

FAILURE PREDICTION

An Application in the Railway Industry

by

Pedro Mota Pereira

Master's Thesis in *Data Analytics*

Supervisors:

Prof. Doutor João Manuel Portela da Gama

Doutora Rita Paula Almeida Ribeiro

Faculdade de Economia

Universidade do Porto

2013/2014

Memoir

Pedro Mota Pereira has a Mechanical Engineering Degree from Porto's Faculty of Engineering. Later he extended his knowledge in the management field taking the Magellan Master's in Business Administration at Porto Business School. In 2012, enrolls the Master in Data Analysis, Simulation and Decision Support Systems at Faculty of Economics at Porto University, looking for specific expertise in Data Analytics, subject of his deepest interest.

Regarding his professional experience, he has been working in areas spreading from automotive industry to transportation. Presently, he is senior operations manager at Metro do Porto, where is responsible for contracts management. In this position he has been involved in subjects such as projects coordination, design review, service level definition, public tendering and bid evaluation.

As a result of the combination of academic interest in data mining and professional activate in the maintenance field, he has published *Failure Prediction – An Application in the Railway Industry*, Discovery Science 2014, Bled 2014, Slovenia, *Proceedings of the 17th International Conference on DS- Lecture Notes in Artificial Intelligence*, being awarded the Carl H Smith for Best Student Paper.

Acknowledgement

I would like to thank the ones that in some way contributed to this work.

To Rita Ribeiro and João Gama for their guidance.

To my family for their invaluable and unsurpassable support.

This work was supported by Sibila research project (NORTE-07-0124-FEDER-000059), financed by North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT), and by European Commission through the project MAESTRA (Grant number ICT-2013-612944).

Abstract

The purpose of this project is to develop a data mining system that issues an alarm whenever is predictable that an automatic door of a train is about to have a failure.

In this study case we focus our attention in the behaviour of the pneumatic doors from one specific train. Each door is activated by a linear pneumatic actuator, equipped with one pressure transducer on both the inlet and outlet chamber, providing a pressure reading every 1/10 second whenever the door is commanded to move. The available data, representing operations from September to December 2012, consists of almost 232 thousand readings, corresponding to 4590 opening and closing cycles. We must notice, that current opening systems are equipped with sensors that react (inverse the movement) when a passenger interferes. This fact, a system feature, triggers false alarms in fault detection systems that we need to avoid.

To accomplish this task we have come up with a two-stage classification process. First, each cycle is classified as Normal or Abnormal, afterwards we use a low-pass filter in the output to decide if there is evidence that a door breakdown is about to happen. For the cycle classification problem we have experimented three different methods: 1) unsupervised learning based on boxplot; 2) semi-supervised learning with OneClassClassification; 3) supervised learning with Support Vector Machine.

The combination of cycle classification and output post-processing has enabled the development of a system that addresses the problem at hand, anticipating door failures and, at the same time, avoiding disturbing false alarms, two characteristics usually hard to balance.

CONTENTS

1	Introduction.....	1
1.1	Context and Motivation.....	1
1.2	Goals	3
1.3	Document Organization	4
2	Related Work	5
2.1	Time-series Anomaly Detection	9
2.2	Propositional Anomaly Detection	12
2.2.1	Unsupervised Outlier Detection.....	13
2.2.2	Semi-supervised Anomaly Detection	15
2.2.3	Supervised Anomaly Detection	16
3	Case Study	17
3.1	Context	17
3.2	Methodology	19
3.3	Data Description.....	23
3.3.1	Initial Data	23
3.3.2	Data Transformation	30
3.3.3	Labelling	34
4	Experimental Testing.....	35
4.1	Experimental Setup	35
4.1.1	Cycle Classification	36
4.1.2	Sequence Classification	37
4.2	Experimental Evaluation	38
4.2.1	Results using Boxplot Outlier detection	39
4.2.2	Results using Novelty Detection.....	40
4.2.3	Results using Supervised Classification	44
4.3	Discussion	47
5	Conclusions and Future Work	48

6	Bibliography	51
	ANNEXS	55
	Annex I – Dataset Examples.....	56
	Annex II – Original Variables – Statistical Tests	58
	Annex III – Transformed Variables – Statistical Tests.....	63
	Annex IV – Maintenance Reports	65

LIST OF FIGURES

Figure 1 - Anomaly Detection Techniques in Engineering Applications.....	5
Figure 2 - DTW for two time series.....	10
Figure 3 - SAX Phase 1 and 2 representations	11
Figure 4 - Class 156 train from Northern Rail.....	17
Figure 5 - Door Pneumatic schematic.....	18
Figure 6 - Reference Model CRISP-DM Phases	21
Figure 7 - Air pressure evolution in opening and closing cycles.....	24
Figure 8 - Pressure evolution in maintenance cycle	25
Figure 9 - Pressure evolution in an inversed closing cycle.....	26
Figure 10 - Cycle Duration Boxplot and minimum value	29
Figure 11 - Open Pressure across Months – Statistical Test.....	30
Figure 12 - Close Pressure across Months - Statistical Test.....	30
Figure 13 - Open and Close Mean Pressure Difference - Independent Samples Test....	30
Figure 14 - Opening and closing door movement evolution	31
Figure 15 - Data Transformation Process	32
Figure 16 - Daily average of the 5 bins for closing door movements	32
Figure 17 - Open Cycle Attrib. B1 to B5 Distribution Test	33
Figure 18 - Close Cycle Attrib. B1 to B5 Distribution Test.....	33
Figure 19 - The impact of the low-pass filter using boxplot based outlier detection	40
Figure 20 - OCC Knime workflow	42
Figure 21 - The impact of the low-pass filter using OCC for novelty detection	43
Figure 22 - SVM Knime workflow.....	45
Figure 23 - The impact of the low-pass filter using SVM supervised learning.....	46
Figure 24 - Statistical Tests for Duration Distribution Cycles	58
Figure 25 - Statistical Test for Duration is the same across Month.....	59
Figure 26 - Duration across Month - Mean Difference Test – Opening Cycles.....	59
Figure 27 - Figure 35 - Statistical Tests for Duration Distribution Closing Cycles	60
Figure 28 - Statistical Test for Duration is the same across Month.....	61
Figure 29 - Duration across Month - Mean Difference Test – Closing Cycles	61
Figure 30 - Statistical Test for Duration is the same across Cycle type	62
Figure 31 - Duration across Cycle Type - Mean Difference Test	62

Figure 32 - Transformed Variables Distribution - Statistical Tests.....	63
Figure 33 - Transformed Variables Distribution - Statistical Tests.....	64
Figure 34 - Maintenance Report December 1 and 16th.....	65
Figure 35 - Maintenance Report December 6th.....	65

LIST OF TABLES

Table 1 – Part of the Original Dataset for movement 10003712 - Opening Cycle	19
Table 2 - Part of the Original Dataset for movement 10003712 – Closing Cycle.....	19
Table 3 - Door Cycle types	24
Table 4 - Cycle Duration - Descriptive Statistics	28
Table 5 - Cycle occurrence distribution.....	28
Table 6 - Pressure - Descriptive Statistics	29
Table 7 - Attributes B1 to B5 - Descriptive Statistics	33
Table 8 - Identification of Abnormal cycles by domain expert.....	34
Table 9 - Identification of Failures by domain expert	34
Table 10 - Results using boxplot-based outlier detection.....	39
Table 11 - Results using OCC for novelty detection	42
Table 12 - Results using SVM supervised learning.....	46
Table 13 - Original Dataset for movement 10003712 - Opening Cycle.....	56
Table 14 - Original Dataset for movement 10006282 - Closing Cycle	57
Table 15 - Transformed Variables – Opening - Descriptive Statistics	63
Table 16 - Transformed Variables – Closing - Descriptive Statistics	64

1 INTRODUCTION

Predicting the future is an activity that attracts huge interest from humanity.

As the Greek poet C. P. Cavafy said:

Ordinary mortals know what is happening now, the gods know what the future holds because They alone are totally enlightened. Wise men are aware of the future things just about to happen.

The ability to predict what is about to happen can make significant changes in how to run a business. It is hoped that the practical demonstration of the improvements achievable through the application of a data mining system to a specific day-to-day problem can be a further contribution to this area of knowledge, pointing out the advantages at hand to a wide range of corporations once they to embrace these kind of approach.

1.1 Context and Motivation

A Rail Vehicle is a highly complex equipment, consisting on a variety of integrated subsystems, assembled to provide public or freight transport.

Train Passenger Doors have a key role in such a transport system, allowing entering or exiting the vehicle at the right moment and ensuring for the remainder of the trip, the maximum tightness, thermal and acoustic isolation. In addition, modern train doors have safety features, preventing customers from leaving the train while in motion or not stopped at a suitable location for passengers exiting.

If in railways early day's doors were local and manually operated, the challenges posed by the need to reduce on board the human resources, the growth in safety requirements and the advantages associated to a faster operation led to the sophistication of this equipment. Indeed, nowadays doors are a highly complex system, comprising electronic control circuits and pneumatic or electric drive systems, which in many cases reach opening and closing times of less than 2 seconds, and security mechanisms such as anti-

pinch or force limiters. The complexity growth of these functionalities increased reliability and maintenance issues.

In fact, if in the past it was enough to lubricate hinges and adjust door alignments, today each door consists on many subsystems such as pneumatic valves, sensors, micro switches, call buttons and other, which greatly contributes to a huge growth in failure opportunity.

As usual, the growth in the number of components and the increasing complexity of their control poses additional problems in terms of reliability. In the case of train doors, its failure often causes relevant damages to the operation, not only at service level, but also on the costs of operating the system, such as:

- delays;
- trip cancellation;
- operational inefficiencies.

Given the significant impacts of door failures, much has been done to decrease its occurrence. Attention has been paid to areas spreading from the project phase, concerning design simplification or critical devices redundancy, through reinforced preventive maintenance, including, for example, increased equipment replacement rate, to the introduction of new Maintenance Management methodologies, where Conditional Maintenance from [Nowlan and Heap \(1978\)](#) stands out as the most usual trend.

As life expectancy of railway rolling stock often exceeds 30 years, capturing the benefits from changes in the design phase has a residual impact on the short term. Considering the vehicles already in operation, it is necessary to find other ways to improve maintenance costs. Conditional Maintenance Management, also nicknamed Predictive Maintenance, seems to address this need, but its application is not without problems or specific requirements, such as monitored parameters selection or adequate thresholds definition.

Taking into account all this background, the emergence of data mining techniques, seems to represent a line of action with great potential to solve or at least to minimize, some of the problems the rail industry is facing.

Indeed, the prospect that the intensive use of technology can make a more secure, coordinated and efficient transport systems led the European Union itself (EU) to issue a policy, the European Directive 2010/40/EU, on Intelligent Transport Systems (ITS).

Bearing in mind all stated above, there is no doubt that the ability to have a train that could warn us in advance, whenever a door a failure is about to happen, would be an advantage that clearly contributes to a customer service level improvement, as well as to a more efficient operation and maintenance. Therefore, data mining techniques in the field of novelty detection and, more specifically, failure prediction systems seem very promising opportunities to address some of the challenges that the railway industry must face to remain economically competitive.

1.2 Goals

The goal of this project is to develop a system that signals an alarm when a sequence of doors operations indicates a deterioration of the system. We must point out that we are not interested in signalling alarms when a single operation is abnormal. This is not an indication of a problem in the train opening system but, most probably, the interference of a passenger. Most of the predictive machine learning approaches for anomaly or failure prediction assume independent and identically distributed observations. They do not deal with sequential nor temporal information. In this study, we propose the application of a low-pass filter over the output of the predictive model to identify sequences of abnormal predictions that correspond to a deterioration of train door system.

This thesis contribution reaches two different fields. On one hand, the implementation of an anomaly detection system in a practical real world problem, on the other hand, at an academic level, the use of a low-pass filter to process the output of the predictors leading to a strong reduction in the false alarm rate.

1.3 Document Organization

A Master's Thesis should contain the formalization of the work done during the course. Nevertheless, numerous tasks were performed that did not find their space in this document and were fundamental for the final result. Experiments that did not go according to the plan might seem a dead end, but often we could not know a dead end unless we go there.

The document's organization follows the standard outline. In Chapter 1 is set the framework, including context and motivation for this work.

Literature review is included in Chapter 2, covering some anomaly detection techniques in times series, as well as in Attribute Value Matrix datasets.

In Chapter 3, we present details on the case study, including analysis to the original dataset and variables transformation.

Modelling and experimental results on the different experiments done compose Chapter 4, which also includes a final subchapter on results discussion.

Finally, Chapter 5 sums up the work, presenting major conclusions of our study and sets some ideas on future work developments.

2 RELATED WORK

Novelty detection has been the subject of intense investigation by the *academia*, as evidenced by the work of [Markou and Singh \(2003\)](#) which summarizes the main ways of tackling this problem and describes some of the algorithms that excelled in this area. In addition, [Chandola \(2009\)](#) focuses specifically on failure detection and provides a framework for the characterization of these problems, as well as an identification of the best methods for different applications, from intrusion detection to text mining problems. In the same document, the author also refers to the case of anomaly detection in engineering applications, presenting some specific suited techniques, as displayed on Figure 1.

Technique Used	Section	References
Parametric Statistical Modeling	Section 7.1	Guttormsson et al. [1999], Keogh et al. [1997, 2002, 2006]
Non-parametric Statistical Modeling	Section 7.2.2	Desforges et al. [1998]
Neural Networks	Section 4.1	Bishop [1994], Campbell and Bennett [2001], Diaz and Hollmen [2002], Harris [1993], Jakubek and Strasser [2002], King et al. [2002], Li et al. [2002], Petsche et al. [1996], Streifel et al. [1996], Whitehead and Hoyt [1993]
Spectral	Section 9	Parra et al. [1996], Fujimaki et al. [2005]
Rule Based Systems	Section 4.4	Yairi et al. [2001]

Figure 1 - Anomaly Detection Techniques in Engineering Applications
Source: Chandola (2009)

The increased degree of automation and the growing demand for higher performance, efficiency, reliability and safety in industrial systems has resulted in the recently development of on-line fault detection and isolation techniques. [Angeli and Chatzinikolaou \(2004\)](#) survey on-line fault detection techniques for technical systems. In their work, three different methods are considered: 1) numerical methods; 2) artificial intelligence methods; 3) combinations of the two previous methodologies. Fault detection using numerical techniques based on mathematical system models is a well-established subject and a lot of survey papers and books have been written, such as [Isermann \(1984\)](#), [Basseville \(1988\)](#), [Basseville and Nikiforov \(1993\)](#), [Patton, Clark et al. \(2000\)](#). Under this methodology, fault detection is basically signal processing

techniques employing state estimation, parameter estimation, adaptive filtering, variable threshold logic, statistical decision theory and analytical redundancy methods.

On the other hand, a neural networks is trained to learn, from the presentation of the examples, to form an internal representation of the problem. For diagnosis it is needed to relate the sensor measurements to the causes of faults, and distinguish between normal and abnormal states. Some of the work done on this matter has been produced by [Hoskins and Himmelblau \(1988\)](#), [Fujiwara \(1995\)](#) and [Papadimitropoulos, Rovithakis et al. \(2007\)](#). As stated by Angelo, neural networks are able to learn diagnostic knowledge from process operation data. However, the learned knowledge is in the form of weights which are difficult to comprehend In fact, they operate as “black boxes” using unknown rules and are unable to explain the results. Neural networks find application in fault detection due to their main ability of pattern recognition, because they are able to internally map the functional relations that represent the process

Automatic methods for fault detection have been studied for a long time. In [Katipamula and Brambley \(2005\)](#) techniques, such as expert systems, fuzzy logic and data mining, are used to address a diversity of application areas including aerospace, process controls, automotive, manufacturing, nuclear plants, etc. Failure detection can be found in a range of practical applications, being the most captivating ones those relating to aeronautics and aerospace.

One of the most demonstrative examples of the advantage taken by the application of Failure Predictions is on the Health and Usage Monitoring Systems of the International Helicopter Safety Team. [Hardman, Hess et al. \(2000\)](#) showed that the use of data from the vibrations signature produced by the various elements of the traction chain and the aircraft structure could allow us to anticipate situations that lead to breakdowns, some of which could result in fatal accidents.

In the aerospace area, NASA has also implemented a number of failure detection techniques to increase mission's control, particularly on the status of their aircrafts ([Iverson \(2008\)](#)) and thereby increase the success rate of missions.

[Martinez-Heras, Donati et al. \(2012\)](#) clearly demonstrate some of the inherent advantages that can be achieved with anomaly detection, proving that its application anticipated the knowledge of the development of a fault, long before the predefined alarm levels were reached. In their study, authors compare automatic telemetry monitoring, performed by Out-of-Limits (OOL) alarms to a monitoring paradigm based on Novelty Detection. According to their conclusions, the OOL approach, which consists on defining an upper and lower threshold so that when a measurement goes above the upper limit or below the lower one an alarm is triggered, is outperformed by the proposed novelty detection methodology. In fact, often, novel behaviours are signatures of anomalies and their detection allows engineers in some cases to react before the anomaly develops.

There are a number of papers that address the problem of fault detection using neural networks. In chemical engineering, [Watanabe, Matsuura et al. \(1989\)](#), [Venkatasubramanian and Chan \(1989\)](#) and [Ungar, Powell et al. \(1990\)](#) were among the first researchers to demonstrate the usefulness of neural networks for the problem of fault diagnosis.

Later, [Venkatasubramanian, Vaidyanathan et al. \(1990\)](#) presented a more detailed and thorough analysis of the learning, recall and generalization characteristics of neural networks for detecting and diagnosing process failures in steady-state processes. This work was later extended by [Vaidyanathan and Venkatasubramanian \(1992\)](#) to utilize dynamic process data. In 1994 ([Watanabe, Hirota et al.](#)) proposed a hierarchical neural network architecture (HANN) for the detection of multiple faults. According to the author, one of its advantages is that multiple faults could be detected in new data even if the network was trained with data representing single faults.

The impacts of component failures on railway systems can significantly affect technical and operational reliability. Many advanced railway systems and components are equipped with monitoring and diagnostic tools to improve reliability and reduce maintenance expenditures.

In [Yilboga, Eker et al. \(2010\)](#), authors present a neural network based failure prediction algorithm for railway turnouts, using sensory information collected in real-time from

sensors embedded on track electro-mechanical systems. In their work it is used a time delay neural network (TDNN), which is a neural network that incorporates the time information in the structure. The structure of TDNN is set so that the input of the neural network consists of parameters in different time units, important characteristic in time related problems, such as prediction. Thus, the effect of past values of parameters can be incorporated in to the problem and, therefore, the input can also include the past outputs of TDNN. TDNN is especially important in prediction and classification of time related patterns such as speech recognition.

Automatic anomaly detection to forecast potential failures on the railway maintenance task appears in [Rabatel, Bringay et al. \(2009\)](#) and [Rabatel, Bringay et al. \(2011\)](#). Authors start by characterizing normal behaviour taking into account the contextual criteria associated to railway data (itinerary, weather conditions, etc.). After that, they measure the compliance of new data, according to extracted knowledge, and provide information about the seriousness and possible causes of a detected anomaly.

Approaches to predict component failure and remaining useful life are usually based on continuously measured data. The use of event data is limited, especially for predicting failures in railway systems. In [Fink, Zio et al. \(2013\)](#) Extreme Learning Machines (ELM) to predict the occurrence of railway operation disruptions based on discrete-event data are applied. According to the authors, ELM have a good generalization ability, are computationally very efficient and do not require tuning of network parameters. In their study, they use real data concerning failures that cause undemanded service brake application of railway vehicles to demonstrate ELM performance. In fact other machine learning techniques, such as multilayer perceptrons and feed forward neural networks with learning based on genetic algorithms, were not able to extract patterns in the diagnostic event data, whereas the proposed approach was capable of predicting 98% of the operation disruption events correctly.

In the framework of the problem at hand, data is originally of a time series type. In fact, initial data has a sequential structure, as each observation represents the system status at a given time, with data collected at uniform time intervals.

In this context there are at least two different approaches to address the problem:

- as a time series data stream;
- as an attribute-value array after dataset transformation.

The next two subsections contain a briefly overview on techniques for anomaly detection considering both data structures.

2.1 Time-series Anomaly Detection

As mentioned above, the data in its original form is a time-series, representing opening and closing doors episodes.

Time series analyses have been the subject of extensive research. Some of the most important works in this field related to anomaly detection have been produced by Keogh, pointing out the publications from [Keogh \(2002\)](#), Lin et al., [Keogh and Ratanamahatana \(2005\)](#) and [Lonardi, Lin et al. \(2006\)](#), focusing on building efficient time series comparing algorithms, such as Dynamic Time Warping or Symbolic Approximation.

Dynamic Time Warping

The Dynamic Time Warping (DTW) algorithm is a technique to analyse the similarity of time series ([cf. Gama J \(2010\)](#)), which determines an index of similarity between-series. This technique can be seen as an alternative similarity measure to the frequently used Euclidean distance. It is shown that the DTW minimizes the Euclidean distance, looking for a best fit between the two data series, which is especially relevant in cases where the temporal sequences develop at different rates and durations.

The formulation of the DTW is based on seeking the least cost path connecting the points of both series, ensuring synchronization of the start and end of both series. Usually the cost function used is the Euclidean distance. In Figure 2, we can see an a DTW application between two time series, as shown in [Gama J \(2010\)](#).

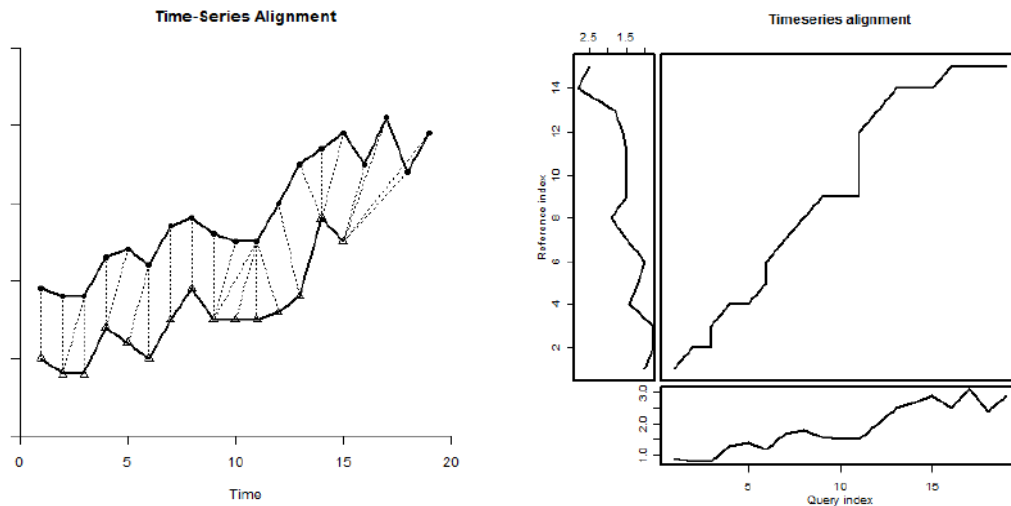


Figure 2 - DTW for two time series
Source: [Gama J \(2010\)](#)

In [Chandola \(2009\)](#) authors present a comprehensive evaluation of a large number of semi-supervised anomaly detection techniques for time series. In their experimental testing using datasets from NASA Dashlink Data Archive and UCR Time Series Archive, DTW was used to detect, among others, a disk defect, a faulty series from an electric motor and the anomalous operations of a NASA valve, using the TEK solenoid current measurements recorded during the normal operation of a Marrotta series valves located on a space shuttle.

Symbolic Approximation – SAX

In 2003 ([Lin, Keogh et al.](#)) presented the SAX algorithm, providing a new symbolic representation. It was proposed with a double objective: reduce the size of the data and

provide the lower limit of the distance measurements of the original data, allowing some algorithms to efficiently perform symbolic representation data mining.

SAX transforms a time series of length n in a string of arbitrary length. The process is divided into the following three main steps:

1. Piecewise Aggregate Approximation (PAA);
2. Symbolic Discretization;
3. Distance Measure.

The first step involves dividing the time series of dimension n into w segments of equal size, where the value of each one corresponds to the original series average.

Then, in the second step, the process of symbolic discretization ensures that each symbol has the same probability of occurrence throughout the series and transforms the resulting vector of PAA into a symbolic string. Figure 3 is a representation of step one and two of the SAX process.

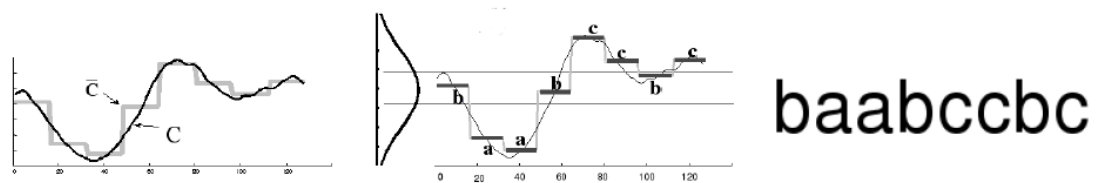


Figure 3 - SAX Phase 1 and 2 representations
Source: Gama J (2010)

Finally, at the third step, we can calculate the distance between the two vectors generated from the time series to compare, using the formula proposed by [Lin, Keogh et al. \(2003\)](#).

The use of strings for the representation of time series allows the application of data mining techniques that would otherwise be difficult or impossible ([cf. Gama J \(2010\)](#)). This type of representation is particularly useful in data mining tasks such as clustering, classification, and anomaly detection.

Concerning novelty detection, SAX has been used to search for motifs ([cf. Lin, Keogh et al. \(2003\)](#)) and discords ([cf. Keogh, Lin et al. \(2005\)](#)). On this regard, the HOT-SAX ([cf. Keogh, Lin et al. \(2005\)](#)) algorithm and the work of [Lonardi, Lin et al. \(2006\)](#) are

developments of the SAX methodology particularly oriented to the detection of discords, which is the longest subsequence of a longer time series that presents the maximum difference from all other series subsequences. The value of discords is relevant to the detection of faults particularly because they depict the most unusual subsequence of time, usually linked to failures.

2.2 Propositional Anomaly Detection

Episode time series, our initial dataset format, can be transformed into an attribute value matrix. This change in the data structure allows us to apply more conventional learning algorithms, such as decision trees. The transformation process usually involves considering each episode as an observation described by n , constant, attributes.

One of the biggest difficulties in anomaly detection is the ability of a machine learning system to identify new or unknown concepts, that were not present during the learning phase [Petsche \(1996\)](#), [Saxena \(2007\)](#). This feature is essential in a good learner, because in practical applications, especially when data streams are involved, the test examples contain information on concepts that were not known during the train of the decision model. The ability to identify what are the new concepts is vital if a classifier can learn continuously, which will require that: 1) the classifier represents the current state, i.e. normal behaviour and 2) systematically check the compatibility between the current model and recent data.

In engineering applications one can consider that an anomaly is an outlier. In fact, bearing in mind that one of the most globally accepted outlier definition was given by [Hawkins \(1980\)](#) and states that an outlier is a data object that deviates significantly from the rest of the objects, as if it were generated by a different mechanism, the assumption that an anomaly must be an outlier seems reasonable. Thus, one way to tackle anomaly detection problems can be using outlier detection techniques.

According to [Han and Kamber \(2011\)](#), an outlier can be further divided into three different types: global, contextual and collective. In this thesis our focus is in global

outliers, also called point anomalies, data objects that are unlikely to follow the same distribution as the other objects in the data set.

Similarly to other learning tasks, depending on the existence of labelled instances, anomaly detection techniques can be divided in three main groups:

1) unsupervised; 2) semi-supervised; 3) supervised.

In the following subsections, we briefly describe these three different approaches and their application to our target problem.

The methods we review here are the most common approaches using data mining: outlier detection and novelty detection [Zhang, Yan et al. \(2006\)](#).

As previously stated, we tested three different approaches for anomaly detection, all of them considering an attribute value dataset. Our choice was based on software packages availability and simplicity.

In fact, other options could have been considered, with methods such as DTW or discord search coming on top of the list of alternatives. But truth is that on one hand their implementation would be much harder and on the other hand, the performance attained with the simpler model makes it the correct choice, avoiding the need to search for more complex solutions.

2.2.1 Unsupervised Outlier Detection

In the machine learning field, several techniques for outlier detection appeared in the area of unsupervised learning. In effect, due to the lack of labelled and known instances of outliers, this constitutes a wide area of research concerning outlier detection. These techniques do not have any premise on previous information, they only assume that for some similarity measure, outliers will appear isolated or in very small groups.

According to [Chandola \(2009\)](#), unsupervised outlier detection methods can be grouped into statistical methods, clustering methods, distance-based and density-based methods.

The choice of the appropriate method relies on several factors, such as the number of dimensions of the data, data type, sample size, algorithms efficiency, and, ultimately on the user understanding of the problem.

Whenever the goal is to identify univariate outliers, such as in the context of our problem, the statistical methods are among the simplest methods.

Assuming a Gaussian distribution and learning the parameters from the data, parametric methods identify the points with low probability as outliers.

One of the methods used to spot such outliers is the boxplot method, introduced by [Tukey \(1977\)](#). Based on the first quartile ($Q1$), the third quartile ($Q3$) and the interquartile range ($IQR=Q3-Q1$) of data, it determines that the interval $[Q1 - 1.5*IQR, Q3 + 1.5*IQR]$ contains 99.3% of data. Therefore, points outside that interval are considered as mild outliers, and points outside the interval $[Q1 - 3*IQR, Q3 + 3*IQR]$ are considered extreme outliers.

Outlier detection in Multivariate distribution is achieved under the concept of transforming the problem into a univariate one. A way to do such transformation can be using the Mahalanobis distance, obtaining a new distribution from each example's Mahalanobis distance. Then if the Mahalanobis distance of an example is an outlier, the example itself can be regarded as an outlier as well.

One of the most popular unsupervised outlier detection, from a density-based approach, is the Local Outlier Factor (LOF) ([Breunig, Kriegel et al. \(2000\)](#)). According to [Aggarwal \(2013\)](#), the classification of LOF as a density-based approach is a relaxation of the definition of density. In fact, even though the same author includes LOF in the density base category, he considers it a relative distance-based approach with smoothing. LOF is a quantification of the outlierness of the data points, which is able to adjust for the variations in the different densities. The local outlier factor is based on a concept of a local density, where locality is given by k nearest neighbors, whose distance is used to estimate the density. By comparing the local density of an object to the local densities of its neighbors, one can identify regions of similar density. Therefore, outliers are points that have a substantially lower density than their

neighbors. The “reachability distance” is a measure to produce more stable results within clusters and its definition is set on [Breunig, Kriegel et al. \(2000\)](#). Finally, the Local Outlier Factor of an observation is then simply equal to the mean ratio of the “reachability distance” of that observation to the corresponding points in its k -neighborhood.

2.2.2 Semi-supervised Anomaly Detection

In many practical anomaly detection applications it is only possible to have training sets consisting on elements from a single class, the normal one, being unavailable examples from the counter-class. In fact, in day-to-day problems in the maintenance field, frequently there are a lot of examples belonging to the Normal class and very few from the Outlier class, maybe not even representing all the failure modes. As stated by [Japkowicz, Myers et al. \(1995\)](#), in engineering anomaly detection problems often only examples from a single class, the Normal class, are available, whereas examples from the counter class might be very rare or expensive to obtain.

Anomaly detection in such a scenario, in which learning is made by only using samples from normal class, are usually given the name of one-class classification or learning from positive-only examples ([Tax \(2001\)](#)).

There are various ways to address OCC, such as one-class SVMs ([Han and Kamber \(2011\)](#)), or auto-associative neural networks, also known as autoencoders ([Japkowicz, Myers et al. \(1995\)](#)).

In this study we have chosen to use the OCC algorithm available in Weka by [Hempstalk, Frank et al. \(2008\)](#), which combines density and class probability estimation. In this algorithm only the Normal class examples are used for training, as the learning phase is done without using any information from other classes. Firstly, a density approach is applied to training data so to generate artificial data used to form an artificial outlier class. Then a classifier is built with examples from both Normal and Outlier classes.

2.2.3 Supervised Anomaly Detection

Supervised outlier detection techniques assume the existence of historical information on all the normal and outlier instances from where predictive models for outliers can be built. Most of the work regarding this area focus on classification tasks and, in particular, on binary classification as it considers only two classes: Normal and Outlier.

By the implicit definition of outlier, these classification tasks have an imbalanced class distribution, a well-known problem and subject of research ([Aggarwal \(2013\)](#)).

There are several different techniques to address unbalanced dataset [He and Garcia \(2009\)](#). Due to the inherent complex characteristics of imbalanced data sets, learning from such data requires new understandings, principles, algorithms, and tools to transform vast amounts of raw data efficiently into information and knowledge representation. Some of the most well-known techniques are 1) random oversampling; 2) informed oversampling. More sophisticated methodologies involve generating artificial data, cluster-based sampling, or the integration of sampling and boosting.

In this thesis, we cover the spectrum of supervised fault detection techniques by using a Support Vector Machine (SVM) ([Han and Kamber \(2011\)](#)).

In the context of our problem, the SVM will search for an optimal hyper-plane that can be used as decision boundary separating the examples from the Normal and Outlier classes.

3 CASE STUDY

The project's goal is to create a data mining system that issues an alert when a specific train door is about to have a failure, thus allowing a proper schedule for maintenance.

To address this task we have to choose a project management methodology. In Section 3.1 we briefly describe the problem. Then, in Section 3.2, we go over the major steps involved in this kind of project and how we covered this subject. Finally, in Section 3.3 we present an exploratory study on the initial dataset.

3.1 Context

The project goal is to create a data mining system that issues an alert when a specific train door is about to have a failure, thus allowing a proper schedule for maintenance.

The data at our disposal are the pressure readings from the inlet and outlet chamber of the pneumatic door actuator, at a 100 milliseconds interval, whenever the train door is operated.

For this case study, we used a database composed of data collected from September to December 2012 at the Northern Rail operation, which is a sample of about 4500 closing and opening cycles of a specific door, designated as Door 1.



Figure 4 - Class 156 train from Northern Rail
Source: northernrail.org

One train from NorthernRail class 156 fleet, depicted in Figure 4, is equipped with a data logging system, including two pressure readers at the door pneumatic actuator. Train door operation in the class156 fleet is pneumatic, having one central air pressure production unit that supplies all doors.

A simplified pneumatic schematic is presented in Figure 5. As in other pneumatic systems, connecting one chamber to the high pressure line and the other to the exhaust allows the piston to move from high pressure chamber to low pressure and, therefore, opening or closing the door. Travel start and ending are controlled by micro switches, allowing for a proper confirmation of door closed or opened.

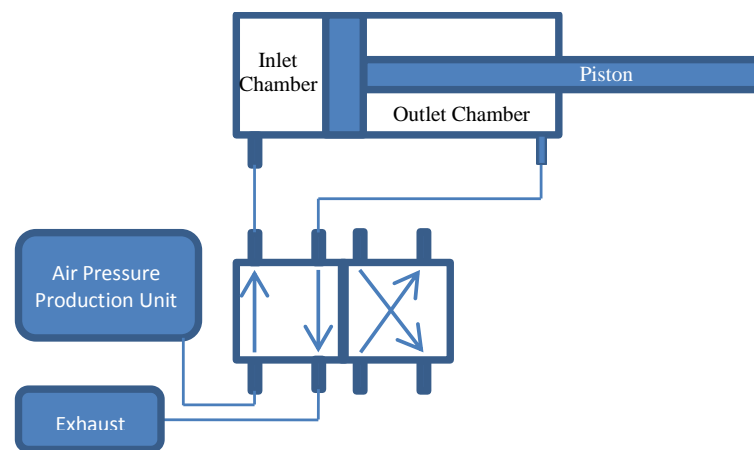


Figure 5 - Door Pneumatic schematic

Data collected from the pressure reader is directed to an on-board central computer that records it and later send it to a central server. In detail, we have seven variables, as described below.

- `start_datetime` - Date and time at the beginning of the movement [dd-mm-yyyy hh:mm:ss]
- `start_uptime` - timestamp at the beginning of the movement
- `date_time` - Date and time at the reading time [dd-mm-yyyy hh:mm:ss]
- `uptime` - timestamp at the reading time
- `variable_id` - pressure reader name
- `value` - pressure read at an particular uptime [Bar]
- `utimestamp` - timestamp in unix format

This logging system starts recording as soon as door is actuated, afterwards, at every 100 milliseconds interval, air pressure in both the inlet and outlet chamber are read, until the end travel micro switch is reached.

It should be emphasized that information on each cycle's class is not available, meaning that in the original dataset there is no variable which classifies whether the cycle should be considered normal, or abnormal.

In order to show the dataset in its original format, Table 1 and Table 2 represent two observations from one opening cycle and one closing cycle.

Table 1 – Part of the Original Dataset for movement 10003712 - Opening Cycle

start_datetime	start_uptime	date	hour	date_time	uptime	Variable_id	value	utimestamp
07-09-2012 15:54:10	10003712	07-09-2012	15:54:12	07-09-2012 15:54:12	10003716	801	1,932	1.347.029.650.004
07-09-2012 15:54:10	10003712	07-09-2012	15:54:12	07-09-2012 15:54:12	10003726	801	1,782	1.347.029.650.014

Table 2 - Part of the Original Dataset for movement 10003712 – Closing Cycle

start_datetime	start_uptime	date	hour	date_time	uptime	Variable_id	value	utimestamp
07-09-2012 15:54:38	10006282	07-09-2012	15:54:38	07-09-2012 15:54:38	10006286	801	2,736	1.347.029.678.004
07-09-2012 15:54:38	10006282	07-09-2012	15:54:38	07-09-2012 15:54:38	10006296	801	2,736	1.347.029.678.014

The complete dataset for both movements can be found in Annex I – Dataset Examples, where Table 13 and Table 14 represent all the data logs from one closing and one opening cycle, randomly picked.

Cycle with initial time stamp **10003712** refers to an opening and **10006282** is a closing movement. It should also be noticed that type of movement is not directly available from the dataset and it must be determined from air pressure evolution in each chamber.

3.2 Methodology

A Knowledge Discovery Project should consist of a certain set of tasks that lead to the desired result. There are several methodologies that can be adopted, but in general such a project should always include three groups of tasks:

- pre-processing;
- process data mining;
- post-processing.

In the pre-processing group are included tasks related to the preparation of the data for the data mining. The data mining process is the use of proper models, while the post-processing phase is dedicated to results validation and interpretation

There are methodologies that are well documented and ensure that knowledge discovery process can be well done and implemented at the end. They all detail the different steps, the required inputs and expected outputs for each one.

In this work, we follow the CRISP-DM methodology for being one of the most used ones and being particularly well suited to the problem that concerns us. Next, we briefly describe each of its stages.

CRISP-DM

This methodology is not associated with any data mining tool and was thought to be widely used, besides being free and open.

This methodology consists of six steps: Business Understanding (comprehension of business), Data Understanding (data analysis), Data Preparation (data preparation), Modelling, Evaluation (assessment) and Deployment (implementation).

Figure 6, taken from The CRISP-DM Process Model [Chapman, Clinton et al. \(1999\)](#), illustrates how to combine the various stages.

CRISP-DM foresees that the various steps can be performed iteratively. It is not expected to go from the first to the last stage and find the desired solution. Rather, the CRISP-DM provides that if you go from one stage to another and, should the need arise, a return to earlier to solve problems or detail necessary details to the following steps. Further, it is even possible to get to the stage of review and verify that the model created not effectively meet the project objectives with the need to return to the initial phase.

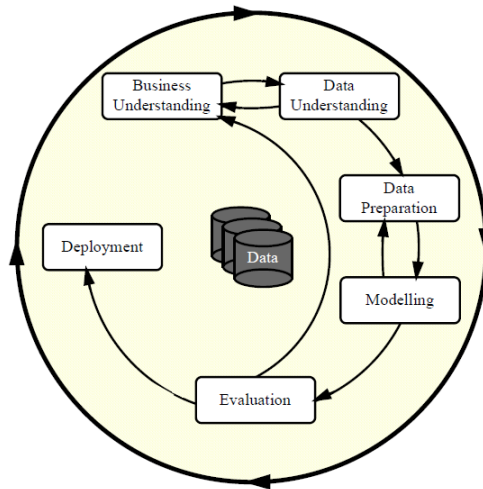


Figure 6 - Reference Model CRISP-DM Phases
Source: Chapman, Clinton et al. (1999)

Reading out our case study, we briefly describe what we have done in each one of the CRISP-DM six the steps.

Business Understanding

This first step is the understanding of the business and the project requirements, as well as defining the objectives of the data mining process.

We did a study of the importance of data mining projects in the railway maintenance management area in order to understand the extent to which the case study is relevant in real life and what are the fundamental objectives of a project of this nature (cf. Chapter 1).

Data Understanding

At this stage the data is collected and there is an initial analysis.

In this work, we did not really had to collect data, we relied on the cooperation of Nomadtech and NorthernRail, who kindly provided us with the data from a train data logging system.

The dataset contained about 500 thousands records, referring to the period from September to December 2012.

The analysis that we have made can be found in the Section 3.3.1.

Data Preparation

This is the stage where data is prepared to be used in the models from following stage (cf. Section Data Transformation).

This was a decisive phase of work since the way we prepare the data contributes significantly to the effectiveness of the tested models.

Finally, as described in Section 3.3.4, it was necessary to assign a label for each cycle that represented its normality, as well as validate its inclusion in the failure incubation period.

Modelling

At this stage the data mining models are configured and trained in order to obtain the best possible results, meeting the intended goals.

In our work we use a two stage model, where after cycle classification we apply a low-pass filter on the output of the classification. For cycle classification we have tested three different ways to do this.

Description of the work done on cycle classification and sequence classification can be found in chapter 4, Sections 4.1.

Evaluation

At this stage we evaluate the models created in the previous step and check if the proposed objectives have been met or whether to review the process and return to some of the earlier phases.

In this document, we review each of the classifiers used, make performance comparisons and issue a conclusion on which is the most appropriate model for our specific problem. Throughout this project we performed several iterations, especially in the stages of data preparation and modelling. See Chapter 4, Section 4.2 on experimental results.

Deployment

This final phase turns out to be the goal of the project, i.e., it is where the results are presented and the knowledge is made available to those who requested the knowledge extraction from data service.

The practical implementation of this project is under evaluation from Nomadtech and it is possible that it might be included in the company's maintenance software, after more comprehensive validation tests are done.

3.3 Data Description

3.3.1 Initial Data

Initial Dataset characterization

The initial dataset contains 500 thousands pressure records from September to December 2012. During this period the train was run in a usual way, without records of any special events and, therefore, one can assume that the available data is a good representation of standard operations.

As stated before, our original database contains two temporal data series, representing opening and closing door episodes. For each episode we have air pressure evolution both in the inlet and outlet chamber, as represented in Figure 7, as an example.

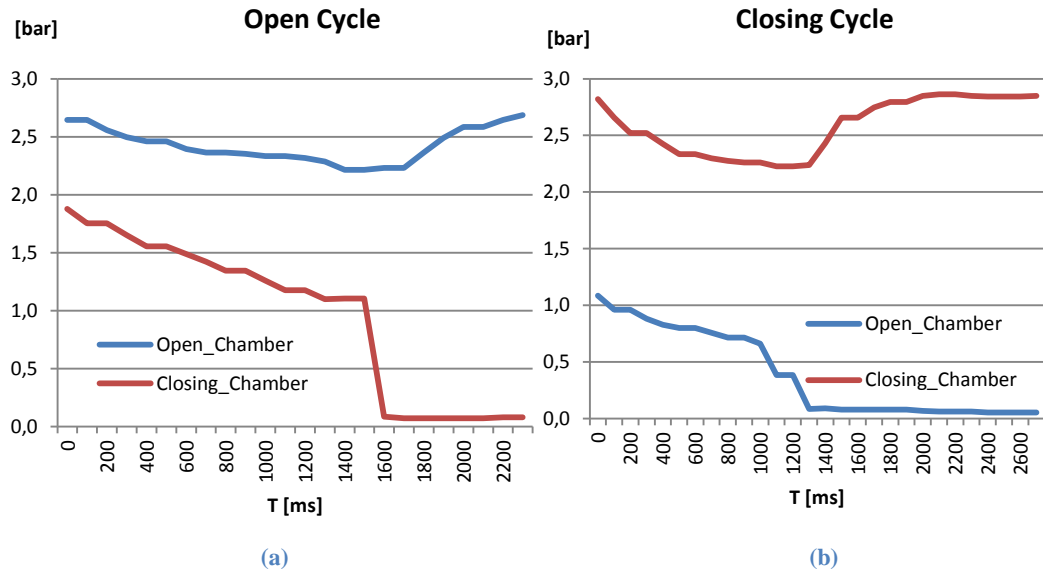


Figure 7 - Air pressure evolution in opening and closing cycles

For an easier and better understanding of the different door cycles, Table 3 summarizes the major patterns. For each door movement we then briefly describe its process.

Table 3 - Door Cycle types

Cycle type	Opening	Standard
		Inverted
		Maintenance
	Closing	Standard
		Inverted
		Maintenance

Standard Open Cycle

Begins with an electric command that can be given by the driver, a client or as an internal input. From that order there is a switch in the pneumatic valve, connecting the air pressure circuit to the opening chamber and the closing chamber to the exhaust. This combination causes the rod to move until it reaches the travel micro-switch, when the door is blocked at the opening position. (b)

Figure 7 a) is a diagram illustrating the development of pressure in both chambers for this movement.

Standard Close Cycle

Begins with an electric command that can be given by the driver, or as an internal input. After that order there is a switch in the pneumatic valve, connecting the air pressure circuit to the closing chamber and the opening chamber to the exhaust. This combination causes the rod to move until it reaches the travel micro-switch, when the door is blocked at the closing position. (b)

Figure 7 b) is a diagram illustrating the development of pressure in both chambers for this movement.

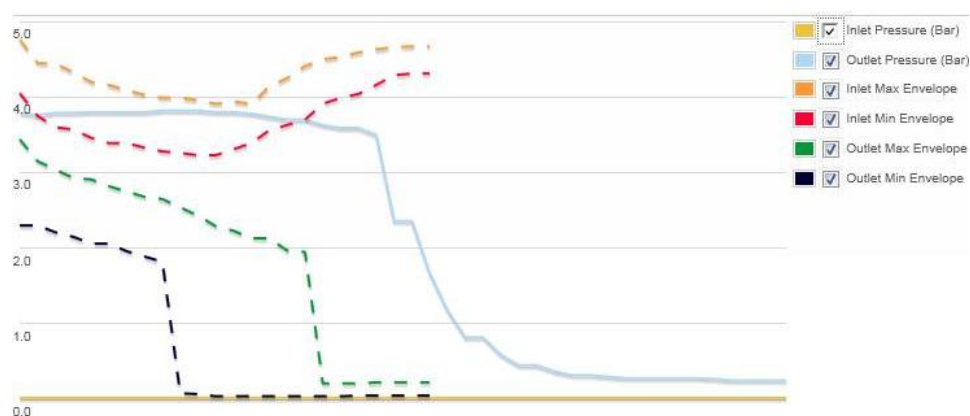


Figure 8 - Pressure evolution in maintenance cycle

Inverted opening cycle

During a standard opening cycle, movement might be inverted, beginning a closing cycle, not allowing the initial cycle to reach the end. This behaviour can be the result of a Drivers order.

Inverted Closing Cycle

During a standard closing cycle, movement might be inverted, beginning an opening cycle, not allowing the closing cycle to reach the end. This behaviour can be the result of a Drivers order, or an automatic command from the anti-pinch system.

Figure 9 is a diagram illustrating the development of pressure in both chambers for this movement.

Opening and Closing in Maintenance

Due to several reasons, sometimes there is the need to operate door on manual mode. In these movements pressure evolution is different from the ones presented before.

Figure 8 is a diagram illustrating the development of pressure in both chambers for this movement.

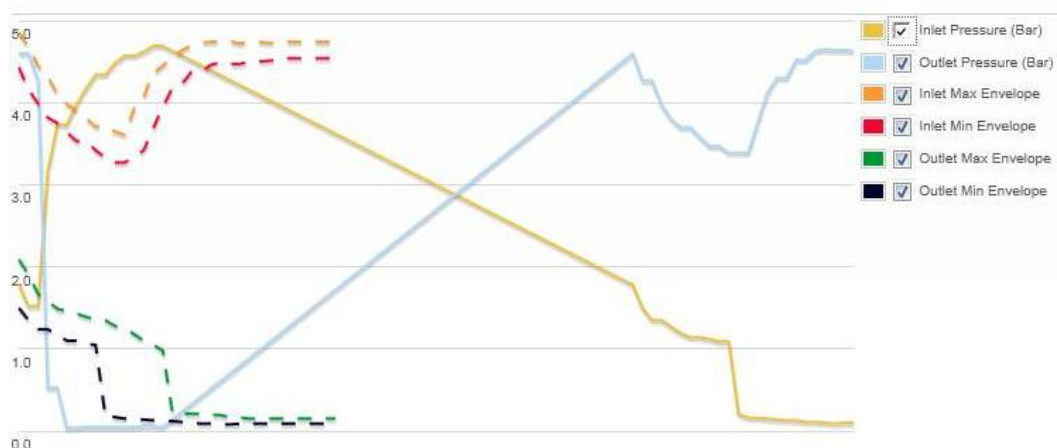


Figure 9 - Pressure evolution in an inversed closing cycle

Initial Dataset Summary

A data mining project usually includes a data validation process. Data validation goes from spotting incorrect or incongruent data, to solve the non-available data difficulty.

There are different ways to tackle the non-available problem, such as replace its value with the mean, or using the previous record, and it is up to the data mining scientist to choose the best solutions for the specific problem at hand.

In the initial dataset each pressure record was associated to a cycle reference, “start_uptime”, which contained information on the movement it belonged to. This data is vital to extract all the pressure readings from one cycle. Unfortunately, the available dataset missed that field in 306998 lines, more than 50% of the total. However, considering that there were still available more than 231500 lines with solid information, correctly representing 4590 door cycles, we decided to discard incomplete cycles and focus our study on the remaining part of the dataset. Table 5 summarizes the cycle occurrence distribution across months and cycle type.

From the initial dataset it is already possible to get a first impression on some cycle characteristics such as the cycle’s duration. In fact, just by looking at the cycle duration mean and standard deviation at Table 4, one can speculate some conclusions. Statistical tests, as shown in Annex II – Original Variables – Statistical Tests Figure 25, Figure 26, Figure 28 and Figure 29, support that cycle duration for opening and closing movement are different and even within the same movement there is significant change from one month to the next.

The conclusion on the increasing cycle length from September to December, both on opening and closing movement, is somehow a surprise, as one could expect that door operation duration should be almost constant. However, apart from equipment failures, changes in environment temperature and thus in air density, snow, or rain can introduce changes in the door movement. In fact, on one hand it must be taken in consideration that air is what drives this door and on the other hand that rain water or snow can be an important obstacle to door displacement.

Table 4 - Cycle Duration - Descriptive Statistics

Duration [1/100 sec.]	Door 1			Door 1 - Opening				Door 1 - Closing			
	Total	Open	Close	Sept	Oct	Nov	Dec	Sept	Oct	Nov	Dec
Minimum	135	135	216	226	251	135	200	216	237	223	230
Q1	272	278	264	270	277	289	288	232	271	273	276
Q2	282	288	277	276	285	297	297	242	279	281	284
Q3	294	299	286	281	293	306	311	265	286	291	296
Maximum	1.981	1.981	1.588	1040	1981	781	578	768	638	1588	726
Q3 + 1,5 x IQR	327	331	319	298	317	332	346	315	309	318	326
Q3 + 3,0 x IQR	360	362	352	314	341	357	381	364	331	345	356
Mean	285	294	276	280	292	302	302	251	279	289	288
Std. Dev	54	59	46	39	96	37	22	38	24	67	28
N.º Cycles	4.590	2.316	2.274	578	628	630	480	569	612	628	465

Table 5 - Cycle occurrence distribution

N.º Cycles	Door 1			Door 1 - Opening				Door 1 - Closing			
	Total	Open	Close	Sept	Oct	Nov	Dec	Sept	Oct	Nov	Dec
N.º Cycles	4.590	2.316	2.274	578	628	630	480	569	612	628	465

Regarding the cycle duration evolution from September to December, on Figure 10 it is possible to have a graphical perspective on the change across months.

Concerning the pressure signature, the pneumatic circuit operates at 3 bar, so correct pressure readings must be in the 0 to 3 bar interval, which was the case for all the observations. Nevertheless, the minimum pressure for both inlet and outlet chambers at November, 0,012 bar, is a surprising figure, since it was expected to be 0,000 bar, as it happened in the other months.

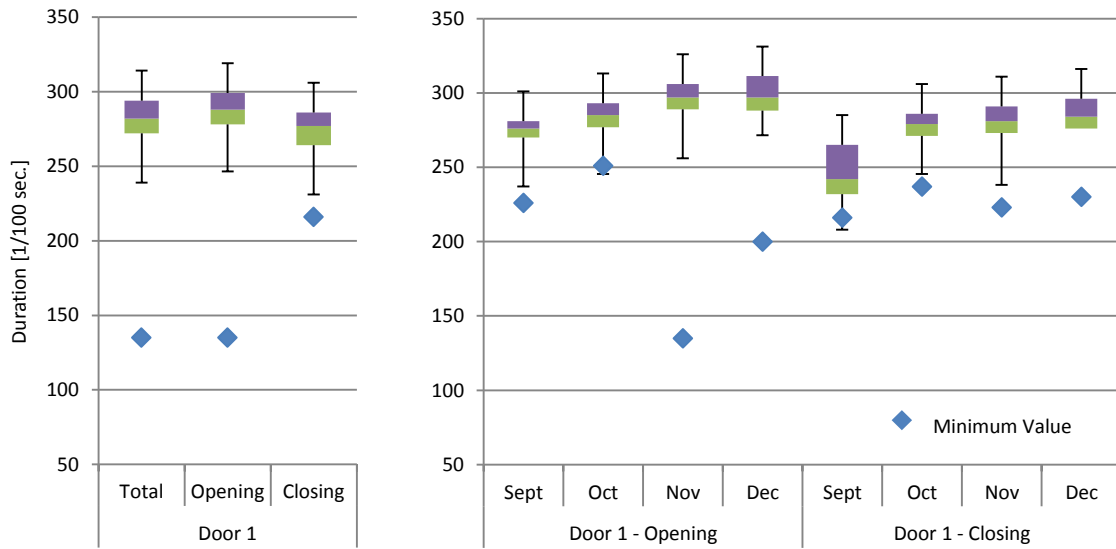


Figure 10 - Cycle Duration Boxplot and minimum value

Table 6 contains descriptive statistics for pressure readings in both chambers of the door pneumatic actuator and from its observation one can spot the constant difference at the mean, with a higher pressure at the close chamber. On the other hand, the apparent steady behaviour across months is not confirmed at a statistical level. These conclusions have statistical significance, has showed at Figure 11, Figure 12 and Figure 13, outputs of the SPSS software.

Table 6 - Pressure - Descriptive Statistics

Pressure [bar]	Open					Close				
	Total	Sept	Oct	Nov	Dec	Total	Sept	Oct	Nov	Dec
Mean	1,358	1,359	1,371	1,349	1,355	1,561	1,576	1,554	1,565	1,551
Std. Dev	1,036	1,033	1,037	1,035	1,038	1,042	1,040	1,042	1,042	1,041
Maximum	2,970	2,934	2,970	2,958	2,960	2,988	2,988	2,958	2,964	2,946
Minimum	0,000	0,000	0,000	0,012	0,000	0,000	0,000	0,000	0,012	0,000

Figure 11 - Open Pressure across Months – Statistical Test

Hypothesis Test Summary				
	Null Hypothesis	Test	Sig.	Decision
1	The distribution of Pressure is the same across categories of Month.	Independent-Samples Kruskal-Wallis Test	,010	Reject the null hypothesis.
Asymptotic significances are displayed. The significance level is ,05.				

Figure 12 - Close Pressure across Months - Statistical Test

Hypothesis Test Summary				
	Null Hypothesis	Test	Sig.	Decision
1	The distribution of Pressure is the same across categories of Month.	Independent-Samples Kruskal-Wallis Test	,010	Reject the null hypothesis.
Asymptotic significances are displayed. The significance level is ,05.				

Figure 13 - Open and Close Mean Pressure Difference - Independent Samples Test

Duration	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	Df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
Equal variances assumed	1.230	,000	-47,054	231.951	,000	-,202987	,004314	-,211442	-,194532
Not equal variances assumed			-47,054	231.943	,000	-,202987	,004314	-,211442	-,194532

3.3.2 Data Transformation

Considering that our plan involved working with classification algorithms, we defined an attribute value matrix with each tuple representing a cycle as our input data.

For that purpose, we have created a new set of 5 variables, as described in detail below.

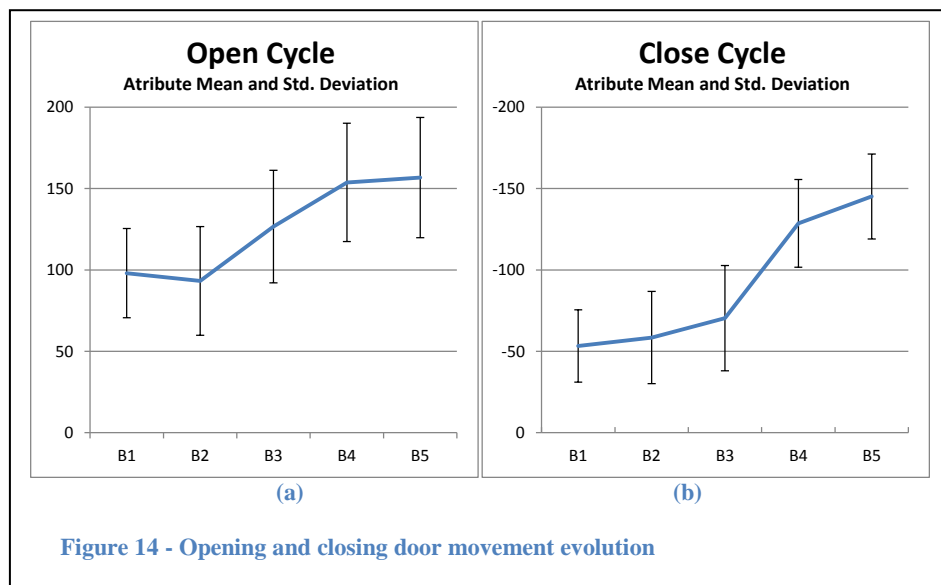
In order to transform a time series dataset into an attribute value matrix, we started by calculating the difference between the inlet and outlet pressure at each moment.

Then, for each cycle, we considered 5 bins of equal time length and calculated the average pressure for each one. Having in mind that the duration of each bin, and therefore the total cycle length, was a vital information, we generated 5 new variables, multiplying the bin average pressure by its duration. Finally, we could rearrange our

dataset, transforming 232 thousand pressure readings into 4950 door cycles, described by 5 variables.

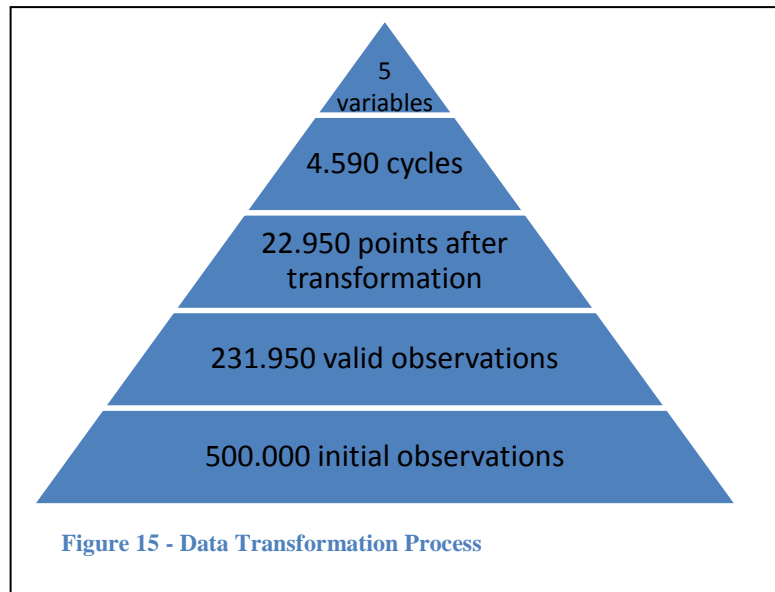
The new set of attributes was named B1 to B5, with B1 being the first bin, when the door has just started to move, and B5 corresponding to the last bin, when the cycle finished.

Figure 14 shows the evolution of pressure and duration of the two types of cycles, opening and closing door movements, by the mean and standard deviation of each of these 5 bins.

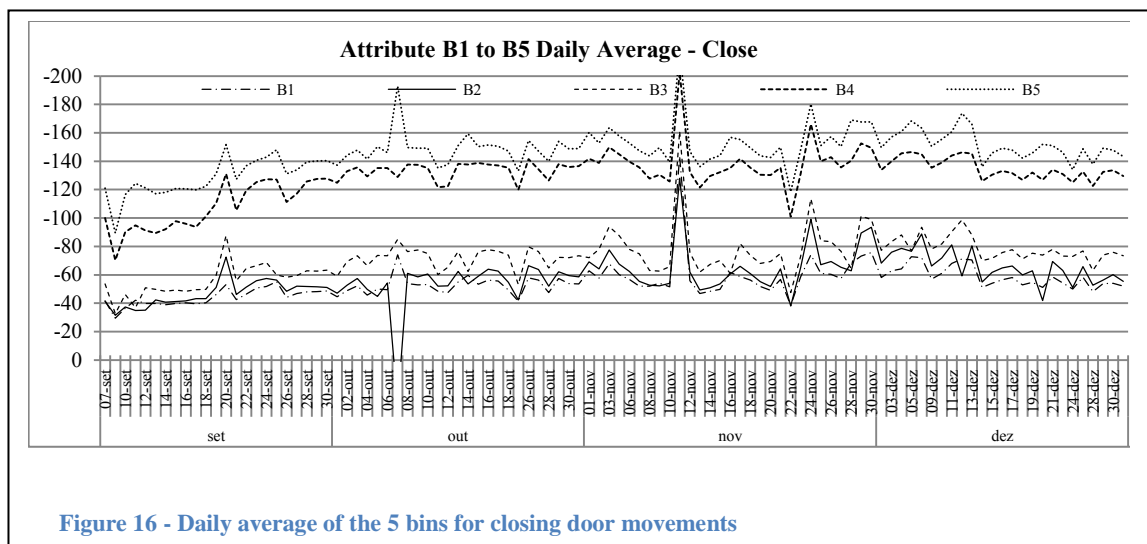


To sum up the importance of the data transformation process, Figure 15 illustrates the reduction from 500.000 observations to 4.590 cycles, each with 5 attributes.

Bearing in mind that the temporal information was an important aspect of the dataset, as showed in Figure 10, daily averages were also calculated for each attribute across all period.



From the analysis of Figure 16 one can observe that, especially in closing movement, attributes average suffered an important shift along September, in what could be concept evolution or the



development of a failure.

In order to allow for an overall understanding of the new 5 variables, used to describe the behaviour of each cycle, Table 7 shows their descriptive statistics. Besides, Figure 17 and

Figure 18 are SPSS software outputs for statistical distribution tests.

Table 7 - Attributes B1 to B5 - Descriptive Statistics

	Open					Close				
	B1	B2	B3	B4	B5	B1	B2	B3	B4	B5
Mean	98	93	127	154	157	-53	-58	-70	-129	-145
Std. Dev	27	33	35	36	37	22	28	32	27	26
Maximum	949	1135	1111	1131	1114	0	158	326	0	0
Minimum	1	-62	-82	-140	-199	-931	-939	-941	-941	-887
Q1	91	85	117	147	151	-58	-66	-80	-139	-152
Q2	96	92	124	153	157	-52	-58	-72	-133	-146
Q3	101	97	134	160	163	-47	-50	-61	-123	-138
IQR	10	12	17	13	12	11	16	19	16	14

Figure 17 - Open Cycle Attrib. B1 to B5 Distribution Test

Hypothesis Test Summary				
	Null Hypothesis	Test	Sig.	Decision
1	The distribution of B1 is normal with mean 98,028 and standard deviation 27,38.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
2	The distribution of B2 is normal with mean 93,277 and standard deviation 33,42.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
3	The distribution of B3 is normal with mean 126,646 and standard deviation 34,52.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
4	The distribution of B4 is normal with mean 153,767 and standard deviation 36,39.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
5	The distribution of B5 is normal with mean 156,774 and standard deviation 36,93.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is ,05.

Figure 18 - Close Cycle Attrib. B1 to B5 Distribution Test

Hypothesis Test Summary				
	Null Hypothesis	Test	Sig.	Decision
1	The distribution of B1 is normal with mean -53,331 and standard deviation 22,22.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
2	The distribution of B2 is normal with mean -58,439 and standard deviation 28,31.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
3	The distribution of B3 is normal with mean -70,388 and standard deviation 32,29.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
4	The distribution of B4 is normal with mean -128,631 and standard deviation 26,93.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
5	The distribution of B5 is normal with mean -145,131 and standard deviation 26,09.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is ,05.

3.3.3 Labelling

Regarding labelling the dataset, two new attributes were introduced, one about the normality of the cycle itself, the cycle class, and another, the sequence class, with information on if a particular sequence of cycles should be considered as Abnormal or Normal.

As usual for these tasks, a domain expert was called in to classify each tuple in the data matrix.

It is important to notice that a cycle can be labelled as Abnormal even though there is no failure associated. Such case may arise from a door being blocked by a passenger, which must not be considered as a door failure. As for the sequence class label, door failure moments were identified from the Maintenance Reports, Annex IV – Maintenance Reports, Figure 34 and Figure 35, and failure windows were determined by encompassing the cycles that occurred before and that should be related to one specific failure.

In the end, after expert classification, there were 194 door cycles labelled as Abnormal, less than 3% of the total and 3 failure events, one of them occurring in both the opening and closing door movements (cf. Table 8 and Table 9).

Month	Total Cycles		Abnormal Cycles		% Abnormal Cycles	
	Open	Close	Open	Close	Open	Close
September	578	569	20	45	3%	8%
October	628	612	20	9	3%	1%
November	630	628	49	25	8%	4%
December	480	465	19	7	4%	2%
Total	2316	2274	108	86	5%	4%

Table 8 - Identification of Abnormal cycles by domain expert

Door Failure	Week	Date	Nr Abnormal Cycles	
			Open	Close
Failure 1	36	07-09-2012	52	0
Failure 2	48	28-11-2012	18	0
		29-11-2012	29	31
		30-11-2012	22	24
Failure 3	49	06-12-2012	33	0

Table 9 - Identification of Failures by domain expert

4 EXPERIMENTAL TESTING

The project's goal is to create a data mining system that issues an alert when a specific train door is about to have a failure. As further explained in this chapter, our strategy involved using a cycle classification technique combined with a low-pass filter to post process the output in order to spot failure development. Section 4.1.1 contains a description on the various cycle classification methodologies applied, whereas in Section 4.1.2 we describe sequence classification using the low-pass filter. At last, in Section 4.2 we present the experimental results of our work and in Section 4.3 we discuss results.

4.1 Experimental Setup

As previously stated, we tested three different approaches for anomaly detection, all of them considering an attribute value dataset. Our choice was based on software packages availability and simplicity. In fact, other options could have been considered, with methods such as DTW or discord searching coming on top of the list of alternatives. But truth is that on one hand their implementation would be much harder and on the other hand, the performance attained with the simpler model allowed us to avoid the search for more complex solutions.

Cycle classification models were implemented using Microsoft Excel and Knime ([Berthold, Cebon et al. \(2008\)](#)). As for the sequence classification, all models were developed using Microsoft Excel.

In Section 4.1.1 we refer to the various cycle classification methodologies applied and in Section 4.1.2 we explain the use of a low-pass filter to achieve sequence classification.

4.1.1 Cycle Classification

To address the cycle classification problem we have divided it into two smaller problems: one for the opening door movements; and another for the closing door movements.

For each problem we have tested cycle classification under 3 different approaches, 1) unsupervised learning based on boxplot; 2) semi-supervised learning with OneClassClassification; 3) supervised learning with Support Vector Machine.

In order to maintain the work as close as possible to a real scenario, training was done using examples belonging only to the two previous weeks, except in the supervised case where we also included the Abnormal examples occurred before that time window. The decision model is then used to predict the following week.

To assess the performance of the classification task we used the approach suggested by [Hempstalk, Frank et al. \(2008\)](#). For each classification we calculated 2 ratios: false alarm rate (FAR) and the impostor pass rate (IPR). The false alarm rate is the ratio of normal instances incorrectly identified as outliers.

The impostor pass rate is the ratio of outlier instances that are wrongly classified as normal.

These metrics, FAR and IPR, are often used in outlier detection domains.

A good outlier detection system should have both low FAR and low IPR. However, this combination is usually hard or even impossible to obtain, Therefore, one has to find the correct balance between a classification system that spots all the outliers (low IPR) and does not wrongly classify normal observations as outliers (low FAR). Usually, a lower higher FAR results in a higher IPR and vice versa. The trade-off between IPR and FAR depends on the problem at hand. In this project, the use of a low-pass filter, post processing the output of the cycle classification, allows us to favour a low IPR even though that could be associated to a higher FAR. A more comprehensive explanation of this process is provided in Section 4.2.

4.1.2 Sequence Classification

Once cycle classification was done, further treatment has been applied to the dataset. In fact, the purpose of this work is to be able to issue an alarm whenever a door is about to have a breakdown, not to distinguish between Normal and Abnormal cycles. To achieve this part of the process, we have use a low-pass filter, as described hereafter. In each problem we have tuned the low-pass filter with a specific parameterization, setting the threshold level and smoothing factor, in order to obtain the best possible result under that specific scenario.

Low-Pass Filters

A filter is a device that removes from a signal some unwanted component or feature [Shenoi \(2005\)](#). The defining feature of filters is the complete or partial suppression of some aspect of the signal. Often, this means removing some frequencies and not others in order to suppress interfering signals and reduce background noise. There are several filters that can be designed to achieve specific goals taking application into account. A low-pass filter is a filter that passes low-frequency signals but attenuates (reduces the amplitude of) signals with frequencies higher than the cut-off frequency.

The low-pass algorithm is detailed by the equation:

$$y_i = y_{i-1} + \alpha \times (x_i - y_{i-1}),$$

where y_i is the filter output for the original signal x_i for instant i and α is the smoothing parameter.

The change from one filter output to the next is proportional to the difference between the previous output and the next input. This exponential smoothing property matches the exponential decay seen in the continuous-time system. As expected, as α decreases, the output samples respond more slowly to a change in the input samples: the system will have more inertia.

4.2 Experimental Evaluation

The main goal of this experimental study is to evaluate anomaly detection performance with a combination of cycle classification and a low-pass filter in the reduction of false alarms.

We have tested 3 decision models in which results are post-processed by the low-pass filter.

As already referred, we train a decision model using a sliding window of two weeks of data and evaluating in the following week.

A different approach, training static models with data from September, resulted in a degradation of the performance in all models as the time horizon increases.

One explanation for this behaviour might come from the attribute evolution seen in, Figure 10 regarding cycle length modification across months, and Figure 16, daily mean for attributes B1 to B5.

Bearing that in mind, another tested option was to train new updated models with the cycle classification result from previous weeks. Depending on the classifier fine tuning, the outcome of this approach provided a high FAR, that could not be attenuated by the low-pass filter, mixed with a high IPR, making the failure prediction system useless.

Finally, we tested training new updated models with the result of the low-pass filter from previous weeks. Once more results were useless with the cycle classifier totally missing the target.

This being so, we opted to focus our attention in comparing the anomaly detection performance of the two stage classification system, with a combination of cycle classification and a low-pass filter, using a data sliding window of two weeks.

4.2.1 Results using Boxplot Outlier detection

Applying the boxplot method for the 2 previous weeks for all 5 variables, outliers were detected if at least 1 of the variables value was an extreme outlier.

Even though this approach could be considered too simplistic, assuming independent and Gaussian distributions, it turned out to work very good, especially when its output was post-processed with a low-pass filter. As seen in

Figure 17 and

Figure 18, variables B1 to B5 do not follow a Gaussian distribution, so it could be argued about the statistical prerequisites to be able to run the boxplot method to find extreme outliers. Considering that this was a real world practical problem, it was decided to discard this non-compliance and apply the boxplot test.

Bearing in mind the above, surprisingly this method granted accurate results, with low IPR and manageable FAR from the cycle classification level.

This performance (see Table 10 and Figure 19) was then enhanced with the low-pass filter setting the threshold at 0.5 and smoothing factor at 0.15.

False Alarms	Open			Close			
	Before Filter	After Filter		Before Filter	After Filter		
	68	0		76	1		
Cycle Label	Abnormal	Normal	Total	Abnormal	Normal	Total	
Abnormal	103	5	108	66	20	86	
Normal	11	2197	2208	24	2164	2188	
Cycle Classification		Opening			Closing		
		W49	W39	Other	W48	W38	Other
	False Alarm Rate	44%	43%	4%	67%	0%	19%
	Impostor Pass Rate	0%	0%	5%	0%	45%	13%

Table 10 - Results using boxplot-based outlier detection

In the end, this system was able to correctly detect the 3 failures present in the dataset with small and acceptable lag, with just one incipient False Alarm at week 45.

Overall, both failures on the end of week 48 and 49 could have been signalled with at least 24 hour anticipation.

Figure 19 shows the ability of the system to raise alarms short after the expert had marked the beginning of the failure. Taking a closer look at the chart, namely at the middle of week 48 and at the end of week 49, one can see that the top lines (domain expert confirmed failure) and the middle lines (predicted failure) have an almost simultaneous sharp drop, meaning that the anomaly detection system could find the failure initial development. On the other hand, during the rest of the period, even though there were a lot of cycles classified as Abnormal, their prevalence was not sufficient to set the low-pass signal below the threshold.

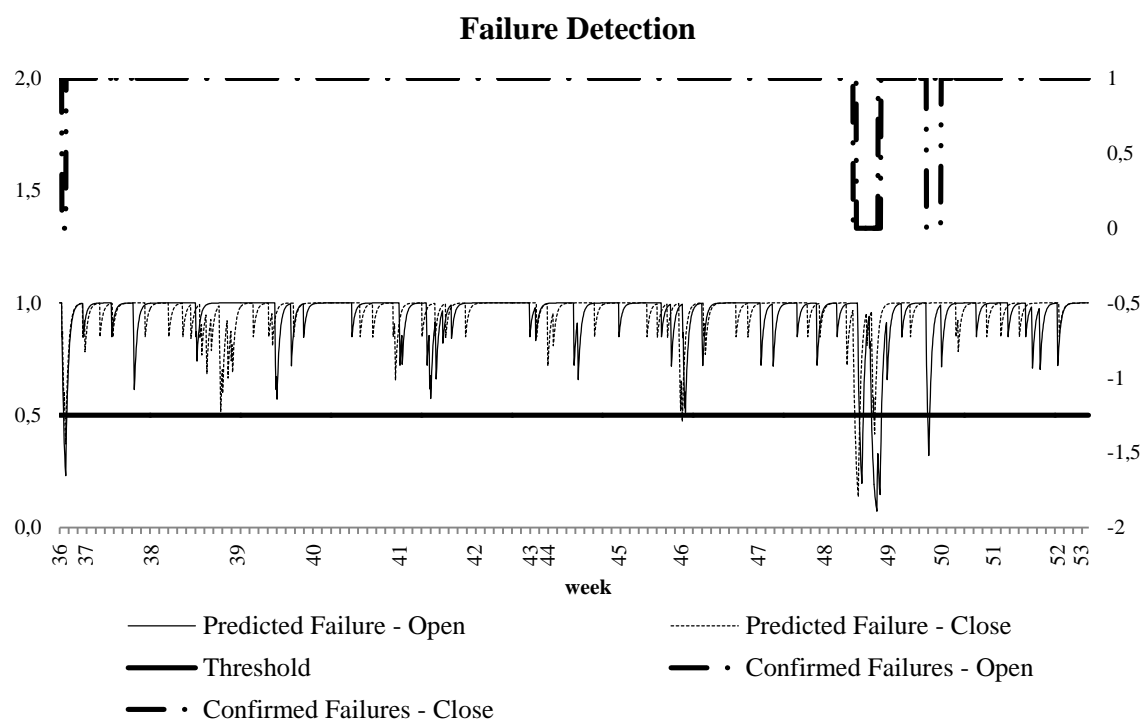


Figure 19 - The impact of the low-pass filter using boxplot based outlier detection

4.2.2 Results using Novelty Detection

As stated before, in a failure detection problem we have to train the classifier with examples only belonging to normal class or at least with unbalanced datasets.

To deal with this challenge, we have used the OCC algorithm available in Weka ([Hempstalk, Frank et al. \(2008\)](#)), which combines density and class probability estimation. The method we have chosen for the class probability estimation was a

decision tree with pruning. To reduce variance, bagging was applied with 10 time iteration and bag size set to 100%.

The OCC Weka implementation, depicted in Figure 20, was trained with Normal examples from the preceding two weeks, previously labelled by an expert. Therefore, classification for examples from week “ w ” was done learning the classifier from expert labelled examples from week “ $w-2$ ” and “ $w-1$ ”.

In a brief explanation regarding the Knime data flow, row splitter node named “Norm/Ab” creates two tables, one for the Normal examples and another for the Abnormal ones. Then, node 425 filters examples for weeks “ $w-2$ ” and “ $w-1$ ”. The output of node 422 only contains Abnormal examples, that are disregarded in the OCC Learner.

Once the classifier is trained, node 421 output, the examples for week “ w ”, output of node 427, are labelled as Normal or Abnormal in node 426. A final table gathering the classification for all tuples is built in node 432 and 434.

In the end, evaluation is done comparing cycle expert label to cycle automatic classification.

Although the training dataset used as input for the OCC Learner node contained Abnormal examples, they were not considered by the specific applied OCC algorithm. The existence of Abnormal examples in the OOC Leaner node input was a consequence of sharing most of the workflow for both the OCC and SVM experiments.

Concerning the semi-supervised method performance, when we were considering aggregated cycle classification, it looked satisfactory, comparing to unsupervised classification. But a closer look at a week level showed an enormous concentration on False Alarms (see Table 11 and Figure 21). In fact, the false alarms recorded from week 38 to 41 could not be handled by the low-pass filter.

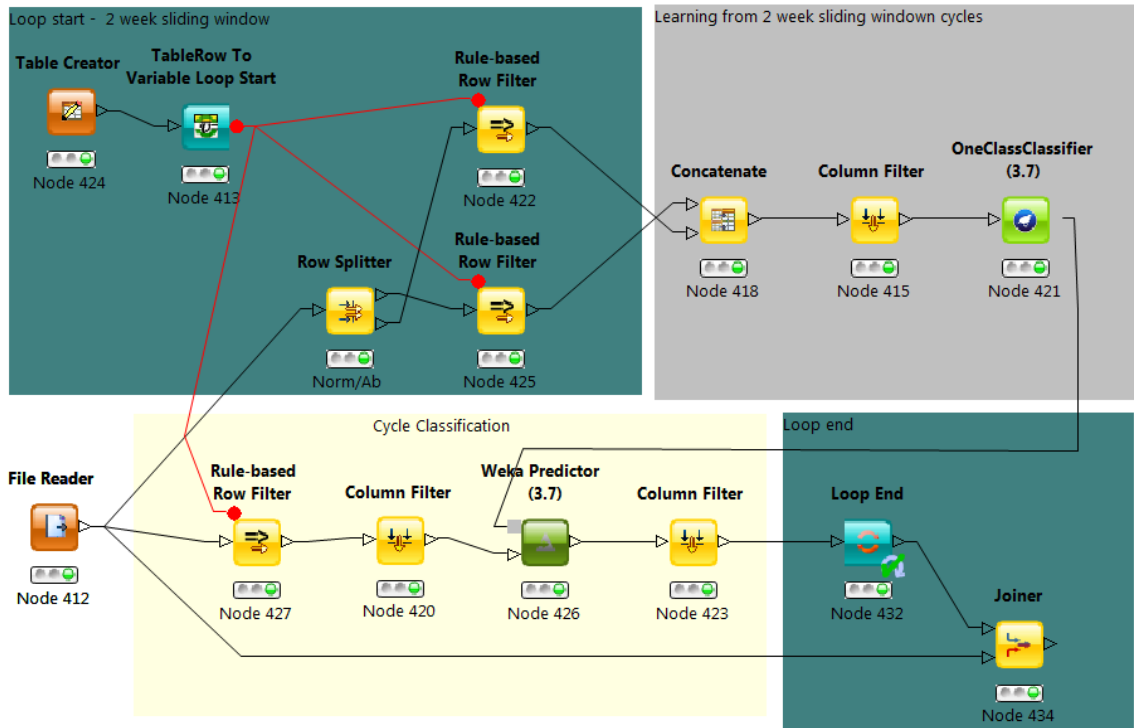


Figure 20 - OCC Knime workflow

As mentioned before, the false alarm concentration in week 39 could not be enough attenuated by the low-pass filter, causing several incorrect door failure alarms.

In this scenario we can distinguish between the performance achieved before week 43 and after. Until week 43 the system was unable to correctly identify Abnormal cycles, raising incorrect false alarms, whereas afterwards the level of accuracy increased significantly, even though clearly worse than the obtained with the unsupervised method.

	Open			Close		
	Before Filter	After Filter		Before Filter	After Filter	
False Alarms	171	5		319	18	
Cycle Label	Abnormal	Normal	Total	Abnormal	Normal	Total
Abnormal	105	3	108	81	5	86
Normal	133	2075	2208	280	1908	2188
Cycle Classification	Opening			Closing		
	W41	W49	Other	W38	W39	Other
False Alarm Rate	69%	82%	49%	56%	98%	70%
Impostor Pass Rate	0%	0%	3%	0%	0%	9%

Table 11 - Results using OCC for novelty detection

One justification for this behaviour might come from concept evolution. When looking at the evolution of the attributes along the time window, there is a strong change in the daily average from week 36 to 40, but nevertheless there were no records of door failures. As one might expect, the OCC classifier was not able to spot the new outliers when the data model was evolving at that pace.

Figure 21 shows the unstable behaviour of the combination of OCC and low-pass filter across weeks. Taking a closer look at the chart, namely after week 43, one can see an acceptable performance with middle lines (predicted failure) showing some parallelism to the top lines (domain expert confirmed failure). Problems arise at the initial part of our study, with at least 3 important False Alarm zones in weeks 38 and 39.

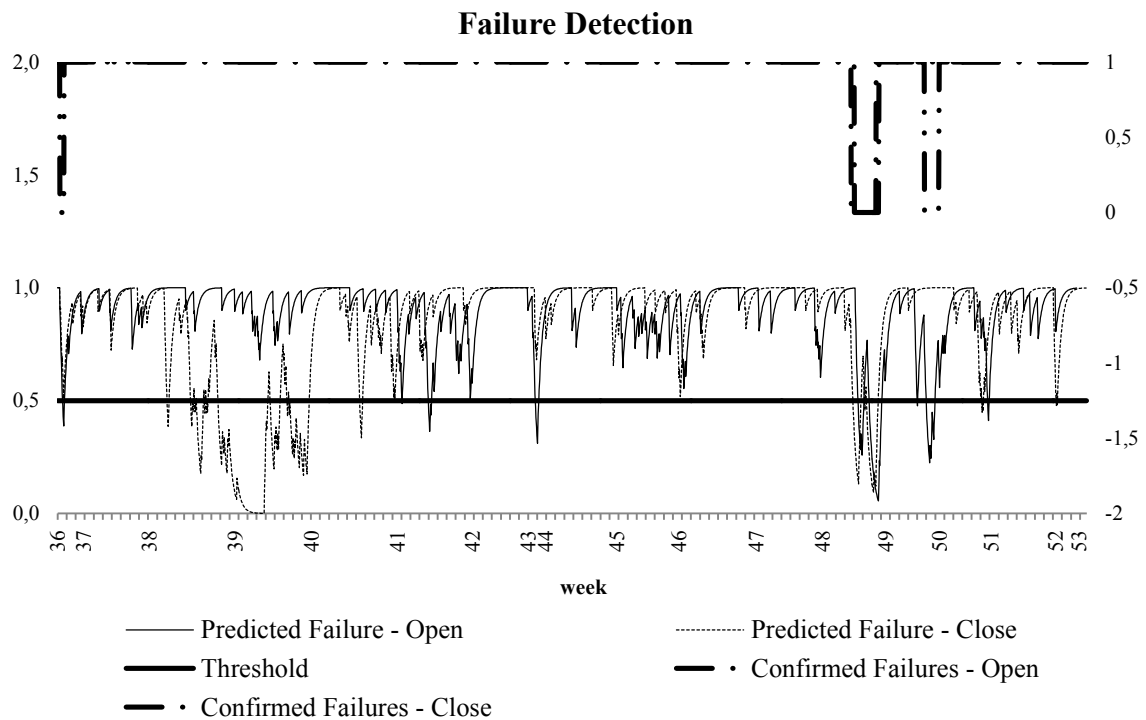


Figure 21 - The impact of the low-pass filter using OCC for novelty detection

On the overall this system clearly provided poorer performance when comparing to the boxplot, even though more human time consuming. In fact, unlike the boxplot, to run this model it was needed to have an expert classifying the observations from the 2 previous weeks.

If this method was to be implemented in a real world environment, the lack of ability to auto update the classifier could be a deal breaker. On a business level, the application of a data mining system is usually associated with less human intervention and not the opposite. Depending on the application, having an anomaly detection system that relies on a human expert constantly preparing the training set could be regarded as useless or not economically feasible.

4.2.3 Results using Supervised Classification

The last experiment conducted in this study was testing the application of a supervised classifier.

After initial trials, we have chosen a Support Vector Machine ([Platt \(1999\)](#)), as implemented in Knime and shown in Figure 22. Knime workflow was shared, as far as possible, with OCC implementation.

As previously mentioned, SVM training was done in a two week sliding window training set, including all the Abnormal examples that had already occurred, trying to deal with an unbalanced dataset. It is this fact that forced the use of two row filters, nodes 428 and 424, allowing us to build a dataset composed by Normal examples for weeks “ $w-1$ ” and “ $w-2$ ” and all Abnormal examples occurred until week “ w ”.

Final SVM configuration involved a linear Kernel function and γ set at 0.2 with an overlapping penalty of 1.0.

Other Kernel functions were tested, but with worse results. On the other hand, changes in γ and overlapping penalty did not show a relevant impact on the cycle classification performance.

When discussing final results, once more the overall outcome seemed acceptable (see Table 12 and Figure 23), but looking at a week level showed the lack of capacity of SVM to deal with this problem.

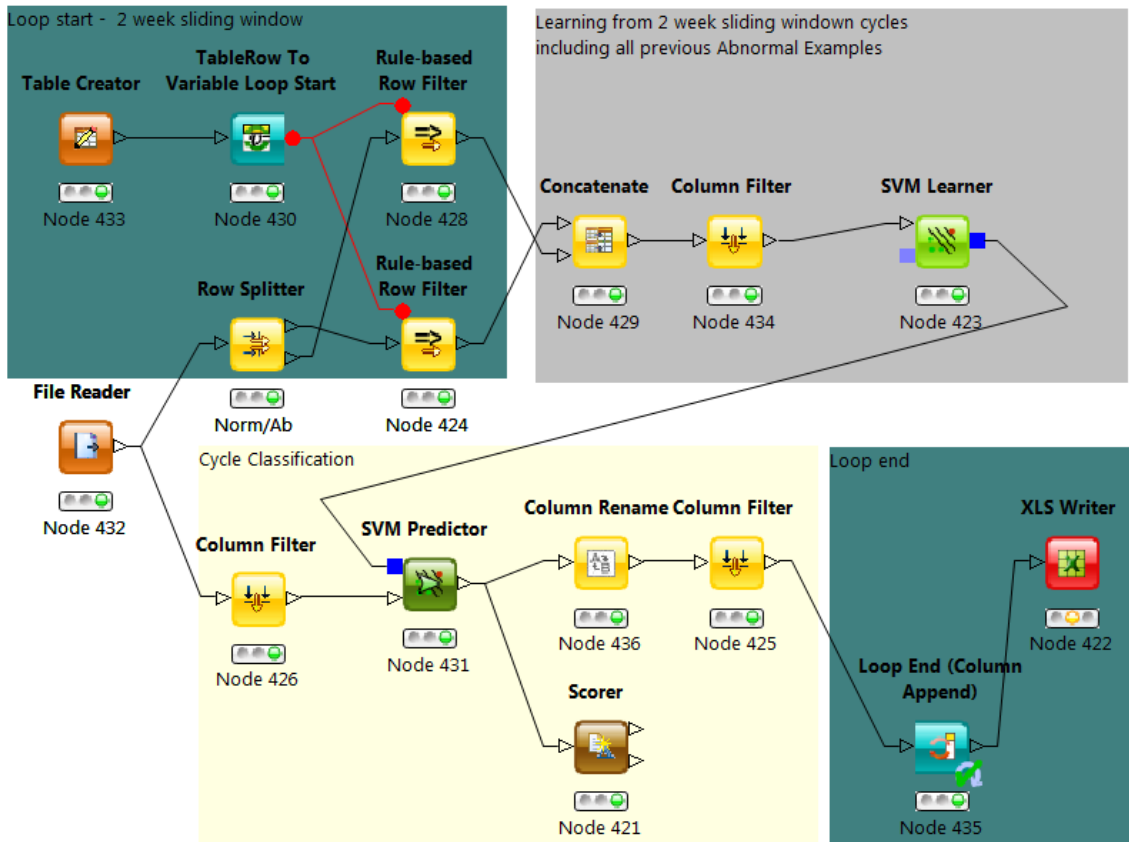


Figure 22 - SVM Knime workflow

In fact, false alarm rate from week 38 to 39 was so high that it was impossible for the low-pass filter to accommodate all the errors. Moreover, SVM was not able to spot 30 Abnormal cycles in week 48, missing the 29-30 November failure on the opening movement.

To sum up the SVM experiments, one has to mention its lack of capacity to spot almost all outliers at weeks 38 and 48. This inability to detect the Abnormal cycles is crucial in a failure detection system.

False Alarms	Open			Close		
	Before Filter	After Filter		Before Filter	After Filter	
	77	3		170	11	
Cycle Label	Abnormal	Normal	Total	Abnormal	Normal	Total
Abnormal	66	42	108	47	39	86
Normal	45	2163	2208	137	2051	2188

Cycle Classification	Opening			Closing		
	W48	W49	Other	W38	W39	Other
False Alarm Rate	0%	86%	19%	11%	99%	33%
Impostor Pass Rate	88%	0%	0%	74%	50%	28%

Table 12 - Results using SVM supervised learning

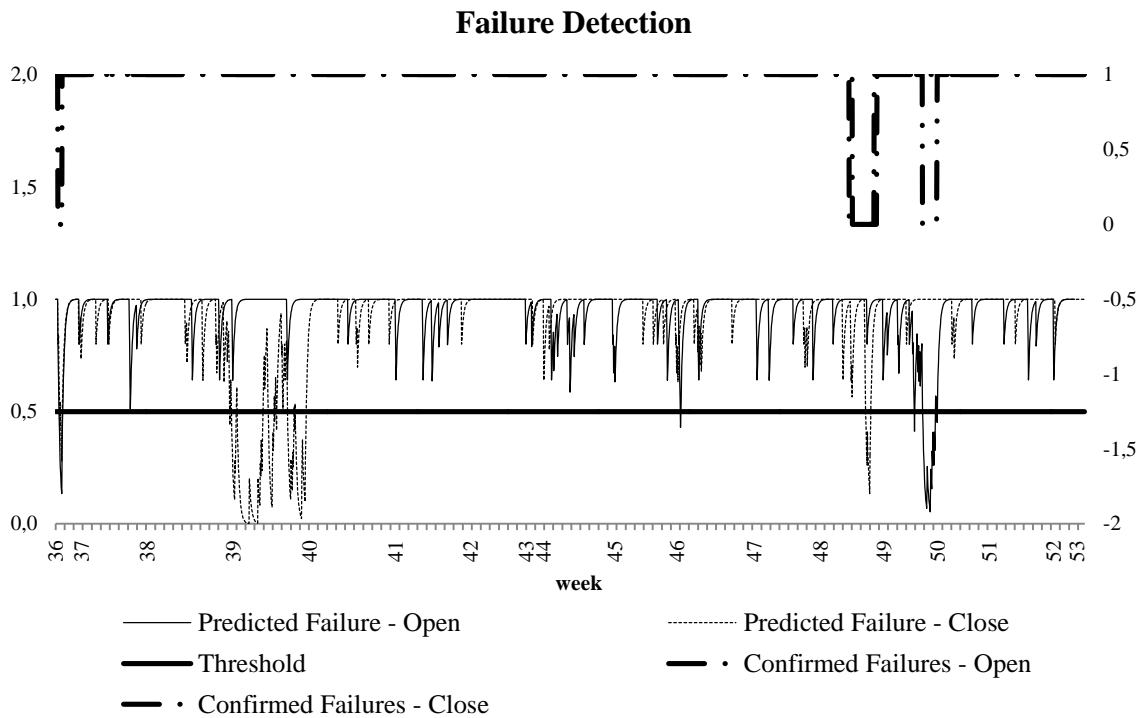


Figure 23 - The impact of the low-pass filter using SVM supervised learning

4.3 Discussion

As stated before, we found that the best approach for our anomaly detection problem was the combination of an unsupervised outlier detection system with a low-pass filter. Even though this 2 stage methodology, where the cycle classification output is post-processed by a low-pass filter, can be considered a slick solution, it came as a surprise that the best cycle classification was achieved with a *straight-forward* extreme outlier detection.

In fact, [Chandola \(2009\)](#) states that failure detection techniques for anomaly detection in engineering applications can spread from Parametric or Non-Parametric Statistical Modelling to Neural Networks. But the truth is that most of the successful applications done in this field apply a One Class Classification approach, with a 2 class classifier, such as an autoencoder.

Bearing this situation in mind, we tried to understand what could be the main reasons for the extreme global outlier detection so good performance.

Operation Regime

In standard engineering applications, equipment is subjected to different operation regimes, we could think about helicopter gearboxes. Gearbox regime is dependent on parameters such as engine rpm, engine power, air density, wind and so on; therefore it is hard to define what should be considered a normal behaviour, as there are some many normal operation regimes.

On the other hand, train doors are expected to work pretty much the same throughout their life cycle. One can accept that pneumatic door opening duration might be a little different in winter from summer, due different to air density or humidity rate, but on the overall, each door normal cycle should always be almost the same. This feature is the main reason for the results achieved with the extreme global outlier detection.

5 CONCLUSIONS AND FUTURE WORK

The study main goal was to point out the high value of data mining tools for maintenance management, with a specific study case in the railway field. To attain that purpose it was decided to develop a data mining system that issues an alarm whenever is predictable that an automatic train door is about to have a failure.

Initial dataset presented 500 thousands observations corresponding to opening and closing doors episodes as time-series. Data transformation, involving discretization, was applied to allow for an easier and quicker computer processing and its representation in an attribute value matrix.

The anomaly detection system consisted in a two stage classification. First, we apply a cycle classification technique and then, the output is post-processed with a low-pass filter.

Three different lines were tested for abnormal cycle detection: unsupervised, semi-supervised and supervised. We started with an unsupervised approach, spotting outliers as examples with at least one attribute with an outlier value determined by the boxplot method. Classification results were processed with a low-pass filter, enabling us to anticipate door failures in 24 hours.

One Class Classification algorithm, a semi-supervised technique, was also tested, an approach widely used for novelty detection, but unfortunately we were not able to guarantee an adequate level of reliability, presumably due to concept evolution.

At last, a more standard, supervised, method was experimented, trying a two class classification problem using a Support Vector Machine. SVM granted good aggregated results, but on the opening cycles it totally missed the most important door failure at the end of week 48 and on the closing movements the False Alarms issued at week 38 and 39 must be considered an important weakness.

In the end we have demonstrated that, at least in this specific problem, with a small investment in sensors, data logging and data mining techniques we are able to minimize

maintenance costs and increase system's reliability, since the 3 failures presented in the dataset were spotted in their development phase.

On an academic level, we have come up with a failure prediction system based on a two stage algorithm: 1) event classification and 2) sequence analysis applying a low-pass filter. Standard approaches in this field do not take into account that in some projects we need to predict a breakdown, not to distinguish between Normal and Abnormal cycles. According to our study, the combination of cycle classification technique and the post-processing of the output with a low-pass filter can provide very interesting results.

A failure detection project is always prone to future developments. Vectors such as lead time or reliability can always be improved with practical implications for those who apply this kind of methodologies in maintenance management.

This project was set on data collected from September to December 2012 and considering that we have observed attributes values changing across months, we would like to verify our conclusion on the boxplot good performance over a larger time frame, including examples from a whole year.

The methodology we presented in this document relies solely in the combination of boxplot outlier detection and low-pass filter, but considering that there are some golden rules that could be considered, e.g. a normal cycle is always less than 3,5 seconds, it could be profitable to include some domain expert rules in cycle classification.

It is also worth considering validating the purposed methodology on another kind of pneumatic door. But maybe it would be more interesting to apply it on an electrical door, imagining the information available at the air pressure development is the same found in electrical current development on an electrical motor.

At last, we would like to address the lead time increase. How could we anticipate the moment when the alarm is raised without compromising the systems reliability in avoiding false alarms? Low false alarm rate and alarm anticipation are apparently two

On a business level, the next step is the practical implementation of this methodology in the Nomadtech maintenance software, which is under evaluation, after more comprehensive validation tests are done.

6 Bibliography

- Aggarwal, C. C. (2013). Outlier Analysis, Springer New York.
- Angeli, C. and A. Chatzinikolaou (2004). "On-Line Fault Detection Techniques for Technical Systems: A Survey." IJCSA **1**(1): 12-30.
- Basseville, M. (1988). "Detecting changes in signals and systems—a survey." Automatica **24**(3): 309-326.
- Basseville, M. and I. V. Nikiforov (1993). Detection of abrupt changes: theory and application, Prentice Hall Englewood Cliffs.
- Berthold, M. R., N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel and B. Wiswedel (2008). KNIME: The Konstanz information miner, Springer.
- Breunig, M. M., H.-P. Kriegel, R. T. Ng and J. Sander (2000). LOF: identifying density-based local outliers. ACM Sigmod Record, ACM.
- Chandola, V. (2009). "Anomaly detection: A survey." ACM Computing Surveys (CSUR) **41**(3): 15.
- Chandola, V. (2009). "Detecting Anomalies in a Time Series Database Varun Chandola, Deepthi Cheboli, and Vipin Kumar."
- Chapman, P., J. Clinton, T. Khabaza, T. Reinartz and R. Wirth (1999). "The CRISP-DM process model." The CRIP-DM Consortium **310**.
- Fink, O., E. Zio and U. Weidmann (2013). Extreme learning machines for predicting operation disruption events in railway systems. Proceedings of the European Safety and Reliability Conference.
- Fujiwara, T. (1995). "Process Modelling for Fault Detection Using Neural Networks." Neural Networks for Chemical Engineers,(ed.) AB Bulsari: 283-296.
- Gama J, R. P., Spinosa EJ, Carvalho A (2010). Knowledge discovery from data streams.
- Han, J. and M. Kamber (2011). Data Mining: Concepts and Techniques: Concepts and Techniques, Elsevier Science.
- Hardman, W., A. Hess and J. Sheaffer (2000). A helicopter powertrain diagnostics and prognostics demonstration. Aerospace Conference Proceedings, 2000 IEEE, IEEE.
- Hawkins, D. M. (1980). Identification of outliers, Chapman and Hall London.
- He, H. and E. A. Garcia (2009). "Learning from imbalanced data." Knowledge and Data Engineering, IEEE Transactions on **21**(9): 1263-1284.

- Hempstalk, K., E. Frank and I. H. Witten (2008). One-class classification by combining density and class probability estimation. Machine Learning and Knowledge Discovery in Databases, Springer: 505-519.
- Hoskins, J. C. and D. Himmelblau (1988). "Artificial neural network models of knowledge representation in chemical engineering." Computers & Chemical Engineering **12**(9): 881-890.
- Isermann, R. (1984). "Process fault detection based on modeling and estimation methods—a survey." Automatica **20**(4): 387-404.
- Iverson, D. L. (2008). "Data Mining Applications for Space Mission Operations System Health Monitoring." Proc. of the Space Ops.
- Japkowicz, N., C. Myers and M. Gluck (1995). A novelty detection approach to classification. IJCAI.
- Katipamula, S. and M. R. Brambley (2005). "Review article: methods for fault detection, diagnostics, and prognostics for building systems—a review, Part I." HVAC&R Research **11**(1): 3-25.
- Keogh, E., J. Lin and A. Fu (2005). Hot sax: Efficiently finding the most unusual time series subsequence. Data mining, fifth IEEE international conference on.
- Keogh, E. and C. A. Ratanamahatana (2005). "Exact indexing of dynamic time warping." Knowledge and information systems **7**(3): 358-386.
- Keogh, E. L., Stefano (2002). Finding surprising patterns in a time series database in linear time and space. Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.
- Lin, J., E. Keogh, S. Lonardi and B. Chiu (2003). A symbolic representation of time series, with implications for streaming algorithms. Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery.
- Lonardi, S., J. Lin and E. Keogh (2006). "Efficient discovery of unusual patterns in time series." New Generation Computing **25**(1): 61-93.
- Markou, M. and S. Singh (2003). "Novelty detection: a review—part 1 statistical approaches." Signal processing **83**(12): 2481-2497.
- Martinez-Heras, J.-A., A. Donati, M. G. F. Kirsch and F. Schmidt (2012). New Telemetry Monitoring Paradigm with Novelty Detection. SpaceOps, Stockholm.
- Nowlan, F. S. and H. F. Heap (1978). Reliability-centered maintenance, DTIC Document.

- Papadimitropoulos, A., G. A. Rovithakis and T. Parisini (2007). "Fault detection in mechanical systems with friction phenomena: An online neural approximation approach." Neural Networks, IEEE Transactions on **18**(4): 1067-1082.
- Patton, R. J., R. N. Clark and P. M. Frank (2000). Issues of fault diagnosis for dynamic systems, Springer.
- Petsche, T. (1996). "A neural network autoassociator for induction motor failure prediction." Advances in neural information processing systems: 924-930.
- Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods, MIT press.
- Saxena, A. (2007). "Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems." Applied Soft Computing **7**(1): 441-454.
- Shenoi, B. A. (2005). Introduction to digital signal processing and filter design, John Wiley & Sons.
- Tax, D. (2001). One-Class classification: concept-learning in the absence of counter-examples. Ph.D., Delft University of Technology.
- Tukey, J. W. (1977). "Exploratory Data Analysis."
- Ungar, L., B. Powell and S. Kamens (1990). "Adaptive networks for fault diagnosis and process control." Computers & Chemical Engineering **14**(4): 561-572.
- Vaidyanathan, R. and V. Venkatasubramanian (1992). "Representing and diagnosing dynamic process data using neural networks." Engineering Applications of Artificial Intelligence **5**(1): 11-21.
- Venkatasubramanian, V. and K. Chan (1989). "A neural network methodology for process fault diagnosis." AIChE Journal **35**(12): 1993-2002.
- Venkatasubramanian, V., R. Vaidyanathan and Y. Yamamoto (1990). "Process fault detection and diagnosis using neural networks—I. Steady-state processes." Computers & Chemical Engineering **14**(7): 699-712.
- Watanabe, K., S. Hirota, L. Hou and D. Himmelblau (1994). "Diagnosis of multiple simultaneous fault via hierarchical artificial neural networks." AIChE Journal **40**(5): 839-848.
- Watanabe, K., I. Matsuura, M. Abe, M. Kubota and D. Himmelblau (1989). "Incipient fault diagnosis of chemical processes via artificial neural networks." AIChE Journal **35**(11): 1803-1812.
- Yilboga, H., O. F. Eker, A. Güçlü and F. Camci (2010). Failure prediction on railway turnouts using time delay neural networks. Computational Intelligence for Measurement Systems and Applications (CIMS), 2010 IEEE International Conference on.

Zhang, J., Q. Yan, Y. Zhang and Z. Huang (2006). Novel fault class detection based on novelty detection methods. Intelligent Computing in Signal Processing and Pattern Recognition, Springer: 982-987.

ANNEXS

Annex II – Original Variables – Statistical Tests

II.1 - Statistical Tests for Variable: Duration and Cycle: Opening

Results:

Applying Kruskal Wallis Tests it cannot be accepted that the distribution of Duration is the same across all months, at a 0.05 significance level.

Applying T-Test for 2 independent samples it can be said that at a 95% confidence Duration Mean:

- **Is** different from September to October;
- **Is** different from October to November;
- **Is not** different from November to December.

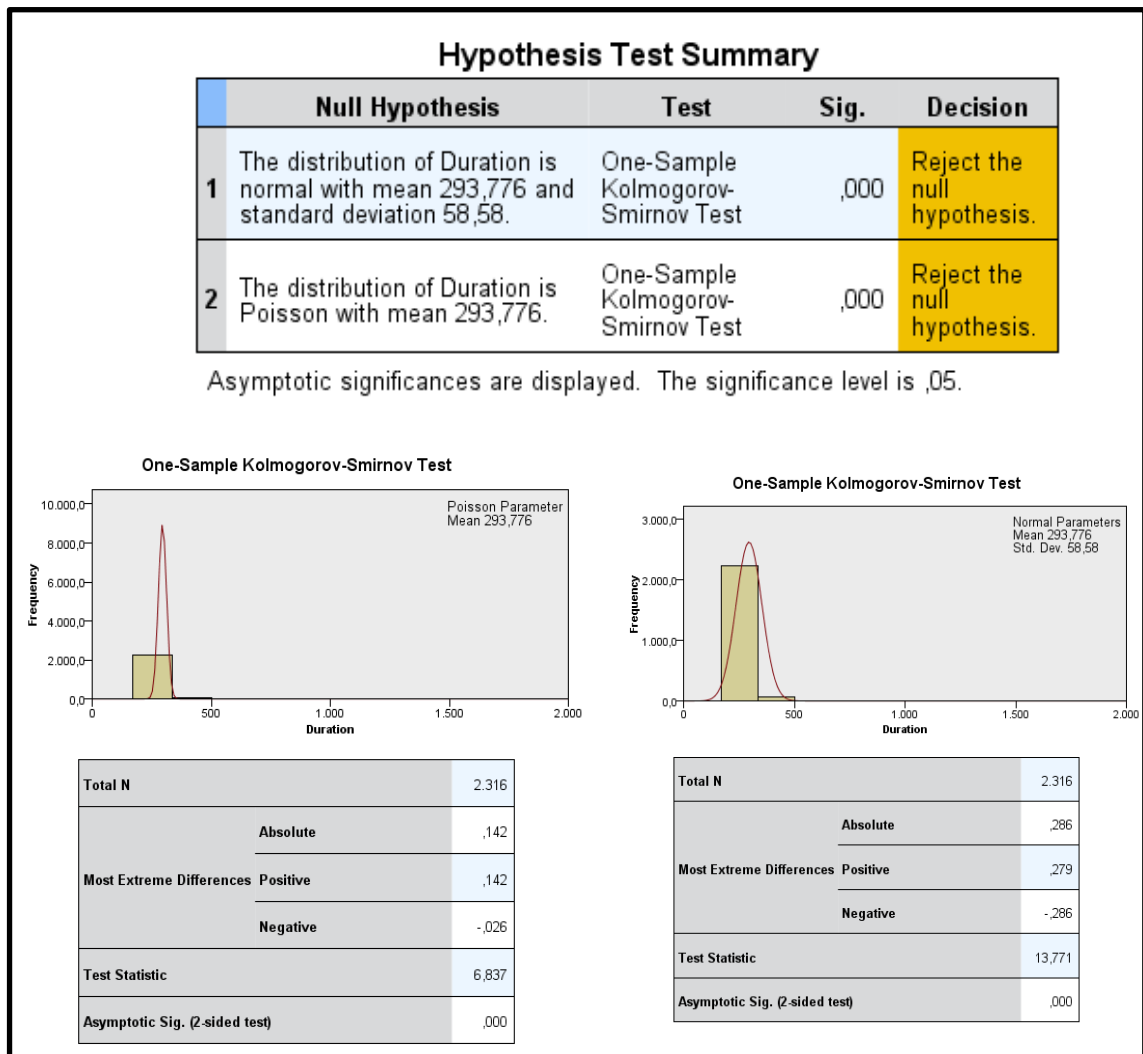


Figure 24 - Statistical Tests for Duration Distribution Cycles

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of Duration is the same across categories of Month.	Independent-Samples Kruskal-Wallis Test	,000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is .05.

Figure 25 - Statistical Test for Duration is the same across Month

Months 9, 10, 11 and 12 (September to December)

	Month	N	Mean	Std. Deviation
Duration	9	578	279,78	39,155
	10	628	292,37	96,026
	11	630	302,14	36,811
	12	480	301,50	22,262
	Total	2316	293,78	58,576

Independent Samples Test Month 9 and 10

Duration	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	Df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
Equal variances assumed	2,850	,092	-2,936	1.204	,003	-12,591	4,289	-21,006	-4,176
Equal variances not assumed			-3,024	844	,003	-12,591	4,164	-20,763	-4,419

Independent Samples Test Month 10 and 11

Duration	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	Df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
Equal variances assumed	,329	,567	-2,384	1.256	,017	-9,770	4,098	-17,810	-1,730
Equal variances not assumed			-2,381	806	,017	-9,770	4,103	-17,824	-1,717

Independent Samples Test Month 11 and 12

Duration	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	Df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
Equal variances assumed	,295	,587	,336	1.108	,737	,638	1,900	-3,090	4,366
Equal variances not assumed			,358	1.057	,721	,638	1,784	-2,863	4,139

Figure 26 - Duration across Month - Mean Difference Test – Opening Cycles

II.2 - Statistical Tests for Variable: Duration and Cycle: Closing

Results:

At a 0.05 significance level, duration distribution is neither Normal nor Poisson, as stated by the Kolmogorov-Smirnov Test.

Applying Kruskal Wallis Tests it cannot be accepted that the distribution of Duration is the same across all months, at a 0.05 significance level.

Applying T-Test for 2 independent samples it can be said that at a 95% confidence Duration Mean:

- **Is** different from September to October;
- **Is** different from October to November;
- **Is not** different from November to December.

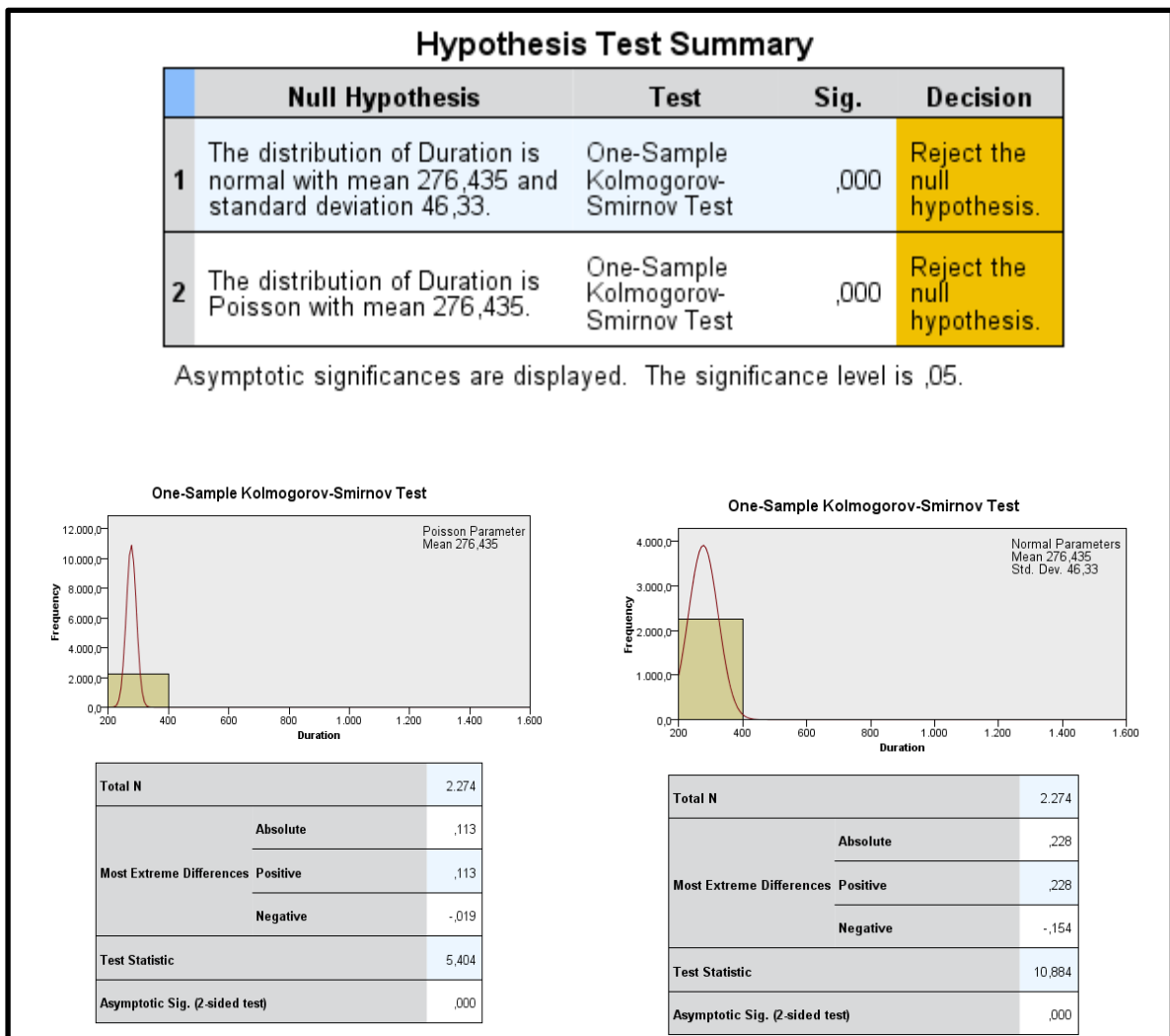


Figure 27 - Figure 35 - Statistical Tests for Duration Distribution Closing Cycles

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of Duration is the same across categories of Month.	Independent-Samples Kruskal-Wallis Test	,000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is .05.

Figure 28 - Statistical Test for Duration is the same across Month

Months 9, 10, 11 and 12 (September to December)

	Month	N	Mean	Std. Deviation
Duration	9	569	250,78	37,709
	10	612	279,33	23,594
	11	628	288,54	67,254
	12	465	287,68	28,055
	Total	2.274	276,43	46,333

Independent Samples Test Month 9 and 10

Duration	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	Df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
Equal variances assumed	30,600	,000	-15,712	1.179	,000	-28,550	1,817	-32,115	-24,985
Not Equal variances assumed			-15,464	940	,000	-28,550	1,846	-32,173	-24,927

Independent Samples Test Month 10 and 11

Duration	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	Df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
Equal variances assumed	10,197	,001	-3,201	1.238	,001	-9,208	2,877	-14,853	-3,564
Not Equal variances assumed			-3,233	782	,001	-9,208	2,848	-14,799	-3,617

Independent Samples Test Month 11 and 12

Duration	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	Df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
Equal variances assumed	3,506	,061	,257	1.091	,797	,851	3,314	-5,651	7,354
Not equal variances assumed			,285	889	,775	,851	2,982	-5,002	6,705

Figure 29 - Duration across Month - Mean Difference Test – Closing Cycles

II.3 - Statistical Tests for Variable: Duration across Cycle Type

Results:

Applying Kruskal Wallis Tests it can not be accepted that the distribution of Duration is the same across the two cycle types, at a 0,05 significance level.

Applying T-Test for 2 independent samples it can be said that at a 95% confidence Duration Mean:

- **Is** diferente from Opening to Closing cycle movements.

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of Duration is the same across categories of Cycle.	Independent-Samples Mann-Whitney U Test	,000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is ,05.

Figure 30 - Statistical Test for Duration is the same across Cycle type

Group Statistics

	Cycle	N	Mean	Std. Deviation	Std. Error Mean
Duration	Op	2.316	293,78	58,576	1,217
	Clos	2.274	276,43	46,333	,972

Independent Samples Test Cycle Opening and Closing

Duration	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	Df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
Equal variances assumed	1,095	,295	11,111	4.588	,000	17,341	1,561	14,281	20,401
Not equal variances assumed			11,135	4.389	,000	17,341	1,557	14,288	20,394

Figure 31 - Duration across Cycle Type - Mean Difference Test

Annex III – Transformed Variables – Statistical Tests

III.1 - Descriptive Statistics and Tests for Transformed Variables B1 to B5 Cycle: Opening

		Statistics				
		B1	B2	B3	B4	B5
N	Valid	2.316	2.316	2.316	2.316	2.316
	Missing	0	0	0	0	0
Mean		98,03	93,28	126,65	153,77	156,77
Std. Deviation		27,380	33,423	34,516	36,391	36,933
Variance		749,644	1.117,067	1.191,374	1.324,318	1.364,083
Minimum		1	-62	-82	-140	-199
Maximum		949	1.135	1.111	1.131	1.114
Percentiles	25	91,00	85,00	117,00	147,00	151,00
	50	96,00	92,00	124,00	153,00	157,00
	75	101,00	97,00	134,00	160,00	163,00

Table 15 - Transformed Variables – Opening - Descriptive Statistics

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of B1 is normal with mean 98,028 and standard deviation 27,38.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
2	The distribution of B2 is normal with mean 93,277 and standard deviation 33,42.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
3	The distribution of B3 is normal with mean 126,646 and standard deviation 34,52.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
4	The distribution of B4 is normal with mean 153,767 and standard deviation 36,39.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
5	The distribution of B5 is normal with mean 156,774 and standard deviation 36,93.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is ,05.

Figure 32 - Transformed Variables Distribution - Statistical Tests

III.2 - Descriptive Statistics and Tests for Transformed Variables B1 to B5 Cycle: Closing

	N	Minimum	Maximum	Mean	Std. Deviation
B1	2.274	-931	0	-53,33	22,217
B2	2.274	-939	158	-58,44	28,314
B3	2.274	-941	326	-70,39	32,287
B4	2.274	-941	0	-128,63	26,927
B5	2.274	-887	0	-145,13	26,091
Valid N (listwise)	2.274				

Table 16 - Transformed Variables – Closing - Descriptive Statistics

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of B1 is normal with mean -53,331 and standard deviation 22,22.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
2	The distribution of B2 is normal with mean -58,439 and standard deviation 28,31.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
3	The distribution of B3 is normal with mean -70,388 and standard deviation 32,29.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
4	The distribution of B4 is normal with mean -128,631 and standard deviation 26,93.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.
5	The distribution of B5 is normal with mean -145,131 and standard deviation 26,09.	One-Sample Kolmogorov-Smirnov Test	,000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is ,05.

Figure 33 - Transformed Variables Distribution - Statistical Tests

Annex IV – Maintenance Reports

3.1 Opening and closing cycles - Door A

Since the end of October some warnings concerning cycles of the door A have been detected that remained during the last two weeks. Such information can be clearly detected by analyzing graph from figure 5 and figure 6.

This situation is reported on the information that has been provided from Northern Rail.

Set	Veh	Date	Fault Code	Defect	Repair
156438	52438	16/12/2012	DOOR OPERATED - POWER	Repair sheet... ...'A' door butterfly very stiff [pdd]	lubricated and tested DW
156438	57438	01/12/2012	DOOR OPERATED - POWER	A door reported sticking MMI ORF9015 required	mmi done. door rollers renewed. lb

Figure 34 - Maintenance Report December 1 and 16th

2.1.2 Opening time dispersion of door A

The opening operations present a greater dispersion having a larger average opening time particularly on Days 4 and 5 of December. High value from opening time, some of them probably correspond to manual cycles, but we can clearly identify a tendency to increase the opening time, see figure 6, and therefore an increase number of critical events over the last days.

After the workshop interventions, the system shows as a better behavior. This means that the CBM system is catching these behaviors. With the door pressure signatures the door condition monitoring will be more accurate.



figure 5.

Figure 35 - Maintenance Report December 6th