usable precision and recall rate for the failures is explained in Chapter 5 for each strategy.

Since the machine operations are real time, predicting failures even hours’ before it happens is a highly regarded information. The machine failure prediction should be performed as early as possible with a trade-off between the prediction accuracy and time before failure. Both in-time predictions and avoidance of false alarms are important for the organization.
this trade-off is considered as one of the research questions as explained in Section 1.4.

**Problem Statement:** To propose failure prediction strategies for a decision support system construction of TWINSCAN machine.

Figure 1.5 shows the general project framework which deals with the proposal and evaluation of specific strategies that could be used to predict machine failures. These strategies with high predictive ability for specific failures could then be incorporated in a decision support system used in the operational site to predict future machine failures.

![Project Framework Diagram](image)

**Figure 1.5: Project Framework**

### 1.3 Business Motivation

Predicting fatal errors could be constructive due to assorted number of reasons. Some of which are the following:

- Through proactive maintenance, a significant amount of machine hours spent on unscheduled downtime could be saved.
- Prevention of stoppage of production.
• Scheduling of people to dispatch (to rectify potential fault in the machine).

• “Stock” refinement of employees who rectify the faults in the machines. If the number of employees who service/repair and rectify the machine is limited and not adequate enough to service the amount of machines that are predicted to be error prone in the near future, the dispatch team could be replenished.

• In addition to lowering system performance and availability, failures can also greatly increase the system management costs. Thus by predicting them in advance and taking necessary contingency plans, management costs incurred by failures could be saved.

1.4 Research Questions

The research questions that are to be considered in realizing the objective are as follows.

• Analyze the predictive characteristics of machines over a period of time and effectiveness of failure predictability as time before prediction increases.

  Motivation: How early failure prediction could be performed depends on how early the machine characteristics change before failure occurrences. It is possible that most of the causes for machine failures are real time and that once the root causes occur, the failure happens instantaneously. It is also possible that depending on the concerned failure that machine shows early signs of characteristic changes in which case a failure could be predicted early. Figure 1.6 shows an example instance for predictive ability of failures over time. It shows two failures (F1 and F2) and the causes (represented as \(\Delta\)) for the failures. It could be understood that F1 shows no early signs of failures and the cause of the failure occurs close to F1 where as F2 shows early signs of failure. Hence, if F1 were to be predicted early, chances are that the predictive accuracy of the failure may decrease. This trade-off of how the predictive accuracy changes as we move away in time from failure occurrence should be analysed.

• Analyze whether specific strategies could be used for specific error type based on its characteristics.

  Motivation: Machine failures, by nature are caused by different critical
errors (Section 2.4). Each failure occurrence from LO module belongs to a specific failure type. The number of types of failures are detailed in Section 2.4. Failures of different types could be caused by critical errors for different reasons by different sub-modules inside LO module. Each such failure could have disjoint set of root causes. The errors which are fundamentally different with disjoint set of root causes, could also have different characteristics. Depending on the characteristic, different strategies could yield different predictive performance. Figure 1.7 shows an example instance for this scenario where two failures are caused by different reasons. Failure F1 is caused by increasing frequency of $\triangle$ whereas F2 is caused by one to one occurrence relation of X to failure. In this case, two different strategies could be used for F1 and F2 to capture these distinct behaviours to increase predictive accuracy. Hence, the possibility of specific strategies yielding better results for specific critical error should also be analyzed.

**Figure 1.6: Example Instance - Predictive ability over time**

**Figure 1.7: Example Instance - Distinct characteristics of root causes**

- Analyze the possibility of characteristic changes in machine behaviour over time.

  **Motivation:** There is a possibility that the machine characteristics could change over time. Therefore, depending on a failure, it is also possible that failure occurrence pattern could change over time. This change could be caused by varied factors such as the possibility of concept drift[17] in the evolving data. The expiration of old data depends on the machine characteristics and the existing data distribution. For instance, the set of events that cause a failure could be rectified and
eliminated by machine experts which suppresses their frequency of occurrence over time or a set of events which causes the failure could change over time. Figure 1.8 shows an example instance for this scenario where the same failure F is caused by different reasons over time.

Figure 1.8: Example Instance - Failure characteristic changes

- Analyze the possibility of an adaptive model.  
  Motivation: If the machine characteristics could change over time, chances are that the base predictive model developed for a critical error could become obsolete. This model had to be updated with recent trends in machine characteristics. Therefore it is necessary in this case for the proposed system to adapt its model to reflect recent changes in machine characteristics. This scenario has to be analyzed.

- Analyze the possibility of creating an ensemble model.  
  Motivation: In the case, when specific strategies yield better results for specific failures, it is more efficient to have multiple predictive models developed from different strategies that have better performance for different failures. Therefore, the possibility of creating an ensemble model which incorporates different models to predict the failures should also be considered. Figure 1.9 shows an example instance of an ensemble model which incorporates multiple strategies.

- In machine learning failure prediction by classification (Section 3.2.2), study suitable ways of constructing feature vector instances.

1.5 Research Approach

The scope of the project is to propose a decision support system while realizing the research questions discussed in Section 1.4. The steps involved in the project are shown in Figure 1.10.

The research in this project discusses various analysis and implementations performed in the major three areas involved in the project, Data Ex-
Introduction

Data exploration, Model Development and Decision Support System Proposal. In particular, the fault prediction systems and existing strategies to build them are discussed. These include various model-based and data-driven strategies, predictive models based on continuous time series data stream that could be employed. Previous works in predictive models and their employability is discussed. The project also focuses on feature vector construction mechanisms and parameter selection heuristics for various models. It also describes data analysis which includes outliers analysis and data cleaning among others.

Research Techniques and Outline

The research techniques and outline of the research approach are as follows.

Conduct a study on input log data set

The first step is to investigate the input log files. This is elaborated in Chapter 2 with a detailed data exploration phase where input log data is analysed, examined for outliers and also analysed for distribution of events.
Figure 1.10: Project Life Cycle

and errors. Statistics are drawn from input log data set. In addition, log file data structure, outliers and failure points analysis in the input data set are also elaborated.

Perform a literature study on relevant concepts

Chapter 3 investigates existing literature on topics such as predictive data mining, fault prediction systems, continuous data stream mining, pattern mining and their applicability. Applicability of novel strategies refined as per characteristics of events and errors encountered in the data set is examined. It also lists the employed strategies that have better relevance to this project.

Choose Evaluation Criteria and Techniques

The next step is to define the evaluation criteria and techniques. These are presented and elaborately discussed in Chapter 4 where the list of evaluation techniques for predictive models across all strategies are analysed.
Experimentation

Chapter 5 explains how the employed models discussed in chapter 3 are experimented. The results from various models are examined. Also, the interpretation of results and validating over fitting of data are discussed. The chapter concludes with a proposal of a decision support system incorporating the best models and main findings from the experimentation.

Conclude

This thesis is concluded by Chapter 6 in which the entire approach is evaluated. The main contributions and the results are discussed. The implications of the results are elaborated. This chapter also includes a discussion of future work aiming at solving existing problems or improving the results of the project.

1.6 Results

The objective of the thesis was realized by formulating several failure prediction strategies (Pattern Mining, Classification and EFSM) based on the analysis performed in data exploration phase and consideration of research questions. The data exploration of the log file data showed that there is a possibility of machine behaviour change. The number of errors logged in the machine decreased over time and the number of events increased. This could potentially mean that there is a change in machine behaviour not only in terms of errors or events that occur but also in terms of root causes that trigger the failure. It could be possible that the decreasing number of errors may or may not eliminate the possibility of certain failures from occurring. In order to analyze this possibility and to understand if there is a need for an adaptive model, sliding window strategy based on classification was performed.

It is possible that certain failures are triggered by one to one occurrence relation of an event with a failure. To capture this scenario, pattern mining strategies were defined. In certain cases, it is possible that an increasing frequency of a specific event could trigger a failure rather than one to one event to failure relation. To investigate such possibility, where increasing frequency of an event triggering a failure, EFSM was performed. In addition,
failure prediction based on classification that focus on learning and classifying occurring faults from historical data without assuming a priori model was experimented. Different feature vector construction mechanisms and how they affect prediction accuracy was analyzed.

The results of pattern mining suggested that machine operations are instantaneous and there is little or no change in machine behaviour as time to failure decreases. However, all the failures but LO-8459 (has only 33 occurrences in total) have rules generated with a confidence level greater than 0.25 for failure prediction before 15 minutes of failures. Four failures have rules generated for half hour before failure. One failure has rules generated for one hour before failure. These rules could be further examined and a failure prediction model based on these rules could be build. This possibility is elaborated and the usability of these rules under operational settings are described in Section 5.1.1. It was understood from domain experts that since machine operations are real time, even prediction 5 minutes before failure could be productive under operational settings. This is because the failure alarms are generated by triggering an alarm whenever the rules are encountered. Therefore, it is possible for a domain expert to use this information as a causal analysis for failure.

The results of classification varied much in terms of parameter settings. Results obtained for three failures were consistent in terms of precision and recall for positive class across various parameter settings such as length of time window, classifying algorithms used, etc. The results for these failures suggested that the failures are expected to be predicted in the range from 1 hour to 1 day before its occurrence. Also, overall on an average, for the three failures, the number of false alarms generated for 100 failure alarms was 25 and the number of failures that are deducted are 39 for every hundred failures. The respective classifying models of these failures could be used in operational settings since the precision rate is high and 39 failures for every hundred failures are deducted. The results of statistical fit did not yield productive results. In addition results obtained from RBC strategy of classification showed employable results in operational settings for 6 failures. In particular, LO-0058 and LO-848F have precision and recall rate of 0.82 and 0.45 on an average which corresponds to 18 false alarms per hundred failure alarms and 45 failures are detected per 100 failures.

The results of sliding window strategy showed no considerable change in machine behaviour over time. The predictive ability of certain strategies differed based on characteristics of respective failures. It was observed that an ensemble model could be created with different strategies that have better
predictive power for specific failures.
Chapter 2

Data Understanding

This chapter introduces different steps involved in data exploration phase (Figure 2.1). The business objective is followed by data exploration process. Data Understanding usually starts with analysing data requirements to achieve business objectives followed by data retrieval. In the data retrieval process, based on the identified data requirements, data from a source is retrieved. In this case, log data generated from the TWINSCAN machines are retrieved. This process is followed by data preparation, data analysis, outliers analysis and cleaning which considers the possibility of data containing out-of-range values and anomalies. Performing analysis and implementing models based on data which was not screened for such anomalies could result in misleading results, in particular in predictive data mining. Finally the data is transposed to a format that is applicable to the to-be-implemented predictive models and strategies which varies for different strategies.

Section 2.1 describes log file data structure generated from TWINSCAN machines. Section 2.2 explains data preprocessing followed by message code analysis in Section 2.3 where the list of events, warnings and errors, their occurrence frequency are analysed. Section 2.4 analyses the failure points and outliers are analysed in Section 2.5. The analysis of raw log data is performed in Section 2.6. This chapter is concluded in Section 2.7 in which the important findings of this chapter are outlined.
2.1 Log File Data Structure

The log files generated from the machines record noteworthy actions that the machine takes. Warnings and errors that happen in the machine are also recorded. Information is recorded in the machine actively as in when events happen. The machine creates a single log record every time an event, warning or an error occurs. Each record is structured and is associated with multiple attributes. The list of attributes contain information that ranges from record type to time stamp to system parameters, domain parameters and customer tuned parameters among others. The nature of the problem is to analyse and extract useful correlations between events based only on their timing characteristics without consideration of machine parameters and other customer related machine settings. For this reason, only a subset of the attributes associated with the records were considered. An example instance of a log file record is shown in Figure A.1 in Appendix A (Log File Analysis).
### Record Type

Record type takes one of the three values: **EVENT**, **ERROR** or **WARNING**.

- An **EVENT** is an individual action of a machine confined to system specifications.
- A **WARNING** is an alert to the customer but a warning by itself does not imply that the system is deviating from its specification.
- **ERROR** is an aberration in the behaviour of the system. In other words, when the system logs an error, it is not performing its intended behaviour.

### Time Stamp

Date, Time and Microseconds associated with the record.

### Message Code

A message code is a code associated with events, warnings or errors. Each distinct event, warning or an error has an unique code associated with it. The number of unique events analysed and other such statistics are shown in Table A.1 of Appendix A.

### 2.2 Data Preprocessing

Before the log data could be used to study the failure behaviour, it must be processed and filtered.

**Attribute Filtering** Each record in the log is associated with multiple attributes. In the first step of data preprocessing, these attributes were
Temporal Compression: Logging mechanism operates at much finer time granularity (e.g. the logging interval is in microsecond). Events sometimes occur in bursts, or as clusters. Some clusters are homogeneous, with their records having identical values in the record attributes. An example instance is shown in Table 2.2. The instance shows three separate records logged for the same event with the same message text with only microseconds difference in time stamp attribute.

In temporal compression, a cluster of homogeneous records with microseconds difference were coalesced into a single record. Identifying clusters of single event, involves grouping of records according to the associated time stamp, message code and a suitable threshold $T$. Records that have the same message code belong to a cluster if the gaps between successive message codes are less than $T$. In this project, the threshold value was set to 1 second. In other words, a message with the same message code when logged in the same time with accuracy of a second is coalesced into a single record. These kind of record compression filtering mechanisms and setting threshold values is also researched on [18][19][20] in which the threshold value was set for 5 minutes. The number of records analysed and other record statistics after performing temporal compression is shown in Table A.1 of Appendix A.

Advantage: In certain cases, the first occurrence of an event in a burst could trigger a failure. But this does not stop the event from occurring multiple times in the same burst at the same second and also does not gives raise to failures on successive occurrences in the same burst. When the raw data is taken as is without compression, certain strategies which performs one to one mapping of an event with a failure such as pattern mining strategies (Section 3.2.1) would not capture these type of relations due to their low confidence levels which is calculated by considering the number of mappings of an event to a failure. To eliminate this scenario, temporal compression
is performed. Domain experts acknowledge that certain events are logged multiple times due to the same reason in a single burst.

**Limitation:** In certain cases, the increasing frequency of an event could relate to a failure. For instance, instead of an one to one mapping of an event with a failure, increase in the number of occurrences of an event could trigger a failure. In such cases, performing temporal compression would compress the multiple occurrences of an event with the same message code that happened in the same second into a single record. Hence, it would not be possible to capture failures that are raised by increasing frequency occurrence of events. Therefore, it is important to experiment on raw data to capture relations between events and failure based on increasing frequency. To this regard, event frequency statistical modelling (EFSM) based on raw log data is defined in Section 3.2.3.

### 2.3 Message Code Analysis

In this section, various analysis performed on the message codes and their occurrence frequency, distribution over time, etc., are described. All the analysis were performed for 441 days of log data from January 2011 to March 2012. All the figures shown henceforth are constructed after data filtering and preprocessing. The analysis of the raw log data is performed in Section 2.6. The total number of records from the filtered data which is considered for the analysis and experimentation is 14 million, the number of unique message codes, errors, failures, etc. are given in Table A.1 of Appendix A. All the figures shown in this section are constructed by considering the complete information including the failure points which is a subset of errors logged in the machine.

Figure 2.2 shows the frequency of messages logged from a single machine per day across the three message types. The log file data generated from this machine was considered for the analysis through out the thesis. It could be inferred that the number of events and errors logged in a day remained closely related until 04/08/2011 and then the number of errors drastically reduced compared to the events. This could imply a change in behaviour of the system over time. It could also be inferred that the number of warnings logged from the machine on an average stayed low in the order of few hundreds initially and then increased relatively in the later half to few thousands. The machine characteristics as observed from Figure 2.2 changed in the later part in terms of the number of events, warnings or errors that occur each day. But the
occurrence distribution of failures (Figure 2.7) showed that the frequency of failure did not reduce over time which suggests that the reduction in the number of errors occurred did not prevent failure. Although, the failure occurrence is evenly distributed all through the analyzed period, it could be possible that the cause for the failure changed over time which could be confirmed through experiments.

![Figure 2.2: Frequency of message codes per day](image1)

Figure 2.2: Frequency of message codes per day

![Figure 2.3: Stacked column - Frequency of total message codes per day](image2)

Figure 2.3: Stacked column - Frequency of total message codes per day

Figure 2.3 shows the total number of messages logged from the machine per day across three message types in terms of stacked column. The total number of messages recorded per day varies considerably. The change in trend of the machine over time in terms of number of events that are logged
Figure 2.4: Stacked percentage column - Distribution based on message types

is evident from the figure. The warning records logged from the machine are lesser compared to the errors and the events.

Figure 2.4 shows the total number of messages logged from the machine per day across three message types in terms of stacked percentage columns. Around 50 percentage of the records generated were events initially which changed considerably to around 70 percentage over time. Also, the average number of errors logged in the machine was around 40 percentage initially which changed to 20 percentage over time. This could possibly mean a concept drift in the machine behaviour. The average number of warnings logged in the machine was lower initially and changed to around 10 percentage of records over time.

Figure 2.5 shows boxplot of frequency of records grouped based on message types. The corresponding boxplot values are shown in Table 2.3. The distribution of the number of events that occur per day is evenly scattered between the maximum and minimum value. However, there are outliers detected in the number of warnings and errors that occur per day. The maximum value of 10000 warnings on a single day when compared to the median value of less than 2000 explains the outliers. In both the number of warnings and errors that occur in each day, values between 1st and 2nd quartile are densely populated and less scattered which suggests that number of warnings and errors per day revolves consistently on that region. The total number of records that are generated each day is evenly scattered between all quartiles suggesting the varied trend analysed in the previous sections. The outliers in the errors boxplot is due to the higher number of errors logged per day initially. Over time, the number of errors logged per day decreased and hence
the initial values are detected as outliers.

<table>
<thead>
<tr>
<th>Type</th>
<th>Minimum</th>
<th>1st Quartile</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Quartile</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warning</td>
<td>55</td>
<td>858</td>
<td>1714</td>
<td>2503</td>
<td>3732</td>
<td>10230</td>
</tr>
<tr>
<td>Error</td>
<td>112</td>
<td>5557</td>
<td>7903</td>
<td>9979</td>
<td>13030</td>
<td>35340</td>
</tr>
<tr>
<td>Event</td>
<td>123</td>
<td>9048</td>
<td>19100</td>
<td>19280</td>
<td>27420</td>
<td>44510</td>
</tr>
<tr>
<td>Total</td>
<td>290</td>
<td>18180</td>
<td>32490</td>
<td>31760</td>
<td>45360</td>
<td>65240</td>
</tr>
</tbody>
</table>

Table 2.3: Boxplot Statistics - Distribution based on message types

2.4 Failure Points Analysis

Failures are critical errors logged from LO module of the system. There were a total of eight unique critical errors analyzed. These critical errors have the potential to trigger machine failures as explained in Section 1.1. Figure 2.6 shows the total number of occurrences per failure type through out the analyzed log. Failures, LO-8459 and LO-8422, have eccentric characteristics in terms of the number of occurrences with 33 and 692 occurrences respectively. The rest of the failures have occurred on an average over hundred times in the entire duration of analyzed log data.
From the analysed logs, it was observed that the occurrence frequency of some failures is more like a pattern in time in which when the failure occurs, it occurs multiple times in a single burst. The burst like occurrence pattern could be observed from Figure 2.7 where the distribution of all the failures with their respective timestamps are shown. A burst as observed from the figure is followed by non-occurrence of failures for 15 to 20 days. This could suggest the possibility of some scenarios or actions triggering the failure since the failure does not occur for 15 to 20 days followed by burst like occurrence pattern.
2.5 Outliers Analysis

The distribution of the number of occurrences of an event is shown in Figure 2.8. The most frequent 11 events have contributed 87 percentage of record generation. The figure shows the number of times an event occurred along with their percentage of the total number of events. These events are not considered for some of the experimented strategies. This is because of the assumption that an event with million occurrences would not cause the failure which happened on an average per failure type only 182 times. For few strategies, it is best to experiment with the original data set since the increasing occurrence of these events could relate to failures. For these strategies all the events were considered.

Figure 2.8: Analysed Log Data - Distribution of Events

Table 2.4 shows the quartile based summary information on the number of times a message code occurred. The median is at 13 which represents half the population is below 13. In other words, 2103 out of 4206 message codes occurred less than 13 times. Also the first quartile value of 3 implies that 1051 messages have occurred less than 3 times. Further analysis showed that 658 message codes occurred only 1 time. This implies that the log data in terms of the number of occurrences of a message is skewed and experiments should be performed by considering this information. The summary of total occurrences for different types of message code is shown in Table A.3 of Appendix A.
2.6 Raw Log Data Analysis

The raw log data as described earlier could be used to capture relations between events and failures where increasing frequency of events could trigger a failure as opposed to one to one mapping of an event with failure. In order to capture these relations, the EFSM strategy was performed on raw log data. The detailed statistical information of the raw data is given in Table A.2 of Appendix A. The distribution of message codes in the raw data statistic followed the characteristics of the data after temporal compression as discussed in Section 2.3. Therefore, there was proportionate increase in the number of events logged per day over time and proportionate decrease in the number of errors logged per day. This scenario is shown in Figure 2.2.

Table 2.5 shows the quartile based summary information on the number of times a message code occurred. It could be observed that the quartile information for different quartiles closely relates to Table 2.4 which shows the quartile based summary information on the compressed log data. The median is at 15 which represents half the message codes have occurred below 15 number of times. First quartile value of 3 is the same as in Table 2.4. This implies that the events below the first quartile did not change their number of occurrences. Even the third quartile value does not show considerable change in the number of occurrences of respective events below third quartile. This shows that the compression have occurred only in the last quartile.

Further analysis showed that the most occurred 10 message codes have total record count of 14 million which is the entire number of records considered after compression. It could be inferred from this that the compression occurred only in the most occurred few message codes. In other words, only
the most occurred few message codes have occurred in the same second (with a difference only in microseconds) multiple number of times. It is also interesting to note that the maximum number of times, single message code has occurred is 4947000. In the compressed log data, this code has occurred 1792740. This shows roughly 3 million records of the same message code was compressed. The summary of total occurrences for different types of message code for raw data is shown in Table A.4 of Appendix A.

2.7 Conclusion

In this chapter, log data exploration was performed. It was understood that the machine characteristics changed over time in terms of the number of events and errors logged by the machine. It was also observed that the decreasing number of errors over time did not prevent the failures from happening. It has to be verified through experiments and strategies that if the root causes for a failure of particular type changed over time. Analysis on the number of times a message code of particular type occurred after compression showed that the least occurred 1000 message codes occurred less than 3 times and 2103 out of 4206 message codes occurred less than 13 times. The raw data analysis showed that the compression of data affected only the most occurred message codes in the first quartile. The comparison of raw log data and the compressed log data showed a difference of 11 million additional records in raw log data. The most occurred 10 message codes in the raw data accounted for 14 million records where as the total number of records in the compressed log data is 14 million. This shows that the compression has affected only the most occurred few message codes.
Chapter 3

Preliminaries and Related Works

This chapter introduces preliminary concepts and employed strategies to realize the thesis objective. The concepts discussed in this Chapter are related to part of the tasks performed in the model development phase as shown in Figure 3.1. The training and evaluation of model performances in the model development phase as shown in the figure are performed in successive chapters. Section 3.1 starts with a brief discussion on fault prediction systems. Section 3.2 introduces employed strategies including an example instance of how these strategies could be applied real time. Section 3.3 builds on Section 3.2 and discusses elaborately the existing literature behind the employable strategies and the recent works in fault prediction systems in general. It also introduces various works performed in ensemble systems.

3.1 Fault Prediction Systems

Fault prediction systems is the one of the principal areas in Reliability Engineering and System Safety. Many variants of fault prediction systems involving static, dynamic and hybrid fault tolerance techniques have been proposed. Examples for hybrid techniques which involve a combination of many fault prediction techniques include meta-learning predictive systems where multiple techniques are incorporated in the system to boost prediction accuracy. Examples include meta-learning failure predictors proposed in [1][4]. This project aims in developing a decision support system for proactive fault handling. The motivations for this has been discussed in Section 1.2.
Broadly, fault prediction can be approached from two different perspectives: model-based or data-driven[1]. A model-based approach derives a probabilistic model of the system [7]. A warning is triggered when a deviation from the model is detected. Examples include systems proposed in [2][3][4]. The statistical based method is effective in leveraging the temporal correlation among fatal events. Data-driven approaches, focus on learning and classifying occurring faults from historical data without assuming a priori model ahead of time[1]. The recent development in the field of data mining opened new research orientation to fault prediction. This include a possibility of creating ensemble systems with multiple models and techniques. Examples of such systems are proposed in [6] where clustering is used in combination with semi-markov models for proactive fault handling, [1] incorporates both statistical modelling and machine learning techniques to develop a meta-learning system. The existing strategies and their proximity towards applying it to this project has been elaborated further in Section 3.3.

All the strategies that are defined in this chapter involve with analyzing
Preliminaries and Related Works

Machine characteristics over a specific period of time before the failure occurrence. For instance, Figure 3.2 shows a scenario in which a machine is observed over a specific time window defined before each failure occurrence separated by a distance represented by lead time from the failure. The value of lead time represents how early failure prediction could be made. In operational settings, the observed behaviour which is common in most of the past failure occurrences are used to trigger failure alarms. The pattern mining strategies defined in Section 3.2.1 are based on this method where rule mining algorithms proposed in [8] are modified to find event failure relationships in an event stream using time window and lead time as shown in Figure 3.2.

Figure 3.2: Failure Prediction - Example Instance 1

Figure 3.3 shows another example in which failure prediction could be applied in an event stream. The example is concerned with arbitrary time windows of variable lengths defined between two successive failures unlike Figure 3.2. Therefore, in this case, machine characteristics are observed over a long period of time between two successive failures. The statistical strategies which maps increasing frequency of respective message codes (events henceforth) with failures use this methodology. A change in machine deviation before failure occurrence in each time window is investigated. If a change in machine characteristic was observed in multiple failures in the past, then it is used in operational settings to trigger failure alarms.

In certain cases in which the failure is caused by multiple distinct reasons, there would not be specific set of events which trigger the failure with high certainty. In such cases, the expected machine behaviour is analyzed and a failure alarm is triggered whenever the machine deviates from its expected behaviour. This scenario is shown in Figure 3.4 where a time window of arbitrary length is set before a specific duration from failure occurrence in which the normal behaviour of the machine is expected. The logs generated during the period in which machine failures are triggered are usually eliminated to
Preliminaries and Related Works

Figure 3.3: Failure Prediction - Example Instance 2

capture the normal behaviour of the machine. To apply this method, specific information regarding when the machine failures are triggered should be known prior. In the case of TWINSCAN machines, there are several failures each with distinct characteristic and time period during which machine deviation occurs before the failure is variable. For this reason, this method was not examined. The following section describes the employed strategies which are built on the above mentioned methods of failure predictions on event streams.

Figure 3.4: Failure Prediction - Example Instance 3

3.2 Employed Strategies

This section discusses the employed strategies inclined towards achieving the project objectives. The implementation model of the strategies with parameter settings and experimentation is discussed elaborately in Chapter 5. In Section 3.3, existing works based on these strategies is discussed. The strategies discussed in this section have the objective to find strong correlations between the occurrence of a failure and several factors, including the time stamp of other events or failures, and the occurrence of non-fatal events. Multiple prediction schemes have been proposed in this section to find these correlations. Their effectiveness have been demonstrated through analysis.
and results presented in Chapter 5. In the described strategies henceforth, an event to failure relationship is synonymous with a message code to failure relationship. A message code could be either of an event or warning or error. For instance, an increasing frequency of an event to failure relationship represents increasing frequency of a specific message code to failure relationship.

3.2.1 Pattern Mining

Pattern Mining is one of the principal areas which is used to establish patterns in data. Well known algorithms proposed in [8][9] deals with establishing patterns through association rules in transactional data. These rules capture relationships between individual items or item-sets. The rules could be established as an item-set as proposed from [8] or an ordered sequence of items as proposed from [9]. Existing fault prediction systems that incorporate these algorithms are discussed in Section 3.3. Although primary algorithms proposed in these papers involves transactional data sets, it could be modified to find relations in continuous data streams as is the case in this project.

Pattern mining is chosen since it efficiently identifies relationships between events and failures. It is also efficient to establish relationships among items that have one to one occurrence characteristics rather than multiple occurrences or increasing frequency distribution of events triggering a failure. Machine characteristics are captured in patterns and a failure alarm is triggered in operational settings whenever the pattern is encountered. In pattern mining, rule mining algorithm effectively captures pattern based relationships between events and failures. Sequence mining as proposed in [9] and partial order sequencing are deemed unfit to be used in this case since the machine characteristics are varied and an ordered sequence of events are not expected to occur with very high confidence. However, the possibility of sequence of events which could trigger a failure is investigated by alleviating the strictness of sequence mining algorithm in [9] by rules state space strategy proposed in this section. In order to investigate the possible sequencing of events with . The subsequent sections discuss the extension of basic rule mining algorithm proposed in [8] a novel batch rule mining strategy proposed as per the events and failure occurrence characteristics.
Rule Mining

Rule mining (RM) aims in capturing machine characteristics in terms of events to failure rules. The system could be trained to trigger failure alarms whenever the rules are encountered.

The algorithm takes the following input parameters.

1. Lead time, $\triangle t_l$.
2. Time window, $\triangle t_w$.
3. Confidence, $c$.
4. Support, $s$.

The lead time defines the prediction time before failure occurrence by observing machine characteristics for the length defined in time window, $\triangle t_w$. The $\triangle t_w$ is characterized by a start time, $t_{ws}$ and an end time, $t_{we}$ as shown in Figure 3.5.

Objective is to create a system which captures frequently occurring patterns in $\triangle t_w$ which is defined with a gap of $\triangle t_l$ before each failure. The scenario is shown in Figure 3.5 where an alarm that goes off at $t(12:00)$ inside $\triangle t_w(9$ hours) gives a guaranteed time of at least $\triangle t_l(7$ hours) before failure is predicted. The exact time the failure would occur will be $(t_{we} - t) + \triangle t_l$. This exact time the failure will happen is not obtained since the order sequence of item in the rules inside $\triangle t_w$ is not maintained. However descriptive statistics could be captured for each rule that contains information such as the mean time to failure (MTTF). The advantages of capturing such information and its usefulness in operational settings is further elaborated in Section 5.1.1.

![Figure 3.5: Rule Mining](image)

The confidence of a rule is obtained by counting the number of times an
item has occurred inside $\triangle tw$ which is defined $\triangle tl$ before each failure divided by the total number of times the considered item has occurred.

The Confidence, $c$ of the rule is given as

$$
Confidence, c = \frac{\sum_{i \in tw} \text{Number of occurrences of item in } tw_i}{\text{Total item occurrences}} \quad (3.1)
$$

The RM algorithm as described in [8], calculates the confidence of the rule of the form $x \rightarrow y$ by the ratio of the number of transactions that contain both $x$ and $y$ to the number of transactions that contain $x$.

The support is used to define the strength of the rule. The support level defines the minimum number of failures that should be mapped to the rule. Support is important since it acts as a filtering mechanism of rules which are mapped to only one or few failure instances yet have hundred percent confidence.

The Support, $s$ of the rule is given as

$$
Support, s = \frac{\sum_{i \in tw} \text{Number of occurrences of item in } tw_i}{\text{Total failure occurrences}} \quad (3.2)
$$

The algorithm is as follows.

- **Step 1** Extract *message codes* (inclusive of failures) and *time stamp* information from log data.
- **Step 2** Define input parameters.
  - Lead Time $\triangle tl$.
  - Time Window, $\triangle tw$.
  - Confidence $c$.
  - Support $s$.
- **Step 3**
  - For each failure type, apply association rule algorithm as discussed in [8] to build rule models. The confidence, $c$ and support, $s$ of the rule are obtained as in equation 3.1 and equation 3.2 respectively.
  - For each failure type, screen rules that do not satisfy $c$ and $s$. 

---

*Data Driven Decision Support for Failure Prediction* 33
• **Step 4** Trigger failure alarm for an error if an association rule is observed at time \( t \). The failure is estimated to occur at \( \Delta tw+\Delta tl \) in the best case and \( \Delta tl \) in the worst case and between the start of \( \Delta tw \) and \( \Delta tw+\Delta tl \).

The events to failure relations are captured only by the single item rules of the form \( Event \rightarrow Failure \). The motivation of perceiving only the single item rules is that the single item to failure correlation has more implementation value in real time operational settings since the alarm should be triggered upon the occurrence of a single item in the item-set. If the time window, \( tw \) value was set to observe machine characteristics for a longer duration, an item-set formed by considering the events that were logged at the beginning and the end of the time window, if used to predict failures, observing for all the events in the item-set to occur could prove fatal since alarm goes off lately at the end of the time window. In addition, chances of item-sets composed of certain events that have very high occurrence (Section 2.5), occurring in multiple time windows satisfying the support threshold value are high. This could produce weak relations. Also, if the length of the time window is large, since the sequencing order of the items in the rule is not maintained, chances of producing weak relations are much higher. An event which happened at the beginning of the time window of length 24 hours, for instance, would be considered in the same item-set, with an event which occurred at the end of the time window which does not reflect the real time operations of the machine.

![Figure 3.6: Example Instance - Rule Mining](image)

For instance, (Figure 3.6) shows the example instance of the log data stream. Consider the following input parameters.

1. **Lead Time**, \( \Delta tl = 2 \) hours
2. **Time Window**, \( \Delta tw = 6 \) hours
3. **Confidence**, \( c = 1.0 \)
4. **Support**, \( s = 0.75 \)
D → Failure is the only rule that satisfies the input parameter as per the RM algorithm. The other rules, A → Failure, for instance does not satisfy input parameters, because A has occurred 3 times inside \( \Delta tw \) and 13 times overall which gives a confidence of 0.23 which does not satisfy input \( c \). The rule E → Failure although has a confidence, \( c \) of 1.0, it does not satisfy the support, \( s \) which is 0.33. As described earlier, this example instance reiterates that if the support value is not defined, the rules which have a confidence value of 1.0 yet having only very few failure mappings would not be screened which decreases the strength of the rule.

**Batch Rule Mining**

Batch rule mining (BRM) aims in capturing machine characteristics in terms of event-failure batches. The motivation for the BRM strategy is the burst like occurrence patterns of the event. The rules are processed in batches to ease the strictness of basic rule mining algorithm defined in the previous section. The following section discusses the algorithm and the next section discusses the motivation behind choosing this strategy.

The algorithm takes the following input parameters.

1. Lead time, \( \Delta tl \).
2. Time window, \( \Delta tw \).
3. Confidence threshold, \( c_b \) for batches.
4. Confidence threshold, \( c_r \) for rules mined in each batch.
5. Support, \( s \).
6. Batch distance interval, \( \Delta d \) (the minimum distance that separates two batches).

![Figure 3.7: Example Instance - Single Event(E) Stream](image)

The batch distance interval \( \Delta d \) parameter is used to create non-overlapping batches of events. The continuous stream of a single type of event is divided into multiple non-overlapping batches based on \( \Delta d \) parameter. For instance, \( \Delta d \) parameter is used to convert a single event stream as shown in Figure 3.7 to Figure 3.8 by considering only the timestamp information of individual
occurrences of that particular event. This is performed by creating a new batch whenever two successive occurrences of respective event is greater than $\Delta d$.

The Confidence, $c_r$ of the rule is the confidence, $c$ as explained in equation 3.1 which is applied locally to each batch.

The Confidence, $c_b$ is given as

$$Confidence, c_b = \frac{\text{Number of batches that satisfy } c_r \text{ for rules}}{\text{Total number of batches}}$$

Support, $s$ of rule is calculated in the same manner as in equation 3.2.

![Example Instance - Batched Single Event(E) Stream](image)

The algorithm is as follows.

- **Step 1** Extract Event, Failure and Time stamp Information from Log Data.
- **Step 2** Define input parameters.
- **Step 3** Divide data stream into multiple non overlapping batches as per $\Delta d$.
- **Step 4** Apply RM algorithm discussed in Section 3.2.1 for each batch.
- **Step 5** Find the confidence, $c_r$ for each rule in each batch. Filter the rules that do not satisfy confidence, $c_r$ and support, $s$.
- **Step 6** The rules that satisfy the batch confidence, $c_b$ are considered as strong rules.
- **Step 7** Trigger failure alarm for an error if a strong association rule is encountered.

The event to failure relations are captured only by the single item rules of the form $Event \rightarrow Failure$. The motivation for this is explained in RM strategy.
Motivation - BRM: It has been observed from the log that certain failure-event types occur in bursts. The basic rule mining algorithm does not capture the burst characteristics since it establishes rules based on a global level considering the overall event and failure occurrence. High number of failure occurrence in a single burst makes the rule mining algorithm inefficient to capture events that have correlations with failures in a subset of bursts.

To explain this scenario, consider an event stream as shown in Figure 3.9. It shows both failure and events that occur in bursts with variable bursts length. It is evident from the figure that the event, E correlates with failure, F in three of four bursts. But as per the normal rule mining algorithm, confidence of the rule $E \rightarrow F$ is only 0.6 which makes the rule weak. This scenario is successfully captured in BRM.

![Figure 3.9: Example Instance - Event Stream](image1)

Figure 3.10 shows an example instance of BRM algorithm. Consider the following input parameters. For simplicity, lead time, $\Delta t_l$ and time window, $\Delta tw$ have been abstracted and could be considered arbitrary.

1. Confidence Threshold, $c_r = 1.0$
2. Confidence Threshold, $c_b = 0.75$
3. Batch distance interval, $\Delta d > 5$ hours

The performance of batch rule mining is as follows.

1. Four Batches ($\Delta d > 5$ hours)
2. Rules obtained in each batch (that satisfy $c_r = 1.0$)
   - Batch 1. $E \rightarrow F$ ($c_r : 1.0$)
   - Batch 2. $E \rightarrow F$ ($c_r : 1.0$)
Batch 3. $E \rightarrow F (c_r : 1.0)$  
Batch 4. $E \rightarrow F (c_r : 0)$  
3. The rule $E \rightarrow F$ satisfy $c_b = 0.75$.

**Limitation:** It is important to also note that BRM is efficient only to capture event-failure relationships that have burst type occurrence characteristics. This is because, the absence of burst will drastically limit the size of each batch. If the size of the batch is negligible even strong rules which would otherwise be mined in RM algorithm will have lower confidence overall. To explain this phenomenon, consider an event stream as shown in Figure 3.12. If BRM algorithm is applied with a $c_b$ of 0.70 and $c_r$ of 1.0, a rule $E \rightarrow F$ that satisfies the input parameter is obtained. But this necessarily, does not mean there is a relation since in only three out of 13 occurrences of the event, the failure is observed.

One possibility to eliminate the above scenario would be to consider an additional support parameter that filters rules based on the number of events in each batch so that batches that contain only one or two events would not be considered for observation. But, if this was applied on an event stream with multiple batches that contain only one or two events yet maps with failure in many batches, the rules, although having predictive power with high confidence, would not be captured. Therefore, it is important that manual analysis of BRM rules are performed under operational settings if incorporated in a DSS.

**Rules State Space**

Rules state space (RSS) aims in capturing machine characteristics in terms of state space composed of single event rules as shown in Figure 3.12. At the root of the state space is an individual failure. Upon applying RM algorithm as explained in Section 3.2.1 to a failure, all the events that are related to the failure in terms of event failure rules are obtained. Each of these events
are added as a separate state and a directed edge is added from each new state to the root, that is the failure.

In the next iteration, all the events represented by states at level one from the root are fed to the rule mining algorithm. The rules obtained for each of those events are further added as new states and directed edge is added from the new states to the corresponding event state at level 1. This process is repeated for each state recursively until either of the following conditions are not met.

- Failure Probability (FP) threshold is not satisfied. The FP represents the probability that the failure will occur at any given state. If this threshold is not met, the process is stopped. In Figure 3.12, for instance, this is represented by the attribute Failure Probability in each state.

- When no further rules that satisfy the rule mining algorithm are obtained for the state.

![Figure 3.12: Example Instance - Rules State Space](attachment:image)

Each state is tagged with the following attributes

- **Next State Probability** (NSP): Represents the transition probability to next state. In this case, the probability that the state pointed by the directed edge (parent state) will occur. For instance, in Figure 3.12, for the state E5, it is the probability that E1 will occur (0.80).
Preliminaries and Related Works

- **Failure Probability (FP)**: Represents the probability that a failure will happen from the target state. This is obtained by multiplying the NSP attribute value of all the states in the path from the target state to the failure.

- **MTNS**: Represents the Mean Time to Next State. This is obtained by from the descriptive statistics of the rule which is represented by the state.

- **MTTF**: Represents the Mean Time to Failure. This is obtained by the summation of MTNS attribute value of all states from the target state to the failure.

For the states at level one, occurrence probability of next state and occurrence probability of failure is the same as is the mean time to next state and mean time to failure.

The advantage of this algorithm is that by recursively applying RM algorithm that increases the number of levels by and distance between any state and the failure, it is possible to predict failure with a least probability of what is defined in the failure probability threshold. Also, when the algorithm is recursively applied the MTTF increases with every level that is added which ultimately increases the time before a failure is predicted.

### 3.2.2 Classification

**Failure**: A failure refers to the presence of a specific error of type LO for which the classifier is being built. For the prediction by classification of individual failure points of the system, the following steps were employed.

- **Step 1** Define time length.
- **Step 2** Divide the entire duration of the log file data into multiple non-overlapping time windows \( tw \). A single time window, \( tw_i \) is characterised by a start time \( ts_i \) and end time \( te_i \).

\[
tw_i : [ts_i, te_i]
\]

where, \( i \in \{ \text{log time} \} \)

\[
 ts_i = te_i - l
\]

- **Step 3** Define a list of features, \( f \) and the values that they take which could capture the behaviour of the machine within a specific time period.
• **Step 4** Then for each time window $tw_i$,
  
  - For each $e$ in $f$, define a value that $e$ takes on $tw_i$. This value could be anything that represents the behaviour of $e$ on $i$. For Example, for the experiments performed in Section 5.2.1, this value was the number of times $e$ has occurred during $i$.
  
  - Define a class label, $cl$. The class label takes either one of the two values: Positive or Negative which represents the presence or absence of failure respectively (occurrence of a critical error) in a future time window depending on how early the prediction is made.

  *A classifier could be built for each failure separately and each model in this case is unique that it is specific to a particular failure and predicts whether or not the failure occurs at a fixed time window in the future. Hence, the model does not capture the overall collective behaviour of the system. This aids in the analysis of the individual failures rather than the complete system in which case, it would not be possible to analyze failure prediction based on individual failures types listed in Section 2.4.*

Two different ways of feature vector construction considered are as follows.

For, $\text{class} \in \{p,n\}$,

$p$, positive class, failure occurs at $tw_{i+c}$, where $c$ is user predefined lead time,

$n$, negative class, no failure occurs at $tw_{i+c}$, where $c$ is user predefined lead time.

**Frequency Based Classification (FBC)** In this method, each message code is considered as a feature. Each feature instance would contain the number of occurrences of a particular feature (i.e) the corresponding event during $tw$ and a class label, $cl$.

Each feature vector is defined as

$$f = (f_1, ..., f_n) \rightarrow \text{class} \quad (3.5)$$

where,

$f_e$, number of occurrences of event $e$ at $tw_i$

A failure in this case represents that the considered failure has occurred at time window $tw_{i+c}$. It could be inferred that the classifier for this problem
would be binary classifier. The detailed methodologies, experimentation, results and analysis of this strategy are further elaborated in Section 5.2.1.

**Rules Based Classification (RBC)** In this method, each rule discovered for a specific parameter setting from RM strategy explained in Section 3.2.1 is considered as a feature. Each feature instance would contain a binary value that represents the presence and absence of a particular feature (i.e) the corresponding rule during $tw_i$ and a class label, $cl$. Each feature vector is defined as

$$f = (f_1, ..., f_n) \rightarrow \text{class}$$

(3.6)

where,

$f_r$, presence or absence of rule $r$ at $tw_i$

Feature vector construction based on the frequency of an event in a time window is preferred over a binary feature construction method based on the presence or absence of event in a specific time window. This is because the number of times an event occurred in a time window is more specific and there is a difference between whether an event has occurred only once or multiple times in a time window which could boost the predictive ability of the classifier. The RBC feature vector construction aids in analyzing the predictive ability of the rules established in pattern mining strategies.

**Class Label Construction** The class labels are constructed based on whether the failure occurs in the next window as shown in Figure 3.13. In this case, if the failure occurs in $tw_{i+1}$, then the class label is marked *positive*, otherwise *negative*. In the second case, if the failure occurs in $tw_{i+2}$, then the class label is marked *positive*, otherwise *negative* as shown in Figure 3.14. This gives a guaranteed buffer period which is equal to length of the time window, $l$ before which a failure is predicted.

By choosing two separate construction mechanisms for class labels, it is possible to compare how far the prediction accuracy changes between the two. The trade-off between how the predictive accuracy of the classifier changes for increasing lead time values could be analyzed based on the results for $tw_{i+1}$ and $tw_{i+2}$. This relates to the research question of studying the predictive characteristics of the machine over a period of time and the effectiveness of failure prediction for increasing lead time values.
any failure, irrespective of the length, $l$ of time window would not be greater than 182 on an average. For decreasing time granularities, the number of instances increases rapidly. For instance, for $l$ value of 1 hour, more than 10,000 instances are created. Therefore, the negative class instances gets heavily skewed which suppresses the performance of classifier. To eliminate this possibility, feature instances compression was performed.

A two step process as shown in Figure 3.15 was performed. The steps involved are as follows.

- **Step 1** For each positive class instances, $n$ number of feature instances immediately preceding the chosen positive class instances. In Figure 3.15, $n$ is set to 2. Formally,

$$P = (p_1, p_2..., p_n), \forall i \in P (c < (i \to 1 - n))$$  \hspace{1cm} (3.7)$$

where,
c, compressed ordered list of instances
P, ordered list of positive class instances
<, addition

- **Step 2** Eliminate duplicate instances from c.

The compression of feature instances, reflects a change from operational settings in terms of the sequencing order of the instances as part of the negative instances between two successive positive class instances are eliminated. However, this does not affect the performance of the classifiers since the classifiers build the model by considering individual information of each instance and does not consider the ordering sequence of the instances.

### 3.2.3 Event Frequency Statistical Modelling

The pattern mining and classification strategies discussed earlier involve the use of time window to capture the machine characteristics for a specific period of time. In order to capture machine characteristics between successive failures irrespective of the size of the time window, the Event Frequency Statistical Modelling (EFSM) is used. This involves the use of capturing the change in occurrence frequency of individual message codes between two
successive failures. It is based on the principle that a message code which increases its occurrence frequency before a failure every time a failure occurs could be related to the failure.

Figure 3.16 shows an example instance of applying EFSM. This involves grouping all the message codes between two successive failures. Each message codes are then individually analyzed for their frequency distribution between two successive failures. In the figure, event A increases its frequency before the occurrence of every failure which could mean that there is a correlation between event A and Failure F. The machine characteristics are analyzed from the point of the previous occurrence of failure up until the start of the lead time, $\Delta t$ window. Therefore by capturing machine characteristics before $\Delta t$ before a failure occurs, it is possible to predict failures before $\Delta t$ units.

![Figure 3.16: Example Instance - Event Frequency Statistical Modelling](image)

For the calculation of increasing frequency of an event, the Inter Arrival Time (IAT) between two successive occurrences of the corresponding event is calculated. This is performed by calculating the difference between two timestamps associated with two successive occurrences of an event. The example instance of this calculation is shown in Figure 3.17. The decreasing trend of the IAT values shows an increase in the frequency of corresponding event.

![Figure 3.17: Calculation of IAT](image)
The calculated IAT values are then grouped per failure occurrence. Associated with each failure occurrence is a list of IAT times for each event. If an event has a decreasing IAT list for multiple failure occurrences, then the event is considered to be correlated with failure. In order to find strong correlations, a threshold value for the number of lists that should satisfy the increasing frequency trend belonging to a failure is defined. An example instance of an event with decreasing IAT and increasing frequency is shown in Figure 3.18. The distribution functions are used to capture the decreasing values of IAT for each list belonging to a failure. The values of the list are fit to a decreasing distribution such as Weibull[21] with corresponding shape parameters to capture the decreasing trend of IAT. If the number of lists that satisfy the goodness of fit tests for the distribution function is greater than the defined threshold level, the event is considered to have a correlation with corresponding failure.

Figure 3.19 shows the holistic picture of the employed strategies. It shows a relation between the employed strategies and their respective domain reasons for employing them and the set of features used for the construction and experimentation of these strategies.
3.3 Related Existing Works

This section describes all the existing related works to fault prediction and the employed strategies in particular.

[22] describes previous work conducted on fault prediction of ASML TWINSCAN machines. The work proposes a condition based predictive maintenance which predicts expected remaining life time of a component in days. To this regard, machine learning classification algorithms were used to represent the remaining life time of a component as an interval. The proposed work is for a condition based predictive maintenance of a single module. The previous failure cases in separate machines occurred in this module along with the conditional parameters specific to the module was used as an input domain. Some of the differences between [22] and this thesis are the following:

- This thesis uses the log file data generated from the TWINSCAN machines as an input domain where as the work proposed in [22] uses...
the conditional parameters of the respective module from different machines.

- The prediction is performed for the estimation of useful remaining life time of a single module, \( X \). To this regard, the past failure cases in \( X \) were analyzed. In this thesis, 8 different failure types of module \( LO \), each occurring on an average of 182 times in the considered log duration of 440 days were analyzed and failure prediction for future occurrences of these failures are experimented separately for each failure type.

- This thesis involves establishing events to failure correlation in an event stream considering the time stamp characteristics of the events and failures. The work described in [22] uses conditional parameters and build model based on the acceptable threshold level of these parameters for machine functioning.

**Pattern Mining**

The rule mining algorithm as proposed in [8] and the sequence mining algorithm as proposed in [9] have been experimented in various works to extend them to an event stream. Apriori algorithm proposed in [8] was originally proposed only when the list of events occurs in a transactional form and hence in this case, it is independent log file events, it is necessary to create transactions of events manually. This could be created by considering the list of events that occurred before the occurrence of every failure within a specified time window as described in Section 3.2.1.

Mining sequential patterns [9] is an extension of the work performed by Agarwal in the base paper of mining association rules between sets of items in large databases to preserve the order of the events in the rules. Therefore each rule corresponds to a sequence of events. The paper deals with the transactional database which implies that the sequences of rules could be directly created from the transactions. The sequence mining algorithm proposed in this paper could be extended to event streams in the same way as RM (Section 3.2.1) by preserving event ordering inside time windows.

The rule mining algorithm for event stream data proposed in [1] is an extension of the algorithm proposed in [8] for transactional data. The extension is applied to an event stream similar to the log data of TWINSCAN machines. The RM algorithm was applied in the thesis the way as proposed in [1].
The work proposed in [11] uses event stream of data and creates disjoint sets of related events. It is basically a clustering algorithm which clusters events which occur together in a data stream which could be modified to find clusters related to failures.

[20] proposes an extension to the previous two rule mining algorithms that dealt with transactional form of data. The proposed algorithm in [20] decomposes stream of events with associated timestamps to frequent sequences of events that occur together. The paper did not follow strict ordering sequence of events but followed a partial ordering sequence. This is because in certain real time machine operations have varied characteristics and chances of discovering specific sequence of events with high confidence is limited. To alleviate this scenario, the paper proposed methods to find partial ordering sequence of events. An example of partial ordering sequence rule is, whenever A or B occurs in either order, it is followed by C. The algorithm proposed in this paper detects all the sequences of events that relate to each other in a continuous stream of data.

[6] proposes a combination of clustering algorithms and Hidden Markov Chains to define probabilities that the failure will happen. The clustering algorithms are used to obtain the relations between individual events and to group a list of events which could be related. All patterns of errors that lead to a failure was identified. This pattern was then used in the state space of the Markov Models. The paper uses a combination of clustering algorithms and Hidden Markov Chains to define probabilities that the failure will happen. The clustering algorithms are used to obtain the relations between individual events and to group a list of events which could be related. All patterns of errors that lead to a failure was identified. This pattern was then used in the state space of the Markov Models.

[12] proposes a learning algorithm for the construction of hidden markov models based on establishing frequent episodes of event sequence. An episode is considered as an ordered list of events occurring at respective time points. A special class of hidden markov models called episode generating HMM are introduced. Each frequent episode whose frequency exceeds a user defined threshold are then associated with a unique episode generating HMM.

The method as proposed in [6] is closely related to the RSS strategy proposed in Section 3.2.1. Difference between [6] and the RSS algorithm proposed in this Chapter is that, in RSS, the list of events which are related are not clustered using clustering algorithms proposed in [6]. The list of events that are related to the failure are grouped at each level by applying the rule mining algorithm proposed in Section 3.2.1. An advantage of using the RM
algorithm than the clustering algorithm is that, the list of events that relate to a failure could be experimented through various lead times and each state could be made to include descriptive statistics which are otherwise not present in [6]. This under operational settings could prove to be effective.

**Classification**

[10] proposes a failure prediction and control system for large clusters and event logs by existing time series algorithms, Rule Based Algorithms and Bayesian Network Models to predict the events. The rule based model was based on the frequently occurring events that precede a target within a fixed time window. The application of this strategy is explained in Section 3.2.2 where rules based frequency vector construction based on the rules obtained from the RM algorithm is proposed. The feature vector is constructed for the RBC in the same way as explained in [10]. However, arbitrary time window to understand how the predictive ability of the models change over different time window lengths would be experimented in RBC proposed in Section 3.2.2. In addition, the class labels are constructed in a different way as explained in Section 3.2.2 for failure prediction of $tw_{i+1}$ and $tw_{i+2}$. Also, the instance compression is an additional parameter heuristic experimented in RBC defined in Section 3.2.2.

An ensemble system is proposed in many works in which multiple strategies that work the best with domain data are developed and made to work for prediction or classification. Examples include [1][14]. [15] discusses assorted number of reasons and scenarios which mandates the use of ensemble classifiers and the history of ensemble classifiers in general and various works and progress made in the area. The closest work related to ensemble model and failure prediction to this project is proposed in [1] where multiple strategies both statistical and data driven based are used to predict failures in the system. The limitations of using only one strategy as opposed to multiple strategies are also discussed.

[17] proposes methodologies to predict the failure with a certain probability in a continuous stream of data when the entire information is not available. It suggests a decision tree with nodes leading to a failure. When the combined probabilities of the nodes leading to a failure is above a threshold, the model triggers a warning alarm.

The work described in [23] concerns time series Heart Failure Hospitalization (HFH) prediction. Model is constructed based on predictive features of
patient data relating to heart failure with respect to a specific time window. Instances are constructed based on a time window. The positive training instances are constructed by considering the measurement features (medical history of patient, for instance) of patient from the day $t_{h-14}$ to $t_{h-1}$ where $t_h$ represents the occurrence of HFH. The prediction for HF hospitalization is performed for a period of from day $t_i+1$ to $t_i+14$ where $t_i$ being the present day. The prediction is performed each day and predictive features of a patient are updated as in when new measurements become available. This work of time series heart failure prediction is closely related to this thesis. The construction of the feature instances in this thesis resembles [23]. However, the instances in this thesis are constructed in non-overlapping time windows. The positive class instances are formed based on whether or not the failure occurs in a time window $tw_{i+\text{leadtime}}$ where $t_i$ being the current window.

**Statistical Modelling**

[13] describes a statistical model based failure prediction of a machine whose failure is not influenced by external factors and variables and is purely based on distribution history of its own failures. The probability of a failure is computed as:

\[ P(T \leq t_0) = \int_0^{t_0} f_T(t) \, dx \]  

(3.8)

Where $f_T(t)$ is the density function which describes distribution of failures in time.

To do these calculations, it is mandatory to have density function which describes distribution of failures in time. Without this density function identified it is not possible to do any predictions of failures. Empirical distribution functions based on history of previous failures could be used for this purpose. Similar approach by considering the Mean Time Between Failure (MTBF) is proposed in [4]. By using empirical distribution functions, although it is possible to suggest future occurrences, it is a naive approach which is impractical to be applied in this case since a failure does not only depend on the previous distributions but also is caused by multitude of external factors and variables in real time.

Regression analysis is the other established method which is used in time series forecasting. The basic form of regression analysis is the bivariate linear
regression which takes a single dependent variable \( Y \) which is dependent on the independent variable(s) \( X \). E.g., house rates could be dependent on the house size. Given a history of the past \((x, y)\) values, the idea is to develop a linear equation which when given future \( x \) values, tries to predict the \( y \) value. In this case, given the house size, tries to predict the house rate. The linear equation would be of the form:

\[
y = \beta_1 + \beta_2 x
\] (3.9)

Where \( \beta_1 \) is a constant and the coefficient \( \beta_2 \) is the slope of the line. This line describes how the mean response of \( y \) variable changes with the change in explanatory variable \( x \). The bivariate linear regression analysis is further extended as multivariate linear regression analysis which attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable \( x \) is associated with a value of the dependent variable \( y \). The equation is as follows:

\[
y = \beta_1 + \beta_2 x_1 + \beta_3 x_2 + ... + \beta_n x_n
\] (3.10)

The linear regression analysis is further extended to accommodate the non linear properties of the data. There are non linear regression analysis methods to model the non linearity in the data. The inherent model of the regression analysis supports the direct dependencies in the data which contains fixed number of independent variables and a single dependent variable. This could be extended to be applied in this thesis by modelling the relation between the occurrence of events with respect to failures. This is performed in the same way as suggested in EFSM by calculating the IAT values of individual events between two consecutive failure occurrences. A regression fit characterizing decreasing IAT values in multiple failure occurrences could mean there is a correlation between the event and the considered failure.

One of the earliest works in Error Log Analysis and Failure Prediction is proposed in [3]. A model based system is proposed which considered that the system is composed of multiple components. These components are called Field Replaceable Units (FRU). Each of these units is vulnerable to failures. A failure prediction heuristic was developed to observe each FRU and fire an alarm upon a prediction of failure for the corresponding FRU. The strategy rides on a very weak assumption that the list of events that causes the failure in each sub component is already known. These events are monitored and the proposed algorithm is primarily based on the fact that these events increase
their occurrence frequency above a certain threshold before failure happens. But this project deals with failure-event relation that is not predefined. This makes the model proposed in this paper to be less effective to be applied to TWINSCAN machine failure prediction.

[10] proposes linear time series statistical models were used in one of the methods for event prediction. The statistical model used in this paper is inefficient to be used in the project since it does not consider multiple events which could be related to the failure.
Chapter 4

Evaluation of Techniques

In this chapter, the evaluation criteria and techniques defined per strategy are presented and elaborately discussed in the following sections.

4.1 Pattern Mining Evaluation

In pattern mining, the strength of the rule and the usability of the rules at operational settings are evaluated with the aid of confidence and support of each rule. The evaluation metrics for the individual strategies are as follows.

In RM, the strength of each rule is evaluated by Confidence, $c$ and Support, $s$ as described in equation 3.1 and 3.2.

The importance of support, $s$ and confidence, $c$ are elaborated in example instance of Section 3.2.1. In addition to confidence and support, descriptive statistics about each rule such as the MTTF when the rule is encountered, the standard deviation, Skewness of MTTF, etc. are generated. The descriptive statistics gives an additional information of when the failure is likely to happen in addition to the lead time, $\Delta t_l$. The importance of these descriptive statistics of the rules are further explained in Section 5.1.1.

In BRM, the strength of each rule is evaluated by Confidence $c_r$, Confidence $c_b$ and Support $s$.

In RSS (Section 3.2.1), evaluation could be performed globally at state space level and locally for each state in the state space. Globally, information such as the number of states, number of levels and the longest MTTF value of the state space are gathered for evaluation purpose.
In addition, the list of attributes that are associated with each state in RSS as described in Section 3.2.1 could be analyzed along with the parameters for the RM algorithm could be manipulated at the local level for each state. Hence, all the evaluation metrics as described in Section 4.4.1 could be used to evaluate the usability of the model.

The confidence and support value of a rule represents the precision and recall rate respectively for that rule. Section 5.1.1 (Usability and implementation under operational settings) describes the calculation of precision and recall rate based on collection of rules for a specific failure. It also describes the acceptable trade-off values for the precision and recall rate for failure prediction model constructed based on these rules.

4.2 Classification Evaluation

The evaluation metrics that were used in evaluating the performance of the classifiers are elaborated in this section. Although all the evaluation metrics such as Mean Absolute Error, Accuracy, F-Measure were used to evaluate the performance of a classifying model, only selected metrics is used in the report.

Confusion Matrix

A confusion matrix[18] contains information about actual and predicted classifications by a classification system. Performance of such systems is commonly evaluated using the data in the matrix. The following table shows the confusion matrix for a binary classifier which is used in prediction in this project. The confusion matrix is shown in Table 4.1.

True Positive, False Positive

True positive refers to the case that the classifier predicted a failure to be a failure. False positive represents the case that the classifier predicted that a class to be a failure when it is not. The rate of False positives is directly proportional the amount of false alarms raised. Similarly, a True Negative represents the case that the classifier predicted that a failure would not happen at time t, and the failure did not happen. False Negative is when the
classifier predicted that a failure would not happen at $t$ but the failure happened.

**Precision, Recall**

Precision of a positive class refers to the fraction of the failure that is predicted to be positive is actually positive. Recall refers to the amount of failures that are detected. Section 5.2.1.2 describes acceptable trade-off values for the precision and recall rate for failure prediction classifying model.

**ROC Space**

The ROC [19] space is constructed by plotting a graph between the True positive and False Positive Rate.

The true positive rate is defined as

$$TPR = \frac{\#TP}{\#TP + \#FN} \quad (4.1)$$

The false positive rate is defined as

$$FPR = \frac{\#FP}{\#FP + \#TN} \quad (4.2)$$
The ROC space is used to represent the trade-off between deducting the failure and the false alarm. In a perfect case low values of false positive and high values of true positive is preferred. In other words, the values plotted at the top left corner of the graph are favoured.

False positives are considered to have huge impact in real time in this case, since a repair team would be scheduled for every error prediction to rectify the fault beforehand. A true positive implies that an error is correctly deducted as an error and represents the percentage of failure that is predicted. Figure 4.1 shows an example instance for the calculation of TP, FP, TN and FN in streaming settings.

Figure 4.1: Example Instance - Calculation of TP, FP, TN and FN in streaming settings

### 4.3 Event Frequency Statistical Modelling Evaluation

The goodness of fit tests for EFSM could be performed using the standard deviation of fitness and the standard error calculated for a specified distribution.

The Anderson-Darling(AD)[16] test could be used to perform the goodness of fit for specific distribution. This is performed by defining the following hypothesis.

\[ H_0 : \text{The data follow a specified distribution.} \]

\[ H_a : \text{The data do not follow the specified distribution} \]
The AD test statistic (A) is calculated for distribution being tested. The null hypothesis \( (H_0) \) is rejected if the AD test statistic, A, is greater than the critical value which is calculated for each distribution.

The AD test is preferred over other goodness of fit tests such as Chi-Squared test since AD is distribution specific. The disadvantage with the chi-square test is that it is not valid for small samples, and the approximation obtained for the chi square test would be invalid. For instance, if the number of occurrences of a specific event type is less between two consecutive failures, the sample size for that particular list would be reduced which eventually affects the test value.

### 4.4 Evaluation metrics in operational settings

The following Table 4.2 explains for each strategy, the important metrics that are used to evaluate the strategy and their relevance in operational settings.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Result</th>
<th>Metric</th>
<th>Operational Settings Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>RM</td>
<td>Rules</td>
<td>Confidence of Rules</td>
<td>Confidence directly reflects failure probability upon the occurrence of an item in a rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lead time of Rules</td>
<td>Lead time reflects the minimum time after which a failure is likely to occur upon the occurrence of an item in a rule</td>
</tr>
<tr>
<td>Strategy</td>
<td>Result</td>
<td>Metric</td>
<td>Operational Settings Relevance</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>--------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>BRM</td>
<td>Rules</td>
<td>Batch Confidence of Rules</td>
<td>Batch Confidence reflects the number of batches that satisfied the confidence threshold $c_r$ for an item in the rule. Higher the batch confidence, higher is the probability that a failure will occur. Lead time reflects the minimum time after which a failure is likely to occur upon the occurrence of an item in a rule.</td>
</tr>
<tr>
<td>RSS</td>
<td>State Space of Rules</td>
<td>Metrics used for RM and additional global metrics</td>
<td>Involves the metrics used for RM and additional global metrics generated for the state space. In operational settings, the FP value of individual states could reflect the failure probability upon the occurrence of the state.</td>
</tr>
<tr>
<td>FBC, RBC</td>
<td>Classifier Model</td>
<td>Precision of Positive Class, Recall of Positive Class</td>
<td>Refers to the fraction of the failure that is predicted to be positive is actually positive. A high precision value is important in operational settings to reduce the amount of false alarms raised by the classifier. Refers to the fraction of the failure that is deducted. A recall value of 100 percent means that all the failures are detected.</td>
</tr>
<tr>
<td>EFSM</td>
<td>Statistical Model</td>
<td>Lead time</td>
<td>Refers to how early predictions could be made.</td>
</tr>
<tr>
<td>Strategy</td>
<td>Result</td>
<td>Metric</td>
<td>Operational Settings Relevance</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>--------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Threshold limit</td>
<td>Ratio of the number of times a particular event increased its frequency satisfying a specific distribution before failure to the number of failure occurrences. Higher the threshold value larger is the probability that a failure will occur if the event is found to satisfy corresponding statistical distribution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Goodness of fit tests</td>
<td>Represents how well the increasing frequency of events fit a specific distribution</td>
</tr>
</tbody>
</table>

Table 4.2: Evaluation of Techniques and their relevance in operational settings
Chapter 5

Experiments

This chapter discusses the results obtained for each strategy. The results are grouped as per Pattern Mining, Classification and EFSM. Section 5.1 describes experiments and corresponding results performed in pattern mining. This is followed by the experiments and results obtained for classification in Section 5.2 and IFDT in Section 5.3. All the experiments performed in various strategies except EFSM were conducted using the log data obtained after temporal compression (Chapter 2). For EFSM, the experiments were conducted on the original log data of 25 million records.

5.1 Pattern Mining

In this section the experiments conducted in pattern mining are elaborated. This includes specific parametric settings, respective results, their applicability in operational settings and DSS. The comparison of RM and BRM strategies is performed in Section 5.1.3.

5.1.1 Rule Mining

The RM strategy (Section 3.2.1) was experimented with varied input parameters for lead time $\Delta lt$, time window $\Delta tw$, confidence $c$ and support $s$. Figure 5.2 shows the results for the varied $\Delta lt$ with a constant $\Delta tw$ of 180 minutes, $s$ of 0.10 and $c$ of 0.25.

The reasoning behind the $c$ value of 0.25 is that the machine operations
Experiments

are real time and various modules deployed inside the machine have inter
dependencies. Because of this, it is possible that a failure could be caused by
collection of variables. Although certain variables are an important factor in
triggering the failure, the variables would not guarantee failure occurrence
consistently when they happen. By setting a \( c \) value of 0.25, it is not only
possible to find rules with high correlation to failures but it is also possible to
find rules that have a relation to failure but does not trigger failure always.
In real time systems, even a rule with only 25 percent failure probability is a
valuable information which could be used in operational settings to analyze
the potential cause of a failure in addition to failure prediction.

Confidence level of 0.25 corresponds to a precision level of 0.25 in which case,
3 in 4 failure predictions raise false alarms. This is an acceptable limit in
pattern mining since the failure alarm raised would have explicit information
of the event that relates to a failure which makes it possible to analyze the
alarm for false positives. Increasing the length of \( tw \) would capture machine
characteristics over a long period of time. From the results it was observed
that increasing the length of \( tw \) did not increase the number of rules generated
since majority of the rules occur close to the failure. Analysis of descriptive
statistics collected for each rule reiterated this scenario that the MTTF and
the SD of MTTF values for each rule concentrated on the range of 3000
seconds from failure on an average.

The support level of 0.10 implies that the rules should be mapped with at
least 10 percentage of the failures occurrences. Increasing the support value
produces rules with very high recall value. However, a support value of 0.10
is exhaustive and inclusive of all the rules that would be generated for high
support values. A rule with a support value of 0.10 but with very high
confidence greater than 0.75 is still usable in operational settings rather than
having only rules with very high support value. This scenario is shown in
Figure 5.1 where two distinct events E1 and E2. The event to failure rule
for E1 and E2 has a confidence of 1.0. The support value of E1 and E2 are
0.5 and 0.25 respectively. Although the support is low for E2, it increases
the recall value overall for the failure to 0.75 since 6 out of 8 failures are
uniquely matched. This is due to the mutually exclusive mapping of failures
with respective events.

Observations: Important observations from RM are as follows

- Machine characteristics are real time and instantaneous such that in-
  creasing \( \Delta lt \) decreases the number of rules generated rapidly. This
Experiments

Figure 5.1: Example Instance - Support, Recall relation in RM

![Graph showing the support and recall relation in RM for various dates.]

Figure 5.2: RM - Number of rules obtained for each failure for varied lead time $\Delta lt$ values with a constant time window, $\Delta tw$ of 180 minutes, support level, $s$ of 0.10 and confidence, $c$ of 0.25.

scenario could be observed from Figure 5.2. The number of rules generated with a lead time, $\Delta lt$ of 0 (that is instant rules) is on an average more than twice the number of rules generated for a $\Delta lt$ of 5 minutes. This shows that the machine characteristics are highly real time that in majority of the cases as shown in Figure 5.2 for a $\Delta lt$ of 0 minutes, the failure is triggered immediately after the occurrence of the items.

- Failures are distinct with different predictive capabilities. Certain failures for the same input parameters have better results than others in terms of the number of rules generated. For instance, failure type LO-8422, although decreases exponentially with increasing $\Delta lt$, it has better results in terms of number of rules generated for a $\Delta lt$ of 30 minutes than for other failures which are nil or very few. At the same time, failure type LO-8459 with only 33 occurrences in total does not
Experiments

have rules generated for $\Delta lt$ greater than 0.

- Except LO-8459, all other failures have on an average more than 25 rules generated for a $\Delta lt$ of 10 minutes. Also, except LO-8422 with two rules, none of the other failures have rules generated for a $\Delta lt$ of 1 hour. For a $\Delta lt$ of half an hour, although five failures have rules generated, only LO-8422 has 16 rules generated. The rest of the failures have less than 3 rules generated.

- The results obtained for the failure LO-8422 is better than the rest as it has 16 rules generated for a $\Delta lt$ of half an hour which suggests that the machine shows signs of failure and characteristics changes before half an hour for LO-8422.

![Graphs showing frequency of rules generated across confidence levels for different failures.](image)

Figure 5.3: RM - Frequency of rules generated across confidence levels obtained for each failure for a lead time, $\Delta lt$ of 5 minutes, time window, $\Delta tw$ of 180 minutes, support, $s$ of 0.10 and a confidence, $c$ of 0.25.

Figure 5.3 and Figure 5.4 shows the number of rules generated across confidence levels for each failure with a $\Delta lt$ of 5 and 15 minutes respectively for varied parameter values for $\Delta tw$ of 180 minutes, support, $s$ of 0.10 and a confidence, $c$ of 0.25. These figures could be used to compare the change in confidence levels of the rules generated for increasing $\Delta lt$ values. It also be used to analyze the distribution of confidence values for rules generated for their respective $\Delta lt$ values. The distribution of confidence values for the rules generated with other experimented $\Delta lt$ values is shown in Appendix B.

The observations from Figure 5.3 are as follows.
• Majority of the rules were generated with a confidence range of 0.25 to 0.5 which suggest that even for a $\Delta lt$ of 5 minutes, comparatively not many rules were obtained with high confidence values (greater than 0.75, for instance) that suggests strong correlations. However, on an average for a failure type except for LO-848F, there are more than 5 rules obtained with high confidence values in the range of 0.70-1.0.

• Failure LO-848F has no rules generated with a confidence higher than 0.45 for a $\Delta lt$ of 5 minutes. No rules obtained with high confidence levels suggest that machine characteristics do not change considerably for LO-848F even five minutes before its occurrence.

• For a $\Delta lt$ of 15 minutes except for LO-8422 and LO-8434, none of the failures have rules generated with confidence values greater than 0.50. This suggests the change in machine behaviour increases as we move towards the failure. In other words, as we move away from the failure and increase $\Delta lt$ value, the observed change in machine behaviour is less.

• LO-8422 has comparatively consistent performance in terms of confidence levels of the generated rules for the $\Delta lt$ of 5 and 15.

The rules were generated along with the descriptive statistics. This as explained in Section 3.2.1 could be used as a factor in deciding what is the
normal MTTF when corresponding rules occur. MTTF in addition with other statistics would be helpful in operational settings to roughly estimate the generic historical behaviour of failure occurrence after the occurrence of items in the rules. The following descriptive statistics regarding the rules were generated.

- **MTTF**: Mean Time to Failure in seconds was generated for each rule. MTTF value will always be higher than \( \Delta lt \) value since the \( tw \) as described in Section 3.2.1 is defined before \( \Delta lt \) and only the rules inside \( tw \) is captured. This allows an estimated time defined in \( \Delta lt \) before which the failure will not happen.

- **Standard Deviation and Skewness of MTTF**: For each rule, the standard deviation from MTTF information was generated. Skewness of MTTF generated for MTTF of each rule aids in analyzing asymmetry of MTTF.

- **Range(Max-Min)**: The range value describes the range of failure occurrences in seconds. The range value is obtained for each rule by considering the time to failure value of every single match of the item in the rule with the failure. The maximum and minimum value in the set were extracted. This in operational settings gives an estimate of what was the longest and shortest possible time to failure as observed in the past.

- **Number of Matches**: This value corresponds to the number of matched occurrences of an item in the rule with the failure. The number of matches for a particular rule directly relates to the support and recall value. The relation between support and how it contributes to recall value of failure prediction is shown in Figure 5.1.

Figure 5.5 shows MTTF values grouped per range generated for the rules for each failure with a \( \Delta lt \) of 5 minutes, \( \Delta tw \) of 180 minutes, \( s \) of 0.10 and a \( c \) of 0.25. The MTTF value for some of the failures have few occurrences in the range of 8000. However, these rules have high standard deviation for MTTF as well. Due to this reason, rules with 8000 MTTF were not generated for increasing \( \Delta lt \) values. Therefore, it is important in operational settings to consider the MTTF values of the rules by also taking into account the skewness and standard deviation of MTTF.

**Usability and implementation under operational settings**: The steps involved in failure prediction under operational settings using RM is shown in Figure 5.6. The individual rules generated for specific failures across varied lead time values are filtered based on the precision and recall capability...
Experiments

Figure 5.5: RM - Frequency of MTTF of rules grouped to respective bins obtained for each failure for a lead time, \( \Delta t_l \) of 5 minutes, time window, \( \Delta tw \) of 180 minutes, support, \( s \) of 0.10 and a confidence, \( c \) of 0.25.

of the rule and used for failure prediction of the corresponding failure. The objective is to find the list of rules that together maximize the precision and recall value for failure prediction for the respective failure.

Figure 5.6: Failure Prediction using RM under operational settings

The present analysis of the cause of failure and remedial action followed by domain experts is by going through the latest log file data and finding potential cause for the failure manually. Depending on the failure, this process could take approximately any where from 1 hour to 1 day on an average.
However, since the failure prediction model would be implemented upon the
rules generated from RM, a failure alarm is raised when the rule is encoun-
tered. It is possible to validate the correlation of the specific event to failure
(along with the descriptive statistics generated with each rule) in approxi-
mately 2 to 3 minutes since it does not involve going through the entire log.
Therefore, the amount of time used in analyzing false alarms would be less
compared to a black box failure prediction model which does not generate
explicit rules but raise failure alarms. Hence, the amount of time spent in an-
alyzing each false alarm is less. This increases the tolerance of total amount
of false alarms acceptable which subsequently leads to usability of failure
prediction models with low precision rate compared to black box prediction
model.

It is assumed that for a specific prediction model, an operator could spend on
an average under approximately 2 to 3 hours of time in false alarm analysis.
An amount of approximately 2 to 3 hours is assumed since there are possi-
bilities of using multiple strategies each with multiple models with specific
parameter settings and failure alarms raised for all the failures would in-
crease the amount of time spent in false alarm analysis for an operator. For
instance, a prediction model constructed from rules by considering a lead
time of 10 minutes is different from a prediction model constructed for a lead
time of 15 minutes. It is also possible that the company could deploy a team
of operators on the operational site exclusively monitoring the failure predic-
tion models. This decision of number of operators that could be deployed for
monitoring future failures would be influenced by various factors including
but not limited to the health of the machine, number of days the machine has
been in production, age of the machine, cost incurred by respective failures.
As understood from domain experts, presently, there are no teams or person
deployed at operational site exclusively for analyzing the failure prediction
models to monitor potential future failures. If the company deploys a team of
operators rather than one operator, it is possible to use models with reduced
precision rates since there would be multiple operators analyzing the failure
alarms and with that the tolerance of the amount of time spent in analyzing
false alarms could be increased.

There are on an average 3.3 failure occurrence per day. It could be consid-
ered that a precision of 5 percent is usable in RM with a high recall value
approximately equal to 0.75. With 0.75 recall rate approximately 2.5 fail-
ures are detected each day. With 5 percent precision rate, 19 false alarms
are generated for every one correct prediction. For 2.5 failure detections,
47.5 false alarms are generated per day for the corresponding model which
approximately relates to 142 minutes of time spent on false alarm analysis.
Experiments

considering an analysis time of 3 minutes each for a failure alarm. The tolerance of precision rate threshold is also influenced by recall rate of the failure prediction model. For instance, for low recall values, the precision rate could be reduced since the number of failures detected are less and with that the amount of false alarms raised would be reduced. In this case, a failure prediction model constructed for a failure by considering the set of rules for a specific lead time that satisfy a threshold precision rate of 0.05 with a recall of approximately 0.75 is considered usable in operational settings. However, irrespective of the recall rate, the best possible set of rules with corresponding recall rate is still usable provided if it satisfies the threshold time limit assumed per operator on false alarms analysis.

In the experiments conducted in RM for a log duration of approximately 440 days, rules were generated for specific failures with varied lead times between 5 minutes to 1 hour. Rules created are set to satisfy a confidence level greater than 0.25 which implies that, a rule has occurred at least on the 25 percent of past failure occurrences. This corresponds to a precision of at least 0.25 which corresponds to three false alarms on four failure predictions for that specific rule. The number of false alarm decreases with increasing confidence values. For instance, for rules with confidence levels of 0.50, one false alarm is raised for every two failure predictions. False alarms also increase based on the cumulative confidence levels of individual rules employed to predict a specific failure.

The precision and recall values for the rules are calculated as follows:

- **Precision:** The confidence value of a specific rule directly reflects the precision of the rule. The expected overall precision rate for failure prediction of a specific failure is the cumulative confidence values of all the rules employed to predict the failure.

- **Recall:** Recall for a specific rule directly reflects the number of unique failure to event mappings found for the rule. The overall recall value for failure prediction of specific failure is improved by employing a set of rules that have maximum mutual exclusive property obtained for a specific failure for a specific lead time. The mutual exclusive property in this case could be defined as the number of unique failure mappings performed by considering the rules generated for the corresponding failure. This is shown in Figure 5.7 where the number of failure (F) mapping for the rules E1, E2 and E3 are 3, 2 and 2 respectively. The total number of unique failure mappings considering all the rule is 6.
Challenges: Construction of a failure prediction model based on collection of rules from RM for specific failures and specific lead times, includes selection of rules. This involves the following challenges:

- It is not possible to include all the rules generated for a specific failure since the inclusion of multiple low confidence rules will reduce the precision rate overall in the model considerably which increases the number of false alarms.

- A threshold level could be set for filtering low confidence rules. This however will filter out rules having high recall (support value of a rule represents its recall value) and low precision values. This could be eliminated by setting a threshold level for both confidence and support values. Thus, only the rules which satisfy a specific confidence and support values would be selected for the prediction model. But there are cases, in which multiple rules that have overlapping mapping to the same failure points in which case although the best rules with high support values are selected, it would not represent an optimal solution for recall value.

This scenario is shown in Figure 5.7 in which both E2 and E3 have the same confidence and support value of 1.0 and 0.28 respectively. According to the figure, E1 has the highest support value of 0.42 followed by E2 and E3. Since E2 has a overlapping failure mapping with E1, E3 should be favoured to E2 as E3 has mutually exclusive mappings to failure, F. The generalization of this scenario would mean that if multiple rules with the same failure mappings satisfy a support threshold all those rules would be selected which actually would not increase the recall value. This involves analyzing individual time stamp mapping values to failure of each occurrence of every event in the collection of rules.

![Figure 5.7: Improving Recall values in RM](image)

Due to the aforementioned reasons, the construction of failure prediction model based solely on the rules generated by pattern mining was considered as a potential possibility for future work. However, the predictive ability
Experiments

of these rules extracted for individual failure types have been established through experiments performed in RBC. The collection of rules generated for each failure type are used in the construction of feature vectors for RBC.

**Conclusion:** It could be concluded from the experiments performed in RM strategy that for increasing $\Delta lt$ values, majority of the rules tend to have confidence levels on the range of 0.25 to 0.50. Few rules were observed with high confidence levels for a $\Delta lt$ of 15 minutes. The confidence of the rules generated for a $\Delta lt$ of 30 minutes, for the failure LO-8422 has rules in the range of 0.40 to 0.50 which is high compared to the decreasing trends of confidence levels observed in other failures. These rules could be used to construct a failure prediction model to predict LO-8422 half hour before with a considerable precision levels in the range of 0.40 to 0.50 which has high usability in operational settings. Four other failures have less than 3 rules generated for a $\Delta lt$ of 30 minutes which could be used to predict the respective failures half hour before with a confidence value at least greater than 0.25 in operational settings.

For other failures except LO-8459, number of rules that satisfy the confidence level of 0.25 were generated (Figure 5.2) for a $\Delta lt$ of 15 minutes. These rules could be used to predict respective failures at least fifteen minutes before its occurrence. In addition, under operational settings, these rules could also be used to identify the cause of the failure once the failure occurs which in real time systems is a highly regarded information.

5.1.2 Batch Rule Mining

The BRM strategy (Section 3.2.1) was experimented with varied input parameters for lead time $\Delta lt$, time window $\Delta tw$, confidence $c_r$, batch confidence $c_b$, distance separation $\Delta d$ and support $s$. The results for the varied $\Delta lt$ with a constant $\Delta tw$ of 180 minutes, $s$ of 0.10, $c_r$ of 0.50, $\Delta d$ of 24 hours and $c_b$ of 0.25 is shown in Figure 5.8.

The reasoning behind the confidence, $c_r$ of 0.50 as compared to confidence, $c$ value of 0.25 in RM is that since the $c_r$ is computed locally to each batch setting a low value for $c_r$ decreases the strength of the rules generated in a batch. This scenario when considered globally will produce rules which are not strong. The batch confidence $c_b$ level of 0.25 was chosen since there are multiple batches generated for each event and even if 25 percent of the total batches pass the threshold $c_r$ for a single rule, the rule is be considered to be related to a failure.
Experiments

Observations: Observation from BRM are as follows

- As observed from the experiments conducted in RM (Section 5.1.1), it was found that machine characteristics are real time and instantaneous that increasing $\Delta lt$ decreases the number of rules generated rapidly. This scenario is visible from Figure 5.8. The number of rules generated with a lead time, $\Delta lt$ of 0 is on an average three times more than the number of rules generated for a $\Delta lt$ of 5 minutes. This difference in the number of rules for BRM generated between $\Delta lt$ of 0 and $\Delta lt$ of 5 is much higher than RM. This suggests that BRM is comparatively better than RM in capturing instant machine characteristics with a $\Delta lt$ of 0 and a $\Delta lt$ of 5.

- Except LO-8459, all the other failures have on an average less than 25 rules generated for a $\Delta lt$ of 10 minutes. Unlike the results from RM, except LO-8422, all the failures have only very few rules generated for a $\Delta lt$ of 15 minutes. For a $\Delta lt$ of half an hour, four failures have rules generated of which LO-8422 has 9 rules and the rest of the failures have one or two rules.

- The results obtained for the failure LO-8422 is better than the rest as it has 9 rules generated for a $\Delta lt$ of half an hour which suggests that the machine shows signs of failure and changing characteristics before
Experiments

half an hour for LO-8422.

- It could be inferred from Figure 5.8 that most of the failures have none or one to two rules generated with a △lt greater than 15.

Figure 5.9: BRM - Frequency of rules generated across batch confidence levels, cb obtained for each failure for a △lt of 5 minutes, △tw of 180 minutes, support, s of 0.10 , fixed confidence, c_r of 0.50, distance separation △d of 24 hour and fixed cb of 0.25.

Figure 5.9 shows the number of rules generated across confidence levels for each failure with a △lt of 5 minutes, △tw of 180 minutes, support, s of 0.10, distance separation △d of 24 hour and a confidence, c of 0.25. Figure 5.10 has identical parameters to Figure 5.10 except for the lead time, △lt which is 15 minutes in Figure 5.10. The distribution of confidence values for the rules generated with other experimented △lt values are shown in Appendix C. The observations are as follows.

- Majority of the rules were generated for a confidence level between 0.25 and 0.5 which suggests that even for a lead time, △lt of 5 minutes, not many rules are obtained with high confidence values. Similar scenario was also observed from the results for RM in Section 5.1.1.

- Failure LO-848F has no rules generated with a confidence higher than 0.40 for a △lt of 5 minutes.

- For a △lt of 15 minutes, none of the failures have rules generated with confidence values greater than 0.50. In contrast the results obtained
Figure 5.10: BRM - Frequency of rules generated across confidence levels, $c_b$ obtained for each failure for a $\Delta lt$ of 15 minutes, $\Delta tw$ of 180 minutes, support, $s$ of 0.10, confidence, $c_r$ of 0.50, distance separation $\Delta d$ of 24 hour and $c_b$ of 0.25.

for RM with a $\Delta lt$ of 15 minutes have rules generated for LO-8422 and LO-8434 with greater than 0.50 confidence levels.

- LO-8422 has comparatively better performance for the $\Delta lt$ of 5 and 15 than other failures.

- Most of the rules obtained for BRM as observed from Figure 5.9 and 5.10 have less confidence ranges compared to results obtained for RM as shown in Figures 5.3 and 5.4.

**Conclusion:** The rules generated for BRM for various failure types could be used in operational settings the same way as explained in RM. Similar to RM, it could be concluded that the rules obtained for LO-8422 with a $\Delta lt$ of half an hour has high usability in real time systems. Three other failures have one to two rules generated for a $\Delta lt$ of 30 minutes which could be used to predict the respective failures half hour before in operational settings. For most of the other failures, number of rules have been generated as seen from Figure 5.8 that satisfy the confidence level of 0.25. These rules could be used to predict respective failures at least fifteen minutes before their occurrence.
5.1.3 Comparison of RM and BRM

The results as shown in Figures 5.2 and 5.8 imply that RM strategy yields comparatively better result overall in terms of the number of rules generated and also the confidence levels of the generated rules. This scenario could be observed from Figures 5.4 and 5.10 which shows the distribution of confidence levels of the generated rules for a $\Delta lt$ of 15. The results of the RM strategy as shown in Figure 5.4 yields better confidence levels than BRM as shown in Figure 5.10. Similar trend was observed overall for higher $\Delta lt$ values. However, the BRM performs better than RM in terms of the number of rules generated for $\Delta lt$ values of 0 and 5. This could be observed from Figure 5.2 and 5.8 where more rules were generated for BRM than RM strategy for $\Delta lt$ values of 0 and 5. The distribution of the batch confidence levels of these rules generated from BRM for $\Delta lt$ of 5 revealed that the additional rules obtained were concentrated more on the confidence range of 0.25-0.35. Nevertheless, in operational settings of a real time system, it could be valuable information as discussed in Section 5.1.1.

Table 5.1 shows the comparison of RM and BRM strategies for the $\Delta lt$ values of 0, 5 and 15 based on the union (U) of rules generated from RM and BRM which contains the unique rules generated between RM and BRM, relative complement of BRM with respect to RM (RM) which gives the number of rules that were generated in RM but not in BRM and relative complement of RM with respect to BRM (BRM) gives vice versa. The comparison for other values of $\Delta lt$ between RM and BRM with respective comparison metrics is shown in Table C.1, Appendix C, Batch Rule Mining.

It could be inferred from Table 5.1 that the relative complement of RM with respect to BRM (BRM column in Table 5.1) is comparatively higher than the relative complement of BRM with respect to RM (RM column in Table) value for the $\Delta lt$ of 0 and 5. This implies that the number of unique rules generated by BRM is more than the number of unique rules generated for RM for the $\Delta lt$ of 0 and 5. Most of the rules obtained from RM strategy for the $\Delta lt$ of 0 and 5 was also captured by BRM strategy. In addition, many new rules were generated by BRM which is absent in RM for the respective $\Delta lt$ of 0 and 5. However, for increasing $\Delta lt$ values greater than 5 minutes, the RM strategy captures more rules and the number of rules generated by RM which is not present in BRM for the respective $\Delta lt$ is comparatively higher than the corresponding BRM value. The new rules generated in both RM and BRM for the same failure type implies that additional potential cause for the failure is established which would subsequently aid in better
It could be concluded from the comparison of RM and BRM strategies that both RM and BRM strategies are mutually exclusive and it is advantageous to include both in the decision support system as there are unique rules captured in both the strategies which are otherwise absent in the other for respective $\Delta lt$ values. RM strategy yields better results than BRM for increasing $\Delta lt$ both in terms of the number of rules generated and the distribution of confidence values of the rules which in RM contains more rules in the higher confidence ranges.

### 5.1.4 Rules State Space

The rules state space strategy as described in Section 3.2.1 was experimented with varied input parameters for FP threshold and lead time $\Delta lt$. In order to filter the state space for high values of NSP and FP attribute, a lower
Experiments

limit threshold level for these attributes to be satisfied was set. The NSP represents transition probability from the concerned state to next state as described in Section 3.2.1 and FP attribute represents the probability that a failure will happen from the concerned state. This threshold value was set based on the level of the states.

For instance, for the failure LO-8422, more rules were discovered in the high confidence levels. Therefore, the number of states that directly relates to the failure represented at level 1 could be filtered by setting a high threshold value which have to be satisfied by all the states at level 1. This high threshold value at level 1 is set based on the high confidence values obtained for the rules for LO-8422. This is to increase the value of NSP attribute for all the states at level 1. This guarantees that the states in the level 1 for which both FP and NSP attribute value is the same (Section 3.2.1) have high probability transition to failure and also limits the state space from exploding. The high probability threshold value which is to be satisfied for all states at level 1 also guarantees that the subsequent states have high values for FP attribute which is calculated by multiplying the NSP attribute value of all the states in the path from the target state to the failure.

The number of new states formed as a children under a specific state depends on the value of $\triangle lt$. As observed from the results of RM strategy in Section 5.1.1, more number of rules were generated for low values of $\triangle lt$. Therefore using a large global value for $\triangle lt$ value would not guarantee a RSS with more number of states with high probability transition values. Also, with increasing number of levels (i.e. increasing number of states in the path from leaf state to the failure state), the FP value decreases. To eliminate this situation, with each additional state added from a leaf state to the root state (failure), decreasing $\triangle lt$ values were used for each state at respective levels. The values of $\triangle lt$ were decreased linearly for each level until a lower bound threshold limit beyond which the states in successive levels are set the lower bound threshold value for $\triangle lt$.

The evaluation of a state space was performed as described in Section 4.1.3. The RSS developed could be evaluated based on the state level as well as global statistics regarding the state space. Table 5.2 shows the statistics for the RSS generated with a base $\triangle lt$ of 5 minutes. The base $\triangle lt$ is then reduced on increasing levels with a threshold limit of 2 minutes. Maximum MTTF and maximum MTTF state represent the largest value among all states for the attribute MTTF and the corresponding state message code respectively. Overall, similar trends as obtained from the results of RM was observed. For instance, the number of states generated for LO-8422 is too
Table 5.2: Rules State Space generated with a base $\Delta t$ of 5 minutes.

<table>
<thead>
<tr>
<th>Failure</th>
<th>Number of States</th>
<th>Maximum MTTF</th>
<th>Number of Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>LO-848F</td>
<td>13</td>
<td>4113</td>
<td>2</td>
</tr>
<tr>
<td>LO-0058</td>
<td>37</td>
<td>2222</td>
<td>3</td>
</tr>
<tr>
<td>LO-8537</td>
<td>48</td>
<td>2641</td>
<td>3</td>
</tr>
<tr>
<td>LO-8482</td>
<td>43</td>
<td>2482</td>
<td>3</td>
</tr>
<tr>
<td>LO-8422</td>
<td>286</td>
<td>5173</td>
<td>4</td>
</tr>
<tr>
<td>LO-8434</td>
<td>159</td>
<td>12566</td>
<td>4</td>
</tr>
<tr>
<td>LO-0052</td>
<td>30</td>
<td>2510</td>
<td>3</td>
</tr>
</tbody>
</table>

high compared to other failures.

Figure 5.11: Distribution of Failure Probability for a state space generated for each failure with a base $\Delta t$ of 5 minutes.

**Conclusion:** Figure 5.11 shows the distribution of FP obtained for each state under a state space generated of each failure with a base $\Delta t$ of 5 min-
Experiments

It could be observed that overall more states have FP value of less than 0.50. This trend was observed from the results obtained for RM and BRM. It could be concluded that overall the performance of RSS generally depends on the performance of RM since new states are generated by applying RM algorithm for each state. A specific failure with high predictability and the number of rules generated in RM (number of rules generated by RM for RSS reflects the number of states as shown in Table 5.2) for tend to have better results and applicability in RSS. This trend could be observed from the Table 5.2 and Figure 5.11 where more states were generated for failures which performed better with RM. A RSS in operational settings could be useful as it gives a path through which the failure is reached along with the respective failure probabilities. This gives an additional information of what are the possible causes for the failure occurrence once the failure happens. An XML file format was used to specify the list of states along with respective attributes. The directed edge information from the states are stored as nested elements in the XML file. An example instance of an XML file generated is shown in Appendix D.

5.2 Classification

In this section the experiments and results of classification is elaborated. Figure 5.12 shows different phases involved in the experiments conducted in this section. In Section 5.2.1, the experiments performed in FBC is explained. The experiments on RBC are detailed in Section 5.2.2.

5.2.1 Frequency Based Classification

In this section, the analysis performed by constructing the feature vectors based on FBC as described in Section 3.2.2 is presented and analysed.

5.2.1.1 Model Development Parameters

The classification model development parameters heuristics are as follows:

**Length, l** : The time length, l sets the length of the time window, tw. This is explained in detail in Section 3.2.2. Different l value such as 24 hours, 12 hours, 8 hours, 6 hours, etc were set and experimented.
Experiments

Figure 5.12: Classification Phases

Class Label Construction: Two different class label construction mechanisms were experimented as described in Section 3.2.2.

Data: Experiments were performed by constructing features vectors on the original data which contained 4206 unique message codes analyzed in Section 2.3 as well as the compressed data of 3106 unique codes. In the compressed data the most occurred 100 events and the least occurred 1000 events were removed. The most occurred 100 message codes occurred more than 90 percentage of the records (Section 2.5). The least occurred 1000 message codes have not occurred more than twice in the log data. The objective of carrying out experiments on the compressed data is to find if there is a change in classifier performance if the most frequent or least frequent message codes are removed. This is because there is a chance that the most frequent codes could influence and affect the construction of the model because of their comparative high values in each instance.

Algorithms: Several classifying algorithms were used in the experiments. The rules and tree based algorithms such as FT, J48 were analyzed. Since these classifiers work by constructing rules and trees, these could be further used to correlate the list of attributes used to predict the failure. The list of attributes in turn refers to the events which were used to predict. Further
algorithms such as Bayesian, RBF Networks and VFI were also experimented on. Apart from the specific algorithms, specific algorithms such as Bayesian, RBF and VFI were experimented with attribute selected classifier to ensure that some of the attributes which are more frequent and does not contribute to the prediction mechanisms are removed from the model.

**Sliding Window Strategy:** The sliding window strategy was used to find if there is a change in machine behaviour over a period of time. This strategy could be used to analyze the research question about the possibility of characteristic changes in machine behaviour over time as described in Section 1.3. This is performed by dividing the feature instances into multiple smaller chunks and training a classifier for each of them separately. The results obtained from individual classifiers for each chunk of a specific failure are evaluated and compared with the result obtained for the respective failure model by considering the original undivided instances. If there is a change in machine behaviour, the sliding window strategy by capturing the local changes over a small period of time should have predictive ability comparatively.

### 5.2.1.2 Interpretation of Results

The results obtained from the classifiers were evaluated based on the techniques described in Section 4.2. Experiments were performed by choosing different parameter heuristics and conducted on scaled and unscaled feature instances. But no considerable change in performance was observed. Therefore, in order to reflect operational values, the results presented here are based on the feature instances without normalization of instance values. Another reason for training the model by using unscaled feature instances is if the data was not normalized, it would be possible to decipher the rules created by rule based classifiers since the rules will include the original values (i.e respective number of occurrences of individual feature in $tw_i$) in their rules. This could be used for the correlation analysis of a feature with a failure.

Experiments performed on attribute selected classifiers yielded better performance results in terms of precision and recall value of positive classes. Therefore, all the results shown in this section are obtained by using respective algorithms on attribute selected classifier. Ninety percentage of the instances were used for training model and the remaining was used for model testing. The splitting order was preserved for training and testing the instances to emulate operational settings in all the experiments performed.
Experiments

Feature Instance Compression: The feature instances were developed as described in Figure 3.15 of Section 3.2.2. The importance of feature instances compression is explained in Section 3.2.2. The compression of feature instances reflects a change from operational settings in terms of the sequencing order of the instances as part of the negative instances between two successive positive class instances are eliminated as shown in Figure 3.15. However, this does not affect the performance of the classifiers since the classifiers build the model by considering individual information of each instance and does not consider the ordering sequence of the instances.

Baseline Classifier: Feature instances were developed by considering various values for compression factor, $n$. Results obtained from the feature instances constructed for a compression factor, $n$ of 2 were comparatively better in terms of precision and recall of positive class than for other $n$ values. For this reason, for all the results shown in this section, the feature instances developed with an $n$ value of 2 was used. This implies that, for every positive class instance, there would be two negative class instances. The $n$ value of 2 guarantees the ratio of positive to negative class instances at 1:2. Therefore, the baseline classifier that classifies all test instances to be a negative class would have an accuracy of 0.66 and a classifier that classifies all test instances to be a positive class would have an accuracy an 0.33. On an average, a failure occurs 182 times which implies that the total number of instances would be on an average 546 times. This makes the number of test instances on an average, 54. However there is a chance that certain failures occur multiple times in a single time window, $tw_i$ (Section 3.3.2). Hence the number of instances with the positive class is likely to reduce as the length of the time window, $l$ increases.

The evaluation on testing instances was performed on non-overlapping time windows as constructed in training instances. For testing the classifying models in streaming settings, prequential evaluation methods are usually applied. The prequential evaluation method is performed by performing testing on instances constructed by a moving window in time. These sliding windows are generally used to monitor the recent changes in behaviour and how it affects the failure. In other words, it access the evolution of the performance of models that evolve over time. However, in this thesis, the models were learned by considering a stationary non-overlapping window. The prequential evaluation is performed to understand the non-stationarity or concept drift properties of data streams. The concept drift means that the concept about which data is obtained may shift from time to time. The possibility of concept drifting in the log file data used in this thesis was analyzed using the sliding window strategy (in which experiments were performed sep-
arately on multiple smaller chunks of data formed by dividing the entire log data) which showed no considerable change in the performance of the models to other classifying models. For this reason, prequential evaluation methods were not considered and testing was performed on instances constructed from non-overlapping time windows.

**Interpretation of Precision/Recall Chart** : The results shown in this section are analyzed based on the precision and recall values of the positive class. The TP and FP values of positive class for the results analyzed in this section are given in Appendix D. The interpretation of precision, recall chart should be performed by considering the following.

- Precision and recall values used in the Precision/Recall bar chart are shown only for the positive class.
- The positive class refers to the presence of failure and predicting which is more substantial than the negative class.
- It is more important for the classifier to predict the positive class correctly and not raise a false alarm which in this case could be critical and in real time working conditions could even lead to undesirable effects since for every failure alarm, a dispatch team would be scheduled to analyze the potential fault.
- Precision refers to the fraction of the failure that is predicted to be positive is actually positive and recall refers to the amount of failures that are deducted.

Figure 5.13 shows the steps involved in failure prediction of specific failures in operational site for a classification failure prediction model. It was assumed that the amount of time required to analyze a failure alarm is approximately 1 hour since for each alarm raised from a black box failure prediction model, little or no information is associated for the possible cause of the failure. However, prediction models constructed based on rule or tree based classifying models could have the same trade-off as explained in Section 5.1.1 for RM. In the case of black box model, analyzing a failure alarm involves manual analysis of going through the recent log history. On an average, there are 3.35 failure occurrences per day. The average acceptable time spent for processing false alarms for a single prediction model per day is assumed to be approximately 2 to 3 hours. The reasoning behind the assumption is elaborated in Section 5.1.1 (Usability and Implementation under operational settings). Multiple classifying models that satisfy precision-recall trade-off could be used in which case, analysis time of false alarms will increase.
Figure 5.13: Failure prediction using classifying models at operational site

Figure 5.14 gives an example scenario of acceptable precision-recall trade-off values that could be used under operational settings. In an ideal case, high precision and recall values are favoured. Also, high precision and a low recall value is favoured than a low precision and a high recall value. This is because for high precision value, the rate of false alarms would be less compared to the low precision values. In this case, unlike pattern mining strategies where a confidence level of 0.25 for a specific rule and a precision value of 0.05 for a failure prediction model (built on rules) was considered usable, it is not productive under operational settings to have a precision of 0.25 with a high recall (approximately 0.75) value since no explicit patterns that are related to the failure would be perceived as a result of prediction (except in the case of rule based classifiers). In other words, the alarms raised by models would not give reasoning information for the failure unlike in pattern mining strategies where the events that is related to the failure would also be available in the form of patterns. Therefore, each alarm has to be analyzed by the domain experts for the cause and possibility of a potential failure. For instance, in the case of a prediction model with a recall value of 0.75 and a precision of 0.25, 2.4 failures would be detected per day which will generate approximately 7 hours of false alarm analysis for a single model. Hence it is better to have a high precision and low recall value rather than a high recall and low precision value in which case the number of false positives would be high. Therefore, a model which deducts at least 25 percentage of the failures but deducts it with a precision greater than 70 percent was considered to be productive under operational settings. Also, a model with a precision and recall value of approximately 0.50 each corresponds to 3 hours of false alarm analysis which is considered as acceptable. Depending on the number of operators and the number of models deployed for failure prediction, the
false alarm tolerance rate could be adjusted.

\[ 
\begin{array}{c}
\text{High Precision (} > 0.75 \text{)}
\end{array}
\]
\[ 
\begin{array}{c}
\text{High Recall (} > 0.75 \text{)}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Precision (} > 0.50 \text{)}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Recall (} > 0.50 \text{)}
\end{array}
\]
\[ 
\begin{array}{c}
\text{High Precision (} > 0.75 \text{)}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Low Recall (} < 0.25 \text{)}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Low Precision (} < 0.25 \text{)}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Low Recall (} < 0.25 \text{)}
\end{array}
\]

\[ 
\begin{array}{c}
\text{Acceptable Trade-off.}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Customer service will spend less than } ~ 1 \text{ hour a day in remedial actions of false alarms.}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Acceptable Trade-off.}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Customer service will spend less than } ~ 3 \text{ hours a day in remedial actions of false alarms.}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Acceptable Trade-off.}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Customer service will spend } ~ 1 \text{ hour a day in remedial actions of false alarms.}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Not acceptable Trade-off.}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Customer service will spend } ~ 7-8 \text{ hour a day in remedial actions of false alarms.}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Not acceptable Trade-off.}
\end{array}
\]
\[ 
\begin{array}{c}
\text{Customer service will spend } ~ 2-3 \text{ hour a day in remedial actions of false alarms.}
\end{array}
\]

Figure 5.14: False Alarms Acceptable Trade-off

5.2.1.3 Restricting Over-Fitting

A failure is considered to have predictive ability only if many classifying models developed for the failure under a specific parametric setting yield better results in terms of the precision and recall for positive class. This is because when models are developed under multiple parameter settings, it is possible that a specific model (in a set of multiple models constructed from various parameter settings) is likely to give usable results. The over-fitting of training a model with more feature instances is eliminated as discussed in Section 3.2.2 by compressing the feature instances. The same strategy is also used to balance the ratio of positive to negative class which otherwise would be too low to train a model. In addition to the above mentioned reasons, additional measures that ensure that the models and the results are devoid of data over-fitting are as follows. The instances used for the results shown in this section is not scaled and the classifiers were fed the original frequency count of respective instances. Since there was no parameter selection, the same operation settings were used for all the classifiers and no classifier specific parameter selection that would boost the performance of specific classifiers were used in the experiments described in this section.
5.2.1.4 Observations

Results obtained varied much in terms of precision and recall based on different parameter values for length $l$, of the time window $tw$ and also whether the prediction is performed on time window $tw_{i+1}$ or $tw_{i+2}$. For the same parametric values of length $l$, of time window $tw$, different failures yielded different results. In majority of the cases, no consistently better results in terms of precision and recall were obtained for all failures for a specific length $l$ parameter value.

The results for each failure with high value of $l$ is comparatively marginally better than the results set for decreasing $l$ values for both failure prediction on $tw_{i+1}$ and $tw_{i+2}$. This implies that as the length of the time window is decreased, the results obtained for the failures have less predictive ability on an average. Reason for this could be as the length, $l$ of the time window, $tw$ is increased, the characteristic of a machine in terms of the number of occurrences of a message code is observed for a long duration than for lower values of length, $l$. During this period, there is a chance that the increasing number of events observed could be used as a predictive ability. This scenario is shown in Figure 5.15 and 5.16 where the average precision and recall for positive class of top three classifiers with 12 and 1 hour time window for failure prediction of $tw_{i+2}$ performed with original data respectively is portrayed. The TP and FP values along with the results obtained for individual classifiers for respective parameters is shown in Appendix D. It could be observed that the results for Figure 5.15 is better than Figure 5.16. Overall the precision value is better than recall value which is preferred as described in 5.2.1.2.

The results observed in Figure 5.15 also shows that the precision for positive class on an average is 0.84 and the recall for positive class on an average is 0.32. This shows that the predictive capability of the classifiers with 12 hour time window for failure prediction of $tw_{i+2}$ is consistent in terms of precision for positive class. Unlike the results shown in Figure 5.15 consistent results across all classifiers for a specific length $l$ of time window, $tw$ was not observed for other length $l$ values for failure prediction on $tw_{i+2}$.

As explained earlier, varied results in terms of precision and recall were obtained based on different parameter values for length $l$, of the time window $tw$ and also whether the prediction is performed on time window $tw_{i}$ or $tw_{i+1}$. However, overall results for prediction on $tw_{i+1}$ is better than $tw_{i+2}$. The reason is that in the case of prediction for $tw_{i+2}$, the machine characteristics are observed earlier than the failure happens. Since the machine characteristics
Experiments

Figure 5.15: Average Precision, Recall for positive class of top three classifiers with 12 Hour time window for failure prediction of $t_{w,i+2}$ performed with original data.

Figure 5.16: Average Precision, Recall for positive class of top three classifiers with 1 Hour time window for failure prediction of $t_{w,i+2}$ performed with original data.

are mostly instantaneous and change little as we move away from the failure as seen in the results from RM in Section 5.1.1, the predictive capability for $t_{w,i+2}$ is generally less than $t_{w,i+1}$. This scenario is shown in Figure 5.17 which shows precision, recall of positive class for top three classifiers with length $l$ of 1 hour for time window $t_w$ for failure prediction of $t_{w,i+1}$ per-
formed with original data. The results obtained with identical parameters as in Figure 5.17 for \( tw_{i+2} \) is shown in Figure 5.16.

Figure 5.17: Average Precision, Recall of top three classifiers with 1 Hour time window for failure prediction of \( tw_{i+1} \) performed with original data.

Although for a failure, there were classifiers under specific parameters which performed well, all classifiers under the same parametric value did not yield good results. However, in few of the failures, comparatively better results were obtained for failures in multiple classifiers with different values for length \( l \) of the time window \( tw \). These failures performed well in multiple parametric setting which including prediction on \( tw_{i+1} \) and \( tw_{i+2} \). The failures LO-0052, LO-0058 and LO-848F which performed comparatively better than other failures are shown in Figure 5.18. The average of the top three classifiers under a specific parametric setting is shown for different length, \( l \) values.

Figure 5.18: Failures with comparative better precision and recall for positive classes across all parameters. Average of top three classifiers per length \( l \) value is shown.
Experiments

Figure 5.19: Average Precision, Recall of top three classifiers performed with original data for any parameter setting.

It was also observed that the removal of maximum occurred 100 message codes which constituted 93 percentage of the entire log and minimum occurred events (that occurred only once or twice) did not improve the performance of classifiers. Sliding window strategy for learning experimented with different window size for individual chunks did not yield comparatively better results than other strategies. This suggests that the machine behaviour does not change over time. The best precision and recall of positive class results obtained for each failure across all parametric setting is shown in Figure 5.19. The figure shows the average value for precision and recall of top three classifiers which performed better under a specific parametric setting. Considering the fact that the figure shows the best performance of top three classifiers involving any parameter setting, there are still failures with low recall value for positive class which makes these failure models inapplicable in real time operational settings for failure predictions.

It could be concluded that the predictive ability of a failure varied much depending on the parametric setting and the failures LO-0052, LO-0058 and LO-848F have better precision and recall rate comparatively in multiple parametric setting which could be incorporated in a decision support system. These failures shown in Figure 5.18 could be predicted on an average with a precision greater than 0.75 and a recall value of 0.39 with a prediction time...
that ranges from 1 hour to 24 hours before their occurrence.

5.2.2 Rules Based Classification

In this section, the experiments and analysis performed by constructing the feature vectors by considering the set of rules that were generated for each failure described in Section 3.2.2 is presented and analysed. All the rules generated for a failure with a $\Delta lt$ of 0 was experimented. The $\Delta lt$ of 0 was chosen since it captures all possible relations with respect to failure. The set of rules generated for a $\Delta lt$ of 0 would be the superset of all rules obtained for respective failures with increasing $\Delta lt$ values.

Varied experiments with various parameters settings were performed. The different parameter variations experimented with length $l$, class label construction and algorithms are discussed in Section 5.2.1.1. As described in Section 5.2.1.2, feature instances for decreasing length $l$ values were developed by considering compression factor.

![Figure 5.20: Average Precision, Recall of top three classifiers with 24 Hour time window for failure prediction of $tw_{i+1}$ performed for compression factor of 1](image)

Feature instances were developed by considering various $n$ values as compression factor. Results obtained from the feature instances constructed for an $n$...
value of 2 and 1 were comparatively better in terms of precision and recall of positive classes for other \( n \) values. The performance of a baseline classifier for an \( n \) value of 2 is described in Section 5.2.1.2. For an \( n \) value of 1, the baseline classifier that classifies all test instances to be a positive or negative class would have an accuracy of 0.50. The total number of test instances in the case of \( n \) with 1 is 35 on an average considering all the failures.

The results obtained for low values of length \( l \) of \( tw \) did not yield better precision and recall values for positive class. In most of the failures when the length \( l \) of \( tw \) is set less than 12 hours, the classifiers classify all the test instances as negative. The reason could be that when the length time window is decreased, the machine characteristics are observed for less amount of time during which the rules may not occur in respective time windows. However for increasing length values greater than 12 hours, the results are comparatively better. As observed from the results in Section 5.2.1, the prediction for \( tw_{i+1} \) yielded better results than \( tw_{i+2} \). The results obtained

![Figure 5.21: Average Precision, Recall of top three classifiers with 24 Hour time window for failure prediction of \( tw_{i+1} \) performed for compression factor of 2](image)

also varied based on the \( n \) values set as compression factor. Better precision and recall values are obtained for an \( n \) value of 1 compared to 2. This is shown in Figure 5.20 and Figure 5.21. It could be observed that better results than frequency based feature construction as described in Section 5.2.1 are
obtained. Although the results for a compression factor of 1 is better, when
the number of test instances was observed on an average it varied from 12-
16. Since for a compression factor of 1, the ratio of the positive to negative
class is 1:1, half the test instances are positive classes. Hence, very low recall
value with a high precision of 1 is not a usable result. However, when the
performance with compression factor of 1 is compared with a compression
factor of 2, the high performed classifiers still gave better precision and recall
that could be used in operational setting. The failures, LO-0058 and LO-
848F performed comparatively better than other failures in both Figure 5.20
and Figure 5.21 which suggest that these models could be used in real time
operational settings for failure prediction of these two failures on $tw_{i+1}$ with
24 hour length, $l$. The results for the failures LO-8537, LO-8482, LO-8434 and
LO-8422 also show usable results under operational settings. These failures
generate 30, 38, 40 and 40 false alarms respectively for every 100 failure
alarms and approximately 40 failures are detected. It could be concluded
from the results that most of the failures showed predictive ability with high
precision and recall rates for a compression factor of 1 and six failures showed
predictive ability which implies that the rules generated from RM strategy
could be used in operational settings for respective failure predictions.

5.3 Event Frequency Statistical Modelling

The EFSM (Section 3.2.3) was experimented with varied parameter values for
lead time $\Delta l_t$. The increasing frequency of an event could relate to a failure
instead of an one to one mapping of an event with a failure (Section 3.2.3).
In order to capture these type of relations, the original raw data log without
temporal correlations was used. The raw data analysis was performed in
Section 2.6 and the detailed statistical information is given in Table A.2.

For the decreasing IAT fit, Weibull distribution with a shape parameter of
1 to 1.4 was used. This corresponds to the decreasing sample rate close
to exponential. Therefore, the list of events that increase their frequency
exponentially before the occurrence of the failure would be captured in this
case. The experimentation was performed for a threshold limit of 75 percent
of the failures. This implies that the corresponding event should increase its
frequency of occurrence at least in 75 percentage of the failure occurrences.
The goodness of fit test was performed as described in Section 4.3.

The reason for a strong threshold level of 75 is that unlike RM and BRM
strategy where one to one mapping of failures and events were performed,
the frequency related correlation should be examined closely by the domain experts before it could be processed further. This is because, in the case of one to one mapping of event failure occurrences, the time window is closely monitored inside which an event occurrence is mapped to a failure. The descriptive statistics generated and the confidence levels obtained from the rules could be used as a strong correlation measure. But in the case of EFSM, the events are monitored for an arbitrary time period with no boundary condition except that the events are monitored between two successive failures. In order to eliminate the high chance of accidental correlations in an arbitrary time period, the threshold limit was set high.

In order to filter weak correlations, a threshold level for the number of IATs that should be present in each list was set. This means that if the event has occurred only very few times (in the order of less than 20) between two successive failures, it would not be considered for evaluation. This is because, the distribution fit performed on such small samples would not be taken as a strong correlation. For a lead time $\triangle lt$ value of 60 minutes, the results observed had many correlating events for each failure that satisfied the shape parameter value for the Weibull distribution. However, it was found that the standard deviation of fitness and the standard error calculated for the fitness was too high and deviated much from the shape parameter. None of the failures had correlating message codes that satisfied a threshold level of 0.75 and passed the goodness of fit test.

This could be due to the fact that the temporal compression had an effect only in the first quartile (based on the total number of occurrences) message codes of about 250 message codes as seen in Section 2.6. Moreover, the most occurred message code has been compressed more than 3 million times which implies that a single message has contributed to the increase in raw log data by more than 3 million records. Rest of the message codes, that are below the first quartile has occurred only less than 118 times. On an average a failure occurs 182 times, therefore for finding a correlation in the least case, a message code satisfying a threshold level of 0.75 should have occurred in 75 percentage of the times between two successive failures satisfying the threshold limit of 20 occurrences in each list which in total raises the frequency of the number of occurrences of a message code to be considered for EFSM to 2730. Only the most occurred 226 message codes satisfy this threshold. In addition, as certain failures occur in bursts, there is less time between two successive failures and the number of occurrences of a particular message code between two successive failures is reduced.
5.4 Decision Support System

In this section the construction of a DSS (Figure 5.22) is proposed. The necessity of an ensemble model is analyzed by comparing the mutual exclusive behaviour of different strategies with different failures. An ensemble model could include multiple models that work separately to predict failures of different types. Each model could be used to predict only the failures in which the model showed high predictability rate to avoid the number of false positives. Multiple models with varying lead time $\Delta lt$ values could be incorporated in a system to predict same failure if all of them have high predictability rate for that failure.

As discussed in the results in this chapter different strategies yielded different results for specific failures. It could be observed from the results obtained for RM and BRM strategies that there are additional rules discovered in BRM for the lower $\Delta lt$ values. This is shown in Table 5.1 where the relative complement of RM with respect to BRM is more for $\Delta lt$ values of 0 and 5. This implies that BRM is better than RM for $\Delta lt$ times less than 5. This
scenario is described in detail in Section 5.1.3. In addition, even for increasing values of $\Delta lt$ both RM and BRM yielded different results and each produced unique rules which were otherwise not discovered in the other strategy. This is shown in Table 5.1 of Appendix C for varying $\Delta lt$ values where unique rules that were discovered only in either RM or BRM is shown. The new rules discovered in both RM and BRM for respective $\Delta lt$ suggests that it is important to include both the strategies in a DSS.

The results of FBC for the failures LO-0052, LO-0058 and LO-848F showed comparatively better results in terms of precision and recall for positive class than other failures. However, for the failure LO-848F, the pattern mining strategy did not yield better results comparatively with other failures such as LO-8422 and LO-8537 both in terms of the number of rules generated and the confidence interval range of the rules generated. For the failures LO-0052 and LO-0058 there were less than two rules generated for $\Delta lt$ greater than 15. This suggests that both the strategies have different predictive ability for different failures and performs better with specific failures which are otherwise less-predictable in another strategy. For this reason, the classifying models as described for the failures shown in Figure 5.18 could be incorporated into DSS. The usability of incorporating these models in operational setting are discussed in the conclusion of Section 5.2.1.

In addition, results obtained for RBC with a time window of 24 hours and a prediction performed on $tw_{i+1}$ with a compression factor of 2 showed better results for the failures: LO-0058 and LO-848F comparatively. The usability of these results in addition with other classifying models constructed based on the rules of RM was discussed in (Figure 5.21). This could be included in DSS as suggested in the conclusion of Section 5.2.2.

To conclude, a DSS based on specific models obtained across all the strategies which have high predictability for a specific failure could be constructed and used. Figure 1.5 of Chapter 1 shows how DSS fits in the general framework under operational settings. The DSS is used by machine operators to predict machine failures. The evaluation and proposal of specific strategies for specific failures was performed which could be incorporated in DSS. The rules perceived in pattern mining could be helpful in a DSS for the analysis of the causes of failures as they contain the events which could relate to the failure. The set of rules obtained from pattern mining strategies with high confidence levels for specific failures could be used to construct a prediction model based solely on pattern mining which could be included in DSS. All the failures with an exception of LO-8459 have rules generated (RM strategy) for a lead time of 15 minutes and five failures have rules generated for a
lead time of 30 minutes which suggests that the failures could be predicted at least 15 minutes and in some cases half an hour before their occurrences with respective precision which is equal to the cumulative confidence levels of the rules employed. In the case LO-8422, two rules generated for lead time of 1 hour before failure with RM strategy could be incorporated into DSS. It was argued that the precision level could be set as low as 0.05. The predictive ability of these rules was established through RBC experiments performed in Section 5.2.2 which showed that results for the failures LO-0058 and LO-848F have precision and recall level of 0.82 and 0.45 on an average. In addition other failures have satisfying precision and recall rate which suggests that the rules obtained from RM strategy have predictive ability and could be used in DSS for failure prediction.

In the case of classification (FBC), although most of the failures have employable results with satisfying precision and recall level, comparatively better results for precision and recall of positive class were obtained for the failures LO-0052, LO-0058 and LO-848F. Hence, the classifying models for these failures with different length \( l \) of the time window, \( tw \) as shown in Figures 5.18 in the case of FBC could be incorporated to the decision support system.

5.5 Conclusion

In this chapter, the experiments conducted on all strategies as proposed in Section 3.2 were discussed. In pattern mining, experiments were conducted for different \( \Delta lt \) values to analyze the predictive ability of a failure as \( \Delta lt \) increases. It was shown from Figure 5.2 and Figure 5.8 that the machine characteristics are instantaneous and the number of rules decreased exponentially for decreasing \( \Delta lt \) values from 0 to 30 minutes. The rules were analyzed from the point of distribution of the confidence levels for respective failures in both RM and BRM. The comparison of RM and BRM strategy (Table 5.1) revealed that both the strategies produce new rules for the same \( \Delta lt \) value which suggests that they both are necessary to increase failure predictability rate in a decision support system. The comparison of RM and BRM strategy was performed to establish possible additional correlation of events to failure which were not generated in either of the other strategy. This additional correlation would boost the failure prediction rate for the respective failure since additional potential cause for that particular failure is established.

The experiments performed in FBC showed that for same parametric values,
Experiments

different failures yielded different results in terms of precision and recall. The feature instances constructed on the original data yielded better results than the instances constructed by eliminated the most frequent and least occurred items. The sliding window strategy which was constructed by dividing the entire log duration into multiple local chunks and performing experiments on individual chunks separately did not yield better results compared to experiments performed on the compressed instances which suggests that the machine behaviour did not change for the duration of log file performed in the experiments. The specific failures which yielded better precision, recall values for positive class in multiple parametric values for length \( l \) of the time window, \( tw \) shown in Figure 5.18 could be incorporated in DSS. In the case of RBC, comparative better results were obtained for a time window of 24 hours and a prediction on \( tw_{i+1} \) for the failures: LO-0058 and LO-848F. It was suggested that these models in addition with four other failure models could be incorporated in DSS.

Comparison of various strategies showed that it is more productive to build an ensemble model which incorporates multiple models. A decision support system based on ensemble model which incorporates multiple predictive models was proposed in Section 5.4.

Through the experiments, the research questions posed in Section 1.3 were analyzed. The results of the analysis for each of the research question are as follows.

- **Analyze the possibility of characteristic changes in machine behaviour over time.**
  The experiments performed in the sliding window strategy of FBC showed that the results obtained in terms of precision and recall did not show considerable change in machine behaviour compared to other strategies in FBC. It was identified in Section 2.3 that the number of errors decreased considerably over time. However, from the results of sliding window strategy, it was found that this did not affect the performance of the classifiers. It was observed that although the total number of errors logged per day decreased over time, the number of failure occurrences did not decrease over time. This implies that the cause for the failure never diminished over time. This is reiterated by the list of rules obtained for each failure in pattern mining strategies had matches with the failure occurrence dispersed all through the analyzed log which suggests that the reason for failure occurrence did not diminish over time.

- **Analyze the possibility of an adaptive model.**
In the analyzed log duration of approximately 440 days, no observable change in machine behaviour was found through the experiments performed (the rules generated by the rule mining algorithm had failure mappings across the log file duration). Due to this reason, the models need not be updated for the duration of approximately 440 days. However, to reflect the recent trends, prediction models could be updated in the future as in when there is a decrease in failure prediction rate.

- **Analyze the predictive characteristics of machine over a period of time and effectiveness of failure predictability as time before prediction increases.**

It was observed from the results in pattern mining that the machine characteristics are real time and instantaneous that in most of the cases failure happens immediately as soon as the items in the rule are perceived. The exponential decrease in the number of rules generated within a lead time of 30 minutes reiterated this fact. From the results, for all the failures excluding LO-8459, rules that satisfy a confidence level greater than 0.25 were obtained which suggests that the failure could be predicted before 15 minutes of its occurrence in most cases. Also for four other failures, prediction could be performed before half hour with a confidence satisfying 0.25. Apart from this various experiments from FBC and RBC classification showed that a failure could be predicted with arbitrary period ranging from 24 hours to 1 hour with an acceptable precision and recall rate. The prediction period included prediction on both $tw_{i+1}$ and $tw_{i+2}$.

- **Analyze whether specific strategies could be used for specific failure types based on its characteristics.**

As described earlier in this Chapter, different strategies showed different predictability rate for specific failures which shows specific strategy could be used for prediction of specific failures.

- **Analyze the possibility of creating an ensemble model.**

This was elaborately discussed in Section 5.4

- **In machine learning failure prediction by classification, study suitable ways of constructing feature vector instances.**

FBC and RBC strategies were proposed with different ways of feature vector constructing. In addition, various experiments performed with different parameter settings such as experiments on the original data, experiments on outliers removed data with the removal of 100 most frequent and 1000 least occurred message codes showed variations in feature construction.
Chapter 6

Conclusions

In this master thesis, the failure prediction of TWINSCAN machine by predicting the critical errors logged from LO module was experimented through various strategies. The goal of this master project was defined as follows:

*Problem Statement:* To propose failure prediction strategies for decision support system construction of TWINSCAN machine.

6.1 Main Contributions and Results

The predictive ability of failures were experimented on various strategies that were proposed in consideration of the research questions posed and the log file analysis. The analysis of the log file data showed various information regarding the distribution of message codes and failures. The two novel strategies based on the analysis of log file data were proposed. The burst like failure occurrence pattern observed led to the proposal of BRM strategy which captures burst like event failure patterns. The rules state space strategy was proposed as an extension of rule mining algorithm for analyzing the possible sequence of events which could trigger the failure. The comparison of results obtained from various strategies and their applicability in operational settings were analyzed.

The results obtained from pattern mining showed that machine characteristics are real time and instantaneous that the machine does not show early signs of failure. It was observed that results for pattern mining strategy were comparatively better than other strategies. The number of rules generated decreased exponentially with increasing lead time value. The results
Experiments

suggested that failure prediction for most failures could be performed using rules generated with confidence level greater than 0.25. For the failure, LO-8422, the prediction could be performed half an hour before the failure using rules generated with confidence levels ranging from 0.25 to 0.45. LO-8422 also had two rules generated for failure prediction with a $\Delta t$ of 1 hour. Five failures have rules generated for a $\Delta t$ of half an hour with a confidence level greater than 0.25 which suggest that these failures could be predicted before half an hour of their occurrence with a confidence level greater than 0.25.

The results obtained from classification varied for different parameter settings. The results through various parameter values were presented and discussed. Better precision recall results for positive class for the failures LO-0052, LO-0058 and LO-848F were perceived in multiple parameter settings in FBC. The applicability of these classifiers under operational settings was argued. In the case of RBC, most of the failures showed predictive ability although only the results of precision recall obtained for LO-0058 and LO-848F was consistent through various parameters. In Section 5.4, the various classifying models that could be used to predict failures was argued. The classifying models included failure prediction based on both $tw_{i+1}$ and $tw_{i+2}$ for time windows of 24 hours to 1 hour. This suggests that the failure prediction of these failures could be performed before 24 hours in the best case. In addition, through the analysis of the results, research questions were answered. The comparison of results showed that different strategies had different predictive ability for a specific failure. Certain strategies with better results for a specific failure was observed. Based on this, a DSS with an ensemble model incorporating different strategies was proposed.

6.2 Implication

It was understood from domain experts that since machine operations are real time, even prediction before 5 minutes of failure could be productive under operational settings. This is because the failure alarms are generated by triggering an alarm whenever the rules are encountered. Therefore, it is possible for a domain expert to use this information as a causal analysis for failure. The results from pattern mining could be used productively under operational settings to trigger failure alarms for respective failures by observing the rules generated which guarantees failure prediction at least before 15 minutes of failure for all failures but one.

Although machine showed little or no signs of failure even before one hour
of the failure occurrence in majority of the cases, the results obtained for classification (FBC) implied that overall on an average, for three failures (Figure 5.18), the number of false alarms generated for 100 failure alarms were 25 and the number of failures that are detected are 39 for every hundred failures. The respective classifying models of these failures could be used in operational settings since the precision rate is high and 39 failures for every hundred failures are detected.

In case of RBC, the failures, LO-0058 and LO-848F performed comparatively better than other failures (Figure 5.21). These failures have precision and recall level of 0.82 and 0.45 on an average. The results for other failures: LO-8537, LO-8482, LO-8434 and LO-8422 generate 30, 38, 40 and 40 false alarms respectively for every 100 failure alarms and approximately 40 failures are detected. These models could be incorporated in DSS under operational settings.

In addition, the descriptive statistics generated for pattern mining could be useful under operational settings to understand the normal time duration during which a failure would happen upon the occurrence of events in rules. The inclusion of both RM and BRM strategies mean that the strength of the DSS in terms of prediction accuracy for a failure of particular type increases since it was found that both these strategies are mutually exclusive and had certain unique rules generated which were otherwise not present in the other.

6.3 Limitations and Future Works

As suggested in Section 5.1.1 (Usability and Implementation under operational settings), methods to establish the selection of rules to be employed in constructing a failure prediction system based solely on pattern mining rules could be examined. In this thesis, the predictive ability of the rules generated for various failures were analyzed through RBC strategy.

The rules and decision trees generated by the classifying models such as J48 in RBC and FBC could be analyzed further. This analysis could lead to a potential correlation between the failure and the set of features (message codes) that were used to predict the failures.

The sliding window strategy used in FBC could be extended further to all other strategies to observe the possibility of characteristic change in machine behaviour in terms of causes that trigger the failure. As analyzed from the log file, the distribution of failure occurrences was dispersed all through the
analyzed log duration. Although, the failure never diminished over time, the possibility of specific causes that trigger failure could diminish over time and a new motive for the same failure occurrence could emerge over time. This scenario could be investigated by dividing the log file data into multiple non-overlapping parts and performing the experimentation for all the strategies in each of these parts separately.

The DSS and the ensemble model proposed in Section 5.4 could be implemented incorporating the suggested strategies for respective failures. The ensemble model implemented could also contain the failure prediction model built on rules generated for specific lead times for various failure types. The predictive power of these strategies under operational settings could be further examined by deploying the built system.

The pattern mining strategies used in this thesis concentrated on establishing rules based on one to one occurrence correlation of events to failures. This could be experimented further by analyzing correlation possibilities of multiple occurrences of events to a failure. It is possible that a certain event could happen multiple times and trigger a failure at its $n$th occurrence in a given time interval. These correlation behaviour could be established by considering many to one occurrence correlation of events to failures in RM. In this case, the increasing events frequency distribution fit experimented would fail to capture specific occurrences of an event triggering a failure since it captures only the increasing frequency of an event relating to failure and not the precise number of occurrences of an event causing a failure. The EFSM strategy could be experimented with different distributions in addition to Weibull. Non-linear regression fit could also be performed to capture increasing frequency of events triggering a failure. The log files generated from multiple machines of similar machine type (as analyzed in this thesis) could be experimented to reiterate the results obtained.
References


References

of the Eleventh International Conference on Data Engineering (ICDE 95), Taipei, Taiwan, pp. 314.


Appendix A

Log File Analysis

Log File Structure

An example instance of a log file data is shown in Figure A.1. The columns shaded in blue shows the considered attributes for the analysis.

Figure A.1: Example Instance - Log File
Appendix A, Log File Analysis

Log File Statistics

The statistics of log file data after temporal compression and the raw log data statistics are shown in the following Tables A.1 and A.2 respectively.

<table>
<thead>
<tr>
<th>Earliest Date considered</th>
<th>January 04 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latest Date considered</td>
<td>March 22 2012</td>
</tr>
<tr>
<td>Total records after filtering</td>
<td>14 Million</td>
</tr>
<tr>
<td>Distinct EVENTS</td>
<td>402</td>
</tr>
<tr>
<td>Distinct WARNINGS</td>
<td>456</td>
</tr>
<tr>
<td>Distinct ERRORS</td>
<td>3315</td>
</tr>
<tr>
<td>Distinct FAILURES</td>
<td>8</td>
</tr>
<tr>
<td>Total distinct Message Codes</td>
<td>4206</td>
</tr>
<tr>
<td>Total EVENT records</td>
<td>8502009</td>
</tr>
<tr>
<td>Total WARNING records</td>
<td>1103650</td>
</tr>
<tr>
<td>Total ERROR records</td>
<td>4400924</td>
</tr>
<tr>
<td>Total Failure records</td>
<td>1460</td>
</tr>
<tr>
<td>Average Error records logged per day</td>
<td>9979</td>
</tr>
<tr>
<td>Average Event records logged per day</td>
<td>19148</td>
</tr>
<tr>
<td>Average Warning records logged per day</td>
<td>2496</td>
</tr>
</tbody>
</table>

Table A.1: Analysed Log Data Statistics after temporal compression
Appendix A, Log File Analysis

<table>
<thead>
<tr>
<th>Type</th>
<th>Minimum</th>
<th>1st Quartile</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Quartile</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>1</td>
<td>13</td>
<td>170</td>
<td>20640</td>
<td>1412</td>
<td>1792740</td>
</tr>
<tr>
<td>Warning</td>
<td>1</td>
<td>5</td>
<td>32</td>
<td>2420</td>
<td>277</td>
<td>526683</td>
</tr>
<tr>
<td>Error</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>1328</td>
<td>61</td>
<td>526886</td>
</tr>
</tbody>
</table>

Table A.3: Summary of total occurrence of message codes per type after temporal compression
The quartile based information of the analyzed raw log data for the total occurrence of message codes per type is shown in the following Table A.4.

<table>
<thead>
<tr>
<th>Type</th>
<th>Minimum</th>
<th>1st Quartile</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Quartile</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>1</td>
<td>15</td>
<td>226</td>
<td>31950</td>
<td>1972</td>
<td>4947000</td>
</tr>
<tr>
<td>Warning</td>
<td>1</td>
<td>6</td>
<td>40.5</td>
<td>8268</td>
<td>348</td>
<td>1905000</td>
</tr>
<tr>
<td>Error</td>
<td>1</td>
<td>2</td>
<td>11</td>
<td>2463</td>
<td>74</td>
<td>852800</td>
</tr>
</tbody>
</table>

Table A.4: Summary of total occurrence of message codes per type in raw log data
Appendix B

Rule Mining

Confidence Distribution of Rules

The frequency of rules generated across confidence levels obtained for each failure for a $\triangle tw$ of 180 minutes, support, $s$ of 0.10 and a confidence, $c$ of 0.25 for varying lead time $\triangle lt$ values are shown in the following figures.

Figure B.1: RM - Frequency of rules generated across confidence levels obtained for each failure for a $\triangle lt$ of 8 minutes, $\triangle tw$ of 180 minutes, support, $s$ of 0.10 and a confidence, $c$ of 0.25.
Figure B.2: RM - Frequency of rules generated across confidence levels obtained for each failure for a $\Delta t_l$ of 10 minutes, $\Delta t_w$ of 180 minutes, support, $s$ of 0.10 and a confidence, $c$ of 0.25.

Figure B.3: RM - Frequency of rules generated across confidence levels obtained for each failure for a $\Delta t_l$ of 12 minutes, $\Delta t_w$ of 180 minutes, support, $s$ of 0.10 and a confidence, $c$ of 0.25.
### Figure B.4: RM - Frequency of rules generated across confidence levels obtained for each failure for a $\triangle lt$ of 30 minutes, $\triangle tw$ of 180 minutes, support, $s$ of 0.10 and a confidence, $c$ of 0.25.
Appendix C

Batch Rule Mining

Confidence Distribution of Rules

The frequency of rules generated across confidence levels obtained for each failure for $\Delta t_w$ of 180 minutes, support, $s$ of 0.10, confidence, $c_r$ of 0.50, distance separation $\Delta d$ of 24 hour, $c_b$ of 0.25 and varying lead time $\Delta l_t$ values are shown in the following figures.

Figure C.1: BRM - Frequency of rules generated across confidence levels obtained for each failure for a $\Delta l_t$ of 8 minutes, $\Delta t_w$ of 180 minutes, support, $s$ of 0.10, confidence, $c_r$ of 0.50, distance separation $\Delta d$ of 24 hour and $c_b$ of 0.25.
Figure C.2: BRM - Frequency of rules generated across confidence levels obtained for each failure for a $\Delta l_t$ of 10 minutes, $\Delta t_w$ of 180 minutes, support, $s$ of 0.10, confidence, $c_r$ of 0.50, distance separation $\Delta d$ of 24 hour and $c_b$ of 0.25.

Figure C.3: BRM - Frequency of rules generated across confidence levels obtained for each failure for a $\Delta l_t$ of 12 minutes, $\Delta t_w$ of 180 minutes, support, $s$ of 0.10, confidence, $c_r$ of 0.50, distance separation $\Delta d$ of 24 hour and $c_b$ of 0.25.
Figure C.4: BRM - Frequency of rules generated across confidence levels obtained for each failure for a $\triangle t_l$ of 30 minutes, $\triangle t_w$ of 180 minutes, support, $s$ of 0.10 , confidence, $c_r$ of 0.50, distance separation $\triangle d$ of 24 hour and $c_b$ of 0.25.
## RM and BRM Comparison

<table>
<thead>
<tr>
<th>Failures</th>
<th>$\triangle t=8$ Mins</th>
<th>$\triangle t=10$ Mins</th>
<th>$\triangle t=30$ Mins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>U</td>
<td>RM</td>
<td>BRM</td>
</tr>
<tr>
<td>LO-0052</td>
<td>42</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>LO-0058</td>
<td>35</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>LO-848F</td>
<td>8</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>LO-8422</td>
<td>82</td>
<td>13</td>
<td>28</td>
</tr>
<tr>
<td>LO-8434</td>
<td>41</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>LO-8459</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LO-8482</td>
<td>47</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>LO-8537</td>
<td>49</td>
<td>8</td>
<td>6</td>
</tr>
</tbody>
</table>

Table C.1: Comparison of RM and BRM. U-Union of rules generated from RM and BRM. RM-Relative complement of BRM with respect to RM. BRM-Relative complement of RM with respect to BRM.
Appendix D

Classification

Frequency Based Classification

The following Table D.1 shows the TP, FP, Precision and Recall for positive class of top three classifiers (order not preserved) with 12 Hour time window for failure prediction of $tw_{i+2}$ performed with original data. This table relates to Figure 5.15 which shows the average precision and recall of the top three classifiers for each failure for the same parameters.

<table>
<thead>
<tr>
<th>Failures</th>
<th>Algorithm</th>
<th>True Positive</th>
<th>False Positive</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>LO-8434</td>
<td>Naive Bayes</td>
<td>4</td>
<td>2</td>
<td>0.667</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>3</td>
<td>1</td>
<td>0.75</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>LO-8422</td>
<td>Naive Bayes</td>
<td>3</td>
<td>1</td>
<td>0.75</td>
<td>0.273</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>3</td>
<td>1</td>
<td>0.75</td>
<td>0.273</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.182</td>
</tr>
<tr>
<td>Failures</td>
<td>Algorithm</td>
<td>True Positive</td>
<td>False Positive</td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>---------------</td>
<td>----------------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>LO-8459</td>
<td>Naive Bayes</td>
<td>3</td>
<td>1</td>
<td>0.75</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>3</td>
<td>1</td>
<td>0.75</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>3</td>
<td>1</td>
<td>0.75</td>
<td>1</td>
</tr>
<tr>
<td>LO-8482</td>
<td>Naive Bayes</td>
<td>4</td>
<td>2</td>
<td>0.667</td>
<td>0.333</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>4</td>
<td>2</td>
<td>0.667</td>
<td>0.333</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.083</td>
</tr>
<tr>
<td>LO-8537</td>
<td>J48</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.091</td>
</tr>
<tr>
<td></td>
<td>NaiveBayes</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.091</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.182</td>
</tr>
<tr>
<td>LO-0058</td>
<td>J48</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td></td>
<td>NaiveBayes</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>LO-0052</td>
<td>NaiveBayes</td>
<td>3</td>
<td>3</td>
<td>0.5</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>2</td>
<td>3</td>
<td>0.4</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.2</td>
</tr>
</tbody>
</table>
### Table D.1

<table>
<thead>
<tr>
<th>Failures</th>
<th>Algorithm</th>
<th>True Positive</th>
<th>False Positive</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>LO-848F</td>
<td>J48</td>
<td>2</td>
<td>1</td>
<td>0.667</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>Naive Bayes</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>2</td>
<td>1</td>
<td>0.667</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table D.1: TP, FP, Precision and Recall for positive class of top three classifiers with 12 Hour time window for failure prediction of $tw_{i+2}$ performed with original data.

The following Table D.2 shows the TP, FP, Precision and Recall for positive class of top three classifiers (order not preserved) with 1 Hour time window for failure prediction of $tw_{i+2}$ performed with original data. This table relates to Figure 5.16 which shows the average precision and recall of the top three classifiers for each failure for the same parameters.

<table>
<thead>
<tr>
<th>Failures</th>
<th>Algorithm</th>
<th>True Positive</th>
<th>False Positive</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>LO-8434</td>
<td>J48</td>
<td>4</td>
<td>2</td>
<td>0.66</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>Naive Bayes</td>
<td>5</td>
<td>5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>3</td>
<td>4</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>LO-8422</td>
<td>J48</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0.059</td>
</tr>
<tr>
<td></td>
<td>Naive Bayes</td>
<td>2</td>
<td>3</td>
<td>0.4</td>
<td>0.118</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>2</td>
<td>2</td>
<td>0.5</td>
<td>0.118</td>
</tr>
<tr>
<td>Failures</td>
<td>Algorithm</td>
<td>True Positive</td>
<td>False Positive</td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>LO-8459</td>
<td>Naive Bayes</td>
<td>4</td>
<td>3</td>
<td>0.571</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LO-8482</td>
<td>Naive Bayes</td>
<td>1</td>
<td>3</td>
<td>0.25</td>
<td>0.071</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>1</td>
<td>2</td>
<td>0.333</td>
<td>0.071</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LO-8537</td>
<td>Naive Bayes</td>
<td>1</td>
<td>2</td>
<td>0.333</td>
<td>0.077</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LO-0058</td>
<td>Naive Bayes</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0.111</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.111</td>
</tr>
<tr>
<td>LO-0052</td>
<td>J48</td>
<td>3</td>
<td>2</td>
<td>0.6</td>
<td>0.273</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>3</td>
<td>7</td>
<td>0.3</td>
<td>0.273</td>
<td></td>
</tr>
<tr>
<td>RBF</td>
<td>3</td>
<td>6</td>
<td>0.333</td>
<td>0.273</td>
<td></td>
</tr>
</tbody>
</table>
Table D.2: TP, FP, Precision and Recall for positive class of top three classifiers with 1 Hour time window for failure prediction of \( tw_{i+2} \) performed with original data.

The following Table D.3 shows the TP, FP, Precision and Recall for positive class of top three classifiers (order not preserved) with 1 Hour time window for failure prediction of \( tw_{i+1} \) performed with original data. This table relates to Figure 5.17 which shows the average precision and recall of the top three classifiers for each failure for the same parameters.
<table>
<thead>
<tr>
<th>Failures</th>
<th>Algorithm</th>
<th>True Positive</th>
<th>False Positive</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>LO-8459</td>
<td>Naive Bayes</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>1</td>
<td>3</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>2</td>
<td>3</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>LO-8482</td>
<td>Naive Bayes</td>
<td>6</td>
<td>2</td>
<td>0.778</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>3</td>
<td>2</td>
<td>0.6</td>
<td>0.214</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0.214</td>
</tr>
<tr>
<td>LO-8537</td>
<td>Naive Bayes</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0.231</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.154</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.154</td>
</tr>
<tr>
<td>LO-0058</td>
<td>J48</td>
<td>5</td>
<td>2</td>
<td>0.714</td>
<td>0.556</td>
</tr>
<tr>
<td></td>
<td>Naive Bayes</td>
<td>6</td>
<td>3</td>
<td>0.667</td>
<td>0.667</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>3</td>
<td>2</td>
<td>0.6</td>
<td>0.333</td>
</tr>
<tr>
<td>LO-0052</td>
<td>J48</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.091</td>
</tr>
<tr>
<td></td>
<td>Naive Bayes</td>
<td>2</td>
<td>2</td>
<td>0.5</td>
<td>0.182</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>2</td>
<td>2</td>
<td>0.5</td>
<td>0.182</td>
</tr>
</tbody>
</table>
Table D.3: TP, FP, Precision and Recall for positive class of top three classifiers with 1 Hour time window for failure prediction of $tw_{i+1}$ performed with original data.

The following Table D.4 shows the TP, FP, Precision and Recall for positive class of top three classifiers with respective parameters for failure prediction of LO-0052 performed with original data. This table relates to Figure 5.18 A, which shows the average precision and recall of the top three classifiers for LO-0052.
Table D.4: TP, FP, Precision and Recall for positive class of top three classifiers with respective parameters for failure prediction of LO-0052 performed with original data.

The following Table D.5 shows the TP, FP, Precision and Recall for positive class of top three classifiers with respective parameters for failure prediction of LO-0052 performed with original data. This table relates to Figure 5.18 B, which shows the average precision and recall of the top three classifiers for LO-0058.
Table D.5: TP, FP, Precision and Recall for positive class of top three classifiers with respective parameters for failure prediction of LO-0058 performed with original data.

The following Table D.6 shows the TP, FP, Precision and Recall for positive class of top three classifiers with respective parameters for failure prediction of LO-0052 performed with original data. This table relates to Figure 5.18 C, which shows the average precision and recall of the top three classifiers for LO-848F.
### Failures

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>True Positive</th>
<th>False Positive</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0.545</td>
</tr>
<tr>
<td>RBF</td>
<td>5</td>
<td>1</td>
<td>0.833</td>
<td>0.455</td>
</tr>
<tr>
<td>VFI</td>
<td>6</td>
<td>1</td>
<td>0.857</td>
<td>0.545</td>
</tr>
</tbody>
</table>

Table D.6: TP, FP, Precision and Recall for positive class of top three classifiers with respective parameters for failure prediction of LO-848F performed with original data.

The following Table D.7 shows the TP, FP, Precision and Recall for positive class of top three classifiers (order not preserved) with 24 Hour time window for failure prediction of \( t_{w_{i+1}} \) for RBC. This table relates to Figure 5.21 which shows the average precision and recall of the top three classifiers for each failure for the same parameters.

<table>
<thead>
<tr>
<th>Failures</th>
<th>Algorithm</th>
<th>True Positive</th>
<th>False Positive</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>LO-8434</td>
<td>J48</td>
<td>4</td>
<td>2</td>
<td>0.66</td>
<td>0.44</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>4</td>
<td>2</td>
<td>0.66</td>
<td>0.44</td>
<td></td>
</tr>
<tr>
<td>VFI</td>
<td>4</td>
<td>2</td>
<td>0.66</td>
<td>0.44</td>
<td></td>
</tr>
<tr>
<td>LO-8422</td>
<td>RBF</td>
<td>4</td>
<td>4</td>
<td>0.5</td>
<td>0.4</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>4</td>
<td>1</td>
<td>0.8</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>VFI</td>
<td>4</td>
<td>4</td>
<td>0.5</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>Failures</td>
<td>Algorithm</td>
<td>True Positive</td>
<td>False Positive</td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>LO-8482</td>
<td>Naive Bayes</td>
<td>5</td>
<td>3</td>
<td>0.62</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>5</td>
<td>3</td>
<td>0.62</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>VFI</td>
<td>5</td>
<td>3</td>
<td>0.62</td>
<td>0.41</td>
</tr>
<tr>
<td>LO-8537</td>
<td>NaiveBayes</td>
<td>4</td>
<td>2</td>
<td>0.66</td>
<td>0.44</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>4</td>
<td>2</td>
<td>0.66</td>
<td>0.44</td>
</tr>
<tr>
<td></td>
<td>J48</td>
<td>4</td>
<td>1</td>
<td>0.8</td>
<td>0.44</td>
</tr>
<tr>
<td>LO-0058</td>
<td>J48</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.28</td>
</tr>
<tr>
<td></td>
<td>Naive Bayes</td>
<td>4</td>
<td>1</td>
<td>0.8</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>4</td>
<td>1</td>
<td>0.8</td>
<td>0.57</td>
</tr>
<tr>
<td>LO-0052</td>
<td>VFI</td>
<td>3</td>
<td>3</td>
<td>0.5</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>Naive Bayes</td>
<td>3</td>
<td>3</td>
<td>0.5</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>3</td>
<td>3</td>
<td>0.5</td>
<td>0.33</td>
</tr>
<tr>
<td>LO-848F</td>
<td>J48</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>Naive Bayes</td>
<td>4</td>
<td>2</td>
<td>0.66</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>4</td>
<td>1</td>
<td>0.8</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table D.7: TP, FP, Precision and Recall for positive class of top three classifiers for RBC with compression factor of 2 and prediction on $tu_{i+2}$
Appendix E

Rules State Space

Output File - Example Instance

An example instance of the XML file format generated for RSS for the failure LO-8422 is shown in Figure E.1.

Figure E.1: Example Instance - XML file generated for RSS for the failure LO-8422