Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Fault Detection and Explanation through Big Data Analysis on Sensor Streams

Giuseppe Manco, Ettore Ritacco, Pasquale Rullo, Lorenzo Gallucci, Dianne Kimber, Marco Antonelli

RT-ICAR-CS-16-05 **Ottobre 2016**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 7-11C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.icar.cnr.it*
– Sezione di Palermo, Via Ugo La Malfa, 153, 90146 Palermo, URL: *www.icar.cnr.it*

Fault Detection and Explanation through Big Data Analysis on Sensor Streams

Giuseppe Manco[a,*], Ettore Ritacco[a], Pasquale Rullo[e], Lorenzo Gallucci[d], Dianne Kimber[c], Marco Antonelli[b]

[a]ICAR-CNR, Via Bucci 41c, 87036 Rende (CS), Italy
[b]Bombardier Transportation S.p.A. Vado Ligure, Italy
[c]Bombardier Transportation ltd, London, UK
[d]Exeura S.r.l., Via P.A. Cabrai, 87036 Rende (CS), Italy
[e]Dip. di Matematica e Informatica, Università della Calabria, Via Bucci 30b, 87036 Rende (CS), Italy

Abstract

Fault prediction has become an important topic for the recent years as, by providing effective methods for predictive maintenance, allows companies to perform important time and cost savings. In this paper we describe an application developed to predict and explain door failures on metro trains. To this end, the aim was twofold: first, devising supervised techniques capable of early detecting door failures from diagnotic data; second, describing failures in terms of properties distinguishing them from normal behaviour. Data pre-processing was a complex task aimed at overcoming a number of issues with the dataset, like size, sparsity, bias, burst effect and trust. Since failure premonitory signals did not share common patterns, but were only

[*] Corresponding author

Email addresses: giuseppe.manco@icar.cnr.it (Giuseppe Manco), ettore.ritacco@icar.cnr.it (Ettore Ritacco), rullo@mat.unical.it (Pasquale Rullo), lorenzo.gallucci@exeura.eu (Lorenzo Gallucci), dianne.kimber@rail.bombardier.com (Dianne Kimber), marco.antonelli@rail.bombardier.com (Marco Antonelli)

characterised as non-normal device signals, fault prediction was performed by using outlier detection. Fault explanation was achieved by exhibiting device features showing abnormal values. An experimental evaluation was performed to assess the quality of the proposed approach. Results show that high-degree outliers are effective indicators of incipient failures. Also, explanation in terms of abnormal feature values (responsible for outlierness) seems to be quite expressive.

Keywords: fault detection, anomaly detection, outlier explanation, big data, sensor data

1. Introduction

The capability of preventing faults is one of the main issues in nowadays industry. The failure of a component, indeed, may cause the stoppage of an industrial system, with very negative consequences on both the productive cycle and the integrity of the plant itself.

Fault prediction is a recent area of research aimed at providing techniques for forecasting failures based on the observation of sensor signals during the normal working cycle of an industrial device. It represents the premise for predictive maintenance, i.e., the task of preventing potential problems by timely repairing or replacing the possible sources of failure before they actually happen. Preventive maintenance is opposed to corrective maintenance.

Fault prediction systems can be either based on a deductive, top-down approach, or on an inductive, bottom-up approach. In the former case domain experts build a mathematical model based on their knowledge about physical processes. This approach is limited in practice by the high complexity of real systems. In the latter case, a model is learned from past empirical observations. This approach relies on data mining techniques and requires the capability of dealing with the huge amounts of data that are usually sent by system monitoring sensors.

The Cobalt project, financed by Bombardier Transportation (one of the world's largest rail-equipment manufacturing companies) and carried out by Exeura, deals with the inductive approach, with reference to a particularly challenging application scenario: train predictive maintenance.

Trains are highly complex plants, comprising several electronic controlled systems, whose growing complexity has sensibly increased maintenance issues as well the potential opportunity of faults discovery and their prevention. In this context, door failures on metro trains (the SSR fleet in UK) plays a crucial role for Bombardier Transportation, because these failures result is several operational inefficiencies, such as delays or trip cancelations. A successful predictive maintenance process for train doors can help preventing failures, thus improving the operational performances of the fleets by avoiding delays and cancellations, and rewarding the investment needed for the process implementation.

Therefore, the objective of the project was twofold:

- devising supervised techniques which, in the context of an event-based monitoring of complex systems, are capable of early detecting door failures in the near future
- describing failures in terms of justification: that is, discovering the properties distinguishing failures from the normal behavior.

Achieving the above goals is a very challenging task, essentially because of the overwhelming amount of often misleading diagnostic data. There are two main issues associated with these data:

- Diagnostic messages come at very high frequency compared to their supposed mission (fault detection): for example, every minutes or even seconds. Many of these messages are also simply informative of the status of the overall system or of a specific components and the result is an overwhelming amount of data to be processed and filtered.
- Even more importantly, several diagnostic messages can have a low discriminative capability, or they can even be misleading. Diagnostic messages are generated automatically and they adhere to a certain internal programming logic according to a top-down approach. Within this logic, faults are hypothesized according to a given (static) behavior. However, the reality is different and the programming logic may be not faithfully representing the operating behavior of a component. For example, if a given component is manually switched off, a sensor trying to communicate with this component will continuously issue diagnostic messages. However, that does not necessarily result in a system failure.

That is why, when designing a predictive system, there is the need to build a reliable methodology capable of discriminating a potentially faulty status from normal operational working statuses.

The paper is organized as follows. In Section 2 we describe the application domain of door failures on metro trains. In particular, we give an overview of the traditional predictive maintenance process, as well as of the data sources recording historical diagnostic data. In Section 3 we provide the methodology that has inspired our work. First, data abstraction and problem formulation are given. Then, data pre-processing is described. Finally, both fault prediction and fault explanation techniques are provided. In Section 4 the experimental evaluation work is presented. In Section 5 a description of the architecture supporting the described methodology that was set up for the purposes of the project is given. Finally, in Section 7 we draw our conclusions.

2. Scenario: Door Train Failures

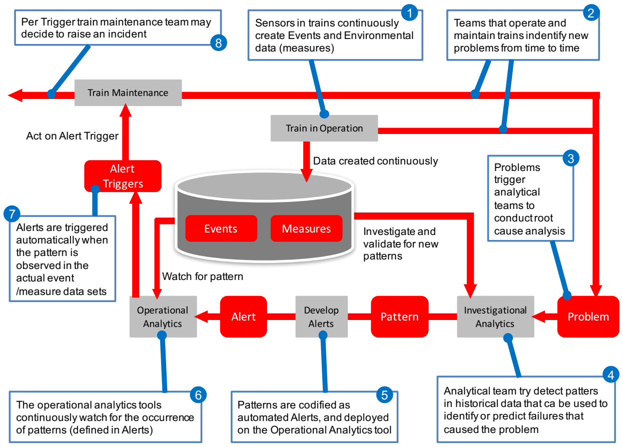
Per Trigger train maintenance team may decide to raise an incident
8
Sensors in trains continuously create Events and Environmental data (measures)
1
Teams that operate and maintain trains indentify new problems from time to time
2
Train Maintenance
Train in Operation
Act on Alert Trigger
Data created continuously
Alert Triggers
Problems trigger analytical teams to conduct root cause analysis
3
7
Alerts are triggered automatically when the pattern is observed in the actual event /measure data sets
Events
Measures
Investigate and validate for new patterns
Watch for pattern
Operational Analytics
Alert
Develop Alerts
Pattern
Investigational Analytics
Problem
The operational analytics tools continuously watch for the occurrence of patterns (defined in Alerts)
6
Patterns are codified as automated Alerts, and deployed on the Operational Analytics tool
5
Analytical team try detect patters in historical data that ca be used to identify or predict failures that caused the problem
4

Figure 1: Overview of the Overall Predictive Maintenance Process.

The focus of this experience is on door failures on metro trains.

2.1. The Predictive Maintenance Process

The traditional predictive maintenance process at Bombardier Transportation is depicted in Figure 1. Trains in operation continuously generate diagnostic messages that are composed by two types of data: *events* and *environmental measures* (sensor measures). Events are triggered by software components installed on the fleet, based on state information generated on the train. Enviromental measures are a set of data collected at the trigger of the software components in order to add a context to the associated events for their punctual interpretation. More precisely, each device (i.e., a door) is equipped with both a set S of sensors and a software component P. Sensors in S feed P with their data (such as ambient temperature, cab condition, battery voltage, derived passenger load and geographic location, etc.) and P, based on a certain programming logic, will eventually trigger one or more events with the associated environmental data.

Events are characterized by a number of features like *type*, *timestamp*, *subsystem* (identifying a specific door), *disturbance*, *duration*, *severity* (five severity levels, "critical" being the maximum), and *description* (free text). On the SSR projects there are 3,031 event types. Both events and related environmental data are stored into a data warehouse.

When teams that operate and maintain trains identify a new problem not triggered by alerts, the analytical team is called to conduct root cause analysis. To this end, it tries to detect patterns in the event/measures data warehouse that can be used to identify and explain failures that caused the current problem. Patterns are then codified as automated alerts, and deployed on the Operational Analytics tool. Alerts are triggered automatically when specific patterns are observed in the event/measures data set.

A crucial step in the above process is the *manual* pattern detection by the analytical team. This is indeed a complex and time consuming task that strongly relies on the experience of the domain engineers. The alerts that are defined are in general boolean combinations of a limited number of events, and their predicting capabilities are in general rather poor. To overcome this drawback, project Cobalt has been launched by Bombardier Transportation to explore how Big Data and Data Analytics techniques can be deployed to *automatically* detect patterns in historical data that can be used to identify, describe and predict failures.

2.2. Data Sources

There are three data sources identified for the application at hand. They record historical data concerning a one-year period of analysis:

- *Orbita*: It is the data set depicted in Figure 1, storing events and environmental measures about critical on-board systems. It comprises 4.5Gb of event data and 30Gb of environmental data for the period being analyzed.

- *Maximo*: used to track component location (Asset Configuration), details of work performed (Work Orders) and component usage data (Counters).

- *Intraxis*: used during the product introduction phase for the recording of failures with a financial impact (Service Affecting Failures - SAF's) and the results of root cause investigation.

As we will see in the next sections, these data collections have been used to provide the raw material for creating a training set for the induction of a fault prediction model. In particular, the Orbita collection was used to obtain the diagnostic data, Intraxis was used to identify the failures with a direct financial impact on the business (which are those of real interest), Maximo was used for filtering out from Orbita false positive events generated during on-depot tests.

2.3. Data Issues

Diagnostic data in Orbita exhibited a number of potential issues that might affect the learning process, including:

- *Size.* The number of events is very high (see Table 1), as sensors may trigger lots of events in a very short time frame of observation.

- *Sparsity.* Since events are associated with subsystems (i.e., devices), not all environmental measures can be captured for all events. The corresponding Event/Sensor matrix is extremely sparse.

- *Burst Effect.* In some scenarios, some devices may trigger a large number of events at a high rate. These emissions, while not necessarily related to failures, strongly unbalance the data distribution.

- *Bias.* Events are only issued when anomalous situations are detected on the corresponding subsystems. In this respect, environmental data is biased, as there is no information on the values of the sensors within the state of normal activities.

- *Trust.* Events are triggered when a sensor detects a value beyond some threshold, that is chosen by the sensor vendor according to some logic that can be different from the actual use of the device.

The problem with Maximo (which provides information about train maintenance activities in the workshop) was essentially related to the low reliability of timestamps (e.g., ”end date” equal to ”start date”), so that the Maximo link to Orbita proved to be error-prone. On the contrary, the Intraxis data set (which keeps track of failures with financial impact) showed to be a high-quality, reliable data source.

The quantitative relationship between diagnostic events and failures is shown in Table 1. Here, the data volumes relative to diagnostic events, as well as the actual service affecting failures (SAF incidents), are reported. We can see that even critical events (i.e., with high severity) are extremely frequent when compared to actual failures. As we will see, this raises a problem of *class imbalance*.

Span	**All events**	**Critical events**	**SAF Incidents**
All systems	18,103,714	183,327	4,616
Specific to doors	8,072,327	27,318	84

Table 1: One year data volumes

3. Methodology

3.1. Data abstraction

The description of the data sources provided in the previous section allows us to define a formal data abstraction. There three basic data types: events, environmental measures and failures.

Events. Events are characterized by a number of features like *type*, *timestamp*, *subsystem.* etc. - see Subsection 2.1. Let us denote by $\mathcal{E}$ the table of all events triggered by the system. Since event data is temporal, $\mathcal{E}$ can be regarded as a stream of events. Given the set D of subsystems (or devices - recall that in our application scenario, devices are train doors) and a specific device $d_i \in D$, we denote by $\mathcal{E}^{(i)} = \{e_1^{(i)}, \ldots, e_{m_i}^{(i)}\}$ the stream of events issued by d_i (i.e., events in $\mathcal{E}$ where $subsystem = d_i$).

Environmental measures. Let $\mathcal{S}^{(i)} = \{s_1^{(i)}, \cdots, s_{k_i}^{(i)}\}$ be the set of sensors associated with $d_i \in D$. Then each event $e_k^{(i)} \in \mathcal{E}^{(i)}$ issued by d_i is equipped with the values generated by (a subset of) the sensors in $\mathcal{S}^{(i)}$ - the environmental measures of $e_k^{(i)}$. We denote by $\langle v_h^{(i)}, t_k \rangle$ the value of the h-th sensor $s_h^{(i)} \in \mathcal{S}^{(i)}$ at time t_k, where t_k is the timestamp of event $e_k^{(i)} \in \mathcal{E}^{(i)}$. Then, we define

1. the time series $\mathcal{V}_h^{(i)} = \{\langle v_h^{(i)}, t_1 \rangle, \ldots \langle v_h^{(i)}, t_{m_i} \rangle\}$ of all values of sensor $s_h^{(i)} \in \mathcal{S}^{(i)}$ associated with the events $\{e_1^{(i)}, \ldots, e_{m_i}^{(i)}\}$ issued by d_i (stream of sensor $s_h^{(i)}$);
2. the set $\mathcal{V}^{(i)} = \{\mathcal{V}_h^{(i)} \mid h : s_h^{(i)} \in \mathcal{S}^{(i)}\}$ of all sensor streams associated with device d_i;
3. the set $\mathcal{V} = \{\mathcal{V}_h^{(i)} \mid i : d_i \in D \ \wedge \ h : s_h^{(i)} \in \mathcal{S}^{(i)}\}$ of all sensor streams.

Failures. Failures are pairs $\langle d_i, t \rangle$ expressing that a fault of device d_i has occurred at time t. We denote by $\mathcal{F}$ the set of all failures occurred in the given observation time.

All together, the above data $\mathcal{E}$, $\mathcal{V}$ and $\mathcal{F}$ provide a complete picture of the history of the entire system. In particular, the pair $\langle \mathcal{E}, \mathcal{V} \rangle$ can be interpreted as a time series whose elements are tuples describing the status of a specific device at a given time. Thus, in the overall, $\langle \mathcal{E}, \mathcal{V} \rangle$ can be regarded as a snapshot tracking the evolving status and behavior of each device over time.

3.2. Approach and Problem Statement

The focus of the approach is the device d_i, for which we would like to devise a fault prediction methodology. To this end, we start from the observation that the possibility for a fault to occur depends on the status of the device, and that the status is progressively tracked only through events triggered by the system. As a consequence, the detection problem has to be formulated relative to a current status. That is, given d_i and a timestamp t, our objective is to predict the time to failure $t + \delta$, given the snapshot $\langle \mathcal{E}_W^{(i)}, \mathcal{V}_W^{(i)} \rangle$, where $\mathcal{E}_W^{(i)}$ and $\mathcal{V}_W^{(i)}$) represent the events triggered by d_i and the relative environmental measures within the time interval W.

The situation is illustrated in Figure 2, depicting a series of events happening during any given time period. Faults occur somewhere in the timeline. An *analysis frame* W is set along the timeline, and the time to the closest failure is measured relative to the last event occurring within W (event x in the figure). An analysis frame is meant to collect information about the status of the device. Thus, our objective becomes to predict the time to failure relative to a target event x, given its timestamp and a backward time-frame (the analysis frame) associated with that timestamp.

There are some issues associated with this approach, which need to be tackled. First of all, the analysis frame associated with the target event should be properly sized. A possible approach can be to consider the analysis frame comprising all events since the last failure. This approach has the drawback that some events correlated with future failures are lost. A better approach is one where the analysis frame is time limited, rather than bound by a precursor failure. Here, if the time period being considered is

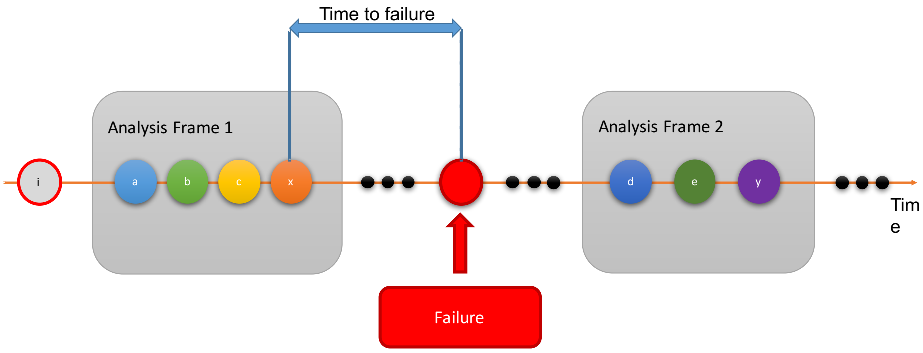

Figure 2: Methodology of the approach.

made long enough then all relevant events will be included in the process of model development. Example strategies could perform the prediction on a weekly or monthly basis (depending on the requirements of the predictive maintenance).

Second, the target time upon which the prediction is made needs to be fixed. Again, several different choices can be made. A sliding window approach would force the prediction on every possible event on the timeline. This is shown in Figure 3. However, a problem we encountered with this approach is the strong correlation between consecutive events. In practice, the prediction on event c is correlated with the prediction on event x, and as a consequence the resulting predictor has an extremely poor generalization capability. We call this issue the *data diversity* problem, to denote that several predictions do not rely on diverse faults. To relieve this problem, a better strategy is one providing for fixed consecutive *milestones* (i.e., timestamps) along the timeline.

Thus, fixed milestones, each associated with a time limited analysis frame, seems to be in the overall the more advisable strategy.

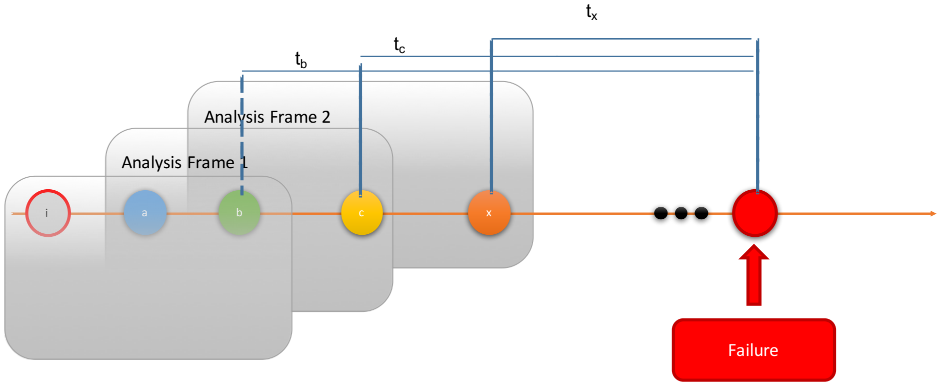

Figure 3: Sliding-window prediction.

However, the data diversity issue is not yet completely defeated, unless we abandon the time-to-failure predictive approach: predicting the time to failure, indeed, unavoidably triggers correlations in subsequent milestones. To overcome this drawback, we then transform the time-to-failure problem into the following binary problem: *Given a milestone, how likely is it that a fault will occur in a given observation window?*

This is illustrated in Figure 4. Here, each milestone is associated with both an analysis frame and an *observation window*. If a failure occurs within this window, then the analysis frame is labeled as positive, otherwise it is labeled as negative. Within the figure, the frame ending with event x is positive, whereas the frame ending with event y is negative.

More formally, we define a milestone as a triple $\langle t, W, O \rangle$, where t is a timestamp, W is the analysis frame and O the observation window. Further, let $\langle \mathcal{E}_W^{(i)}, \mathcal{V}_W^{(i)} \rangle$ be the snapshot of d_i within the analysis frame W, i.e., $\mathcal{E}_W^{(i)} = \{e | e.subsystem = d_i, e.timestamp \in W\}$ and $\mathcal{V}_W^{(i)}$ represents all time series associated with sensors relative to events in $\mathcal{E}_W^{(i)}$.

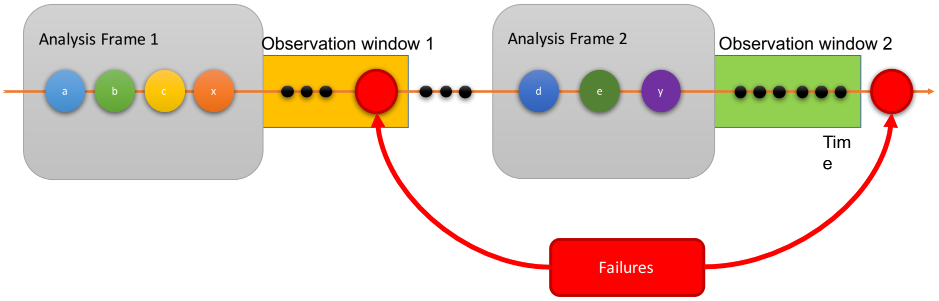

Figure 4: Fixed-observation window prediction.

Detection: Given a milestone $\langle t, W, O\rangle$, can we predict whether a failure will occur in O based on the statistical properties of $\langle \mathcal{E}_W^{(i)}, \mathcal{V}_W^{(i)}\rangle$?

Explanation: Given a failure $\langle d_i, t_k\rangle$ and a milestone $\langle t, W, O\rangle$ such that $t_k \in O$, is it possible to devise a pattern characterizing $\langle \mathcal{E}_W^{(i)}, \mathcal{V}_W^{(i)}\rangle$? Here, a "characterizing pattern" is any set of features that substantially differentiate the elements of $\langle \mathcal{E}_W^{(i)}, \mathcal{V}_W^{(i)}\rangle$ from those of $\langle \mathcal{E}, \mathcal{V}\rangle$.

3.3. Data Manipulation

The above problem formulation allows us to shift the focus from events to milestones, thus enabling to aggregate the data into a more compact format. Given a device d_i and a milestone $\langle t, W, O\rangle$, the basic idea is that both event and sensor data in $\langle \mathcal{E}_W^{(i)}, \mathcal{V}_W^{(i)}\rangle$ can be aggregated into a single tuple summarizing the evolving status and behavior of d_i during the analysis frame W. This is done by devising a number of statistics meant to characterize the evolution of the series $\langle \mathcal{E}_W^{(i)}, \mathcal{V}_W^{(i)}\rangle$ in terms of simple patterns.

Table 2 summarizes the descriptive statistics relative to $\mathcal{E}_W^{(i)}$. To this end, we notice that $\mathcal{E}_W^{(i)}$ can be interpreted as a set of time series as well, relative to discrete values (event types) and continuous values (event duration). Each statistic is computed in four different ways:

- By considering all events within W
- By assuming that W can be partitioned into m fractions (for example, if W is relative to a week then we can partition it into days of the week); Then, for each $f \in \{1, \ldots, m\}$ we consider the statistic relative to the fraction f. For example, $cnt(f)$ represents the number of events within the fraction f.
- By grouping events according to event type. For example, $avg(T)$ represents the average duration of events of type T.
- By grouping events according to type and fraction. For example, $rad(f, T)$ represents the relative average deviation between the events of type T within fraction f.

Besides the standard concentration and dispersion measures, the *relative average deviation* is worth further comments. This statistic measures deviations from a static trend, which can be symptoms of a malfunction. The typical situation is when the number of events increases progressively. Capturing these deviations and measuring a degree of correlation with the timeline turns out to be a crucial predictor for a forthcoming failure.

Similarly, table 3 summarizes the descriptive statistics associated with $\mathcal{V}_{h,w}^{(i)}$, for each sensor s_h associated with d_i. Statistics such as kurtosis, interquantile range and number of outliers were computed to indicate whether the values distribute evenly or exhibit peaks (which, again, could denote malfunctions and hence forthcoming failures).

Feature	*Description*
cnt	the number of events occurred during W.
$rad(f)$	Relative Average Deviation $\frac{cnt(f+1)-cnt(f))}{cnt(f)}$: the relative increment of occurrences of events from the time fraction f to the next one.
rad	the arithmetic mean of $rad(f)$ over all the fs.
rad_corr	the correlation between the periodic $rad(f)$ and the timeline.
h	the harmonic mean of the duration of all the events during W.
avg	the arithmetic mean of the duration of events during W.
med	median of the duration of all the events during W.
var	variance of the duration of events during W.
k	Kurtosis Skewness Index of the duration of events during W.
IQR	Interquartile range ($Q3-Q1$) of the duration of events during W.
e_outl	number of anomalous durations (durations below $Q1-1.5IQR$ or above $Q3+1.5IQR$) for all the events during W.

Table 2: Event features.

It turns out that the dataset resulting from the above data manipulations is a table $\mathcal{T}$ where each row represents the status of a given device in the analysis frame of a given milestone. In other words, a row $\langle d_i, m\rangle$ of $\mathcal{T}$ is a summarization of the snapshot $\langle \mathcal{E}_W^{(i)}, \mathcal{V}_W^{(i)}\rangle$, where W is the analysis frame of the milestone m, through the above described statistics. The number of such rows depends on the choice of the analysis frame (e.g., one week). The data set $\mathcal{T}$ is characterized by a number of attributes essentially depending on the number of sensors (for which statistics are computed) associated with the

Feature	*Description*
$corr(s_h)$	correlation coefficient of the values of s_h with respect to the timeline
$e\_h(s_h)$	harmonic mean of the values of s during W
$e\_avg(s_h)$	arithmetic mean of the values of s_h during W
$e\_med(s)$	median of the values of s_h during W
$e\_var(s_h)$	variance of the values of s_h during W
$e\_k(s_h)$	Kurtosis Skewness Index of the values of s_h during W
$e\_IQR(s_h)$	Interquartile range of the values of s_h during W
$e\_outl(s_h)$	number of outliers (values below $Q1-1.5IQR$ or above $Q3+1.5IQR$) observed by s_h during W

Table 3: Environment features.

different devices. There is an additional binary attribute (the class attribute) denoting whether the given device experienced a failure in the observation window of the milestone (this is done by exploiting the set $\mathcal{F}$ of all failures occurred in the given observation time).

We conclude this section by remarking that this condensed representation helps to mitigate some of the mentioned issues concerning data (see Section 2). First of all, the *size* of the data set $\mathcal{T}$ is dramatically reduced. In addition, by providing a wider and more general view on the data, where the behavior of sensors is summarized relative to analysis frames, both *bias* and *sparsity* issues are alleviated as well.

3.4. Fault Detection

A failure is an abnormal status of a device. There can be several possible causes for such an unexpected behavior and the standard approach in trying to detect failures is trying to detect potential patterns which recur among them. In practice, the prediction is accomplished through a classifier capable of detecting patterns which characterize failures.

However, the situation we face in our scenario is characterized by an extreme class inbalance. In such a context, traditional classification techniques perform poorly: failures are indeed characterized by different pattern, so that typically no failure shares common causes with other failures. This factor hampers the discovery of global patterns through failures, impeding the ground truth retrieval needed by classification/regression models.

More suitable in our context is the adoption of an outlier detection (Chandola et al., 2009; Aggarwal, 2013) framework exploiting the manipulated data defined in Section 3.3. The main idea is to devise a ranking of rows according to their anomaly degree, so that rows with higher rank represent the most probable and imminent system faults. The ranking is essentially based on observing how different a certain pair $\langle \textit{device}, \textit{milestone} \rangle$ is from all the other pairs in the data set $\mathcal{T}$ - that is to say, how anomalous is the behavior of the given device in the associated analysis frame. Hopefully, this altered behavior correlates with failures, thus allowing to achieve high accuracy.

Thus, given a tuple $\mathbf{x} = \langle \textit{device}, \textit{milestone} \rangle$ of $\mathcal{T}$, representing the status of the given device in a given time frame, we define

$$\textit{rank}(\mathbf{x}) = 1 - p(\mathbf{x})$$

where $p(\mathbf{x})$ represents the probability of $\mathbf{x}$ within the given domain. Higher probabilities correspond to frequent working statuses (i.e. regular activities); by contrast, low probabilities represent exceptional situations, which are hence more likely to correlate with anomalous behavior and consequently imminent failures.

Thus, the core of our approach is a methodology for estimating $p(\mathbf{x})$. We approach it in two steps.

3.4.1. Mixture Modeling

In our setting, we hypothesize $\mathbf{x}$ as coming from one of K possible different normal (hence expected) working behaviors. That is to say, we can model $p(\mathbf{x})$ as a mixture

$$p(\mathbf{x}) = \sum_{k=1}^{K} p_k(\mathbf{x})\pi_k \tag{1}$$

where $p_k(\mathbf{x})$ represents the probability of observing $\mathbf{x}$ as complying to the k-th normal behavior, and π_k is the prior probability of observing a device associated with behavior k. In this framework, failures are outlier rows with low degree of aggregation with these K components. The parameter inference of Equation 1 can be performed via traditional mixture model estimations such as an *Expectation Maximization* (EM) approach. We model $p(\mathbf{x})$ as a parametric function $p(\mathbf{x}|\Theta)$, where Θ is the set of all parameters which characterize the components. Let $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ represent the training data of observed devices. Then the likelihood of the data, given the model parameters Θ, can be expressed as:

$$\mathcal{L}(\Theta; D) = \prod_i p(\mathbf{x}_i|\Theta)$$

The corresponding learning problem is finding the optimal $\hat{\Theta}$ that maximizes $\mathcal{L}(\Theta; D)$. Following the standard mixture modeling approach (Dempster et al., 1977), we can rewrite the likelihood, by exploiting Equation 1, as follows:

$$\mathcal{L}(\Theta; D) = \prod_i \sum_{k=1}^{K} p_k(\mathbf{x}_i|\theta_k)\pi_k$$

which can be optimized by resorting to the traditional EM algorithm, introducing a hidden binary matrix Z, where $z_{i,k}$ denotes the membership of the i-th flat row to the k-th mixture component, with the constraint $\sum_{k=1}^{K} z_{i,k} = 1$. Θ can be partitioned into $\{\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K\}$, where θ_k is the parameter set relative to the k-th component.

The *complete-data likelihood* of the model is:

$$p(D, Z, \Theta) = p(D|Z, \Theta) \cdot p(Z|\Theta) \cdot p(\Theta) \tag{2}$$

where

$$\begin{aligned} p(D|Z, \Theta) &= \prod_i \prod_{k=1}^{K} p_k(\mathbf{x}_i|\theta_k)^{z_{i,k}} \\ p_k(\mathbf{x}_i|\theta_k) &= \mathcal{N}(\mathbf{x}_i|\mu_k, \Sigma_k) \\ p(Z|\Theta) &= \prod_i \prod_{k=1}^{K} \pi_k^{z_{i,k}} \end{aligned}$$

Where $\mathcal{N}(\cdot)$ is the normal distribution parameterized by mean μ_k and covariance matrix Σ_k.[1]

$p(\Theta)$, in Equation 2, represents the prior relative to the parameter set Θ. Inspired by (Figueiredo and Jain, 2002), we choose to model the latter as:

$$p(\Theta) \propto \prod_{k=1}^{K} \pi_k^{-\frac{1}{2}\sqrt{|\theta_k|}},$$

with the interpretation that, for fixed K, the parameters π_k allow an "improper" Dirichlet-type prior. This enables a formulation of EM algorithm

[1] The gaussian model is not a limitation here, since the approach can be parameterized to any suitable distribution. However, we found that the normal distribution provides a reliable model even with those features representing counteers (it is worth reminding that a gaussian distribution suitably approximates a Poisson distribution for large counts).

which leads to the automatic detection of the optimal number K of mixture components. In fact, by standard manipulation of Equation 2, the *Complete-Data Expectation Likelihood* (Dempster et al., 1977) is given by:

$$\begin{aligned}\mathcal{Q}(\Theta;\Theta') &= E[\log P(D,Z,\Theta)|D;\Theta'] \\ &\propto \sum_i \sum_{k=1}^{K} \gamma_{i,k}\left\{\log p_k(\mathbf{x}|\theta_k) + \log \pi_k\right\} \\ &\quad -\sum_{k=1}^{K} \frac{N_k}{2} \log \pi_k\end{aligned}$$

where $N_k = \sqrt{|\theta_k|}$, and $\gamma_{i,k}$ represents the posterior probability of choosing component k, given $\mathbf{x}_i$. Optimizing $\mathcal{Q}(\Theta;\Theta')$ with respect to π_k under the constraints $\sum_k \pi_k = 1$ and $0 \leq \pi_k \leq 1$ yields:

$$\pi_k = \frac{\max\left\{0, \sum_i \gamma_{,k} - N_k/2\right\}}{\sum_{k=1}^{K} \max\left\{0, \sum_i \gamma_{i,k} - N_k/2\right\}}.$$

Here, the proposed prior admits an adjustment to the estimation of π_k which enables “annihilation”: a component not supported by a sufficient number of flat rows is removed. Thus, we can start with an arbitrary large initial number of mixture components, and then infer the final number K by letting some of the mixing probabilities π_k be zero.

The overall algorithm can be devised hence as an iterative two-step process, where in the E step we estimate γ given the current values of Θ,

$$\gamma_{i,k} \propto p_k(\mathbf{x}_i|\theta_k)\pi_k$$

and in the M step we exploit the γ values to estimate the π_k as above, and both μ_k and Σ_k as:

$$\mu_k = \sum_{i=1}^{N} \gamma_{i,k} \mathbf{x}_i$$

$$\sigma_k = \sum_{i=1}^{N} \gamma_{i,k} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T$$

3.4.2. Robust Outlier Detection

Applying the EM framework described in the previous section to the data resulting from the preprocessing described in 3.3 exposes to two potential issues, both related to the high dimensionality.

First, probability measures are not robust to high dimensionality. Tuples exhibiting a high number of features indeed express an extremely low density, which makes the precision in the computation of the rank problematic.

The second problem is the scalability of the inference process. In particular, the estimation of the covariance matrix is quadratic in the dimensionality of the data. This can make the process extremely slow.

The solution to both issues is based on a bagging approach, inspired by (Lazarevic and Kumar, 2005). In practice, rather than directly computing the rank on the whole set of features, we combine several rankings computed on small subsets of the features. This approach allows to better mitigate the effects of high dimensionality on the computation of the density of a tuple. In addition, each rank can be computed in parallel, thus substantially speeding up the inference phase.

Formally, let $x_1, \ldots, x_n$ be the features of tuple $\mathbf{x}$. Given $S \subset \{1, \ldots, n\}$, the tuple $\mathbf{x}_S$ represents the subset of $x_1, \ldots, x_n$ corresponding to the indices in

S. Let $A = \{A_1, \ldots, A_q\}$ be a set of q random samplings with replacement of $\{1, \ldots, n\}$ such that $|A_j| << n$ and A is complete, i.e., $\bigcup_{j=1}^{q} A_j = \{1, \ldots, n\}$.

Each A_j represents a subset of features upon which to build a ranker. Then, all q rankers contribute to the final ranking:

$$\begin{aligned} rank(\mathbf{x}) =& 1 - \frac{1}{q} \sum_{j=1}^{q} p(\mathbf{x}_{A_j}) \\ =& 1 - \frac{1}{q} \sum_{j=1}^{q} \sum_{k=1}^{K_j} p(\mathbf{x}_{A_j} | \theta_k^{(j)}) \pi_k^{(j)} \end{aligned} \tag{3}$$

Here, K_j represents the number of components detected for the mixture associated with the A_j subset, and similarly $\pi_k^{(j)}$ and $\theta_k^{(j)}$ represents the parameters associated with the k-th components of such a subset.

Thus, the choice to run the EM algorithm for a small subset of the set of features allows to gain numerical stability in the estimation of the probability distributions. Moreover, the process benefits from a statistically significant improvement in the prediction accuracy, due to the combination of multiple models. Finally, the computational cost can be dramatically reduced by applying parallelism combined to MapReduce techniques. Each mapper contains a ranker which is fed with a small projection of the whole data set, while the reducer implements the collaborative voting procedure.

3.5. Fault Explanation

One of the most desired features in a fault prevention system is the intelligibility of the predictive processes for impending failures. This request is supported by the need to define maintenance procedures that prevent the device breaking during its working time.

A predictive process of failure discovery is intelligible if it allows humans to understand how and why failure predictions are determined, i.e., it provides a description of the device alteration that will shortly bring to the occurrence of a fault. Such a description is a snapshot of the device status in which the features that exhibit abnormal behavior are highlighted and ranked, with respect to the average activity of the normal working operation. A description is then an ordered list of features which likely exhibit abnormal values and hence are symptoms of abnormal behavior. This means that a comparison of features is needed.

We rely on a two-step approach for building the rank. In the first step, we select those features which are likely to characterize the outlierness of an object deemed as an outlier by the process described in sec. 3.4. In a second step, we provide a score for each such features, and provide a rank of those features based on how untypical is their exhibited value.

In the first step, the objective is to find an explanatory subspace, that is a subspace of the original numerical attribute space where the outlier shows the greatest deviation from the other points. We based our approach by encoding the notion of outlierness as separability (Micenková et al., 2013): given an object $\mathbf{x}$ deemed as an outlier, one can devise an artificial set of points $\mathbf{y}$ oversampled from a gaussian distribution centered in $\mathbf{x}$. Then, the outlierness of $\mathbf{x}$ can be measured in terms of the accuracy in separating the artificial points $\mathbf{y}$ from the other points in D. Having encoded the outlierness as a classification problem, the explanatory subspace can hence be reduced to feature selection relative to such a classification problem.

Figure 5 depicts the situation, where we can see a sample of data (depicted

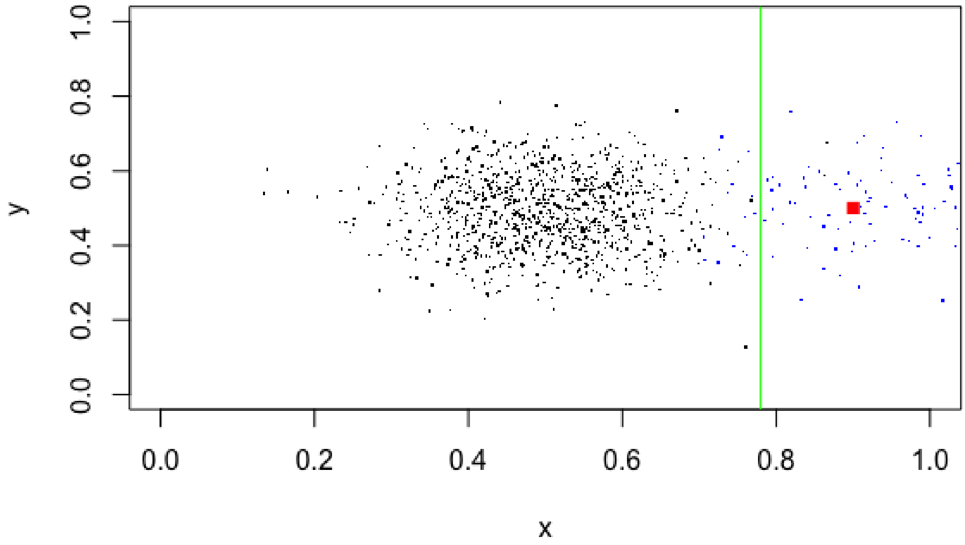

Figure 5: Outlierness as separation: attribute x separates the outlier values from all the others.

as black in the picture) spanning over two attributes x and y. Then, an outlier (depicted as red) is placed within the space. The blue points are objects that are randomly generated based on the features exhibited by the outlier value. We can see then that a separation line (colored green) can be drawn perpendicular to axis x, separating blue points from black points. By contrast, it is not possible to draw a similar separating line on the y axis. As a consequence, the outlier can only be explained based on feature x.

The selection of the most appropriate features allows us to detect which features characterize the outlierness. However, we also would like to rank these features, based on the degree of outlierness of the value they exhibit. Comparing semantically different features is difficult since each feature may have different values at different scales. To the purpose of comparing such features, it is mandatory to provide a normalization task where each feature is mapped into a shared numerical domain.

We approach this problem by providing a custom *z-score* normalization for each attribute. In section 3.4 we modeled our data as normally distributed. Given a set of normally distributed values $\{x_1, \ldots, x_n\}$, the peculiarities of the normal distribution allows us to characterize the population according to a tolerance interval, defined as:

$$\left(\mu - z^* \frac{\sigma}{\sqrt{n}}; \mu + z^* \frac{\sigma}{\sqrt{n}}\right)$$

where μ is the average of the samples, σ their standard deviation and z^* is a range parameter for the interval. For example, when the range includes 99% of the population, then $z^* = 2.576$.

The (quasi) normalization transformation consists in computing the *ti-difference* for each feature, defined as the absolute difference of the outlier value against the normal behavior weighted over the feature's tolerance interval in normal activity:

$$\textit{ti-difference} = |x - \hat{\mu}| \cdot \left(z^* \frac{\hat{\sigma}}{\sqrt{\hat{n}}} \right)^{-1} \tag{4}$$

where $\hat{\mu}$ is the feature average considering only non-outliers, x is the value of the feature for the outlier, $\hat{n}$ is the number of non-outlier tuples and $\hat{\sigma}$ is standard deviation of the attribute considering only non outliers. Here, z^* is chosen so that the population would include all tuples but one. In principle, since we are only focusing on a single outlier, we consider all values but one as "normal" (and hence within the tolerance interval).

Thus features are scored by mapping the values within the same tolerance interval, and observing their position with respect to such an interval. This allows us to provide a rank and ultimately to focus on the most deviating values.

4. Experimental Evaluation

The proposed approach was tested on a set of observations covering a time span of one year (from January to December 2015) and relative to door failures on metro trains (the SSR fleet in UK).

4.1. Data Manipulation

Starting from the data sources described in Section 2, we created a dataset according to the data manipulation procedure described in Section 3.3. To this end, we set both the analysis frame and the observation window equal to one week, so that the resulting table consisted of 2,958 rows (51 weeks for each of the 58 devices). Of such rows, 84 were labelled as "positive" (i.e., represented failures).

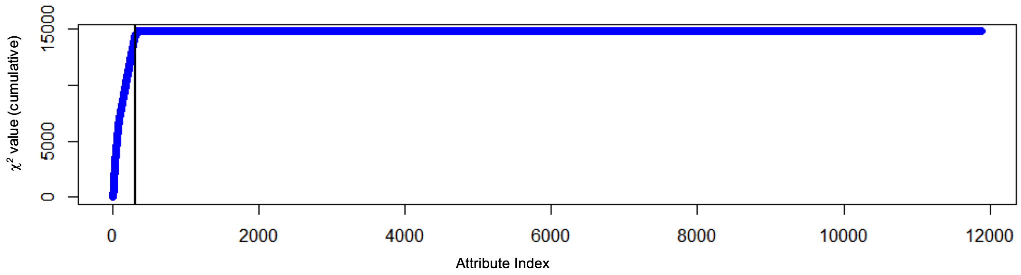

Figure 6: Ranking of attributes according to cumulative χ^2 values.

Attributes were initially 11,893, describing the statistics of Section 3.3 for the different sensors. However, the number of such attributes turned out to be extremely large, and so we had to resort to dimensionality reduction techniques in order to focus on the relevant attributes only. Attribute selection is a well-known problem in the scientific field of knowledge discovery, which is worsened in this case by the extreme class inbalance and the high dimensionality. Some of the most used techniques for choosing an optimal set of features are based on statistics like *Information Gain* (IG), *Odd Ratio* or Chi-squared (CHI) (Mladenic and Grobelnik, 1999; Zheng et al., 2004). For highly skewed data, the class distribution is biased toward the majority class, in the sense that most classifiers would predict the most frequent class to obtain overall accuracy. However, in dealing with highly skewed data, we are more interested in predicting the minor class as to achieve a low false-negative rate while maintaining overall accuracy. According to (Tang and Liu, 2005), when the data is skewed, IG and CHI choose more positive features, thus are biased toward the positive features. Since our purpose was to discriminate and characterize positive features, we chose to adopt CHI and to select the attributes which scored the highest. Figure 6 shows a ranking

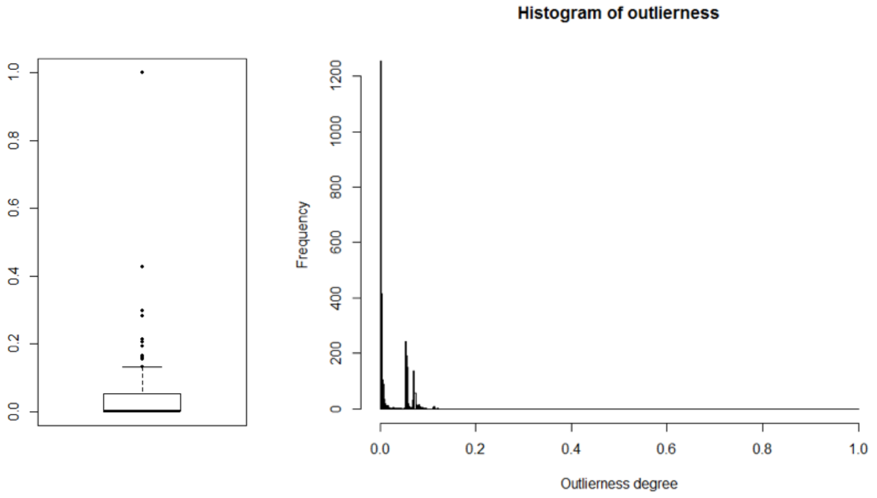

Figure 7: Distribution of ranks resulting from the outlier detection algorithm.

of the attributes according to cumulative χ^2 value. We can notice that, out of the 11,893 features, only 300 attributes exhibited a significant score.

The final dataset was therefore made of 2,958 tuples, each consisting of 300 features plus 1 class label. Of such tuples, 84 were labelled as positive examples, and the remaining as negative ones.

4.2. Outlier detection

Figure 7 shows the results of the outlier detection process. The values plotted in the graphs represent the distribution of the ranks. We can observe that the majority of the values range within the interval $[0, 0.15]$, and in general values greater than 0.18 can be considered exceptional w.r.t. the majority.

The evaluation of the results was performed by exploiting *Cumulative Gains Chart* (see Figure 8). The approach considers the rank associated with each tuple, and then evaluates whether thresholding such values allows to effectively detect actual failures. The Cumulative Gains Charts compare the percentage of the overall number of doors failures (Y axis) against the percentage of the total number of rows (X axis): a point $p = (x, y)$ in the chart means that, sorting the tuples from the largest to the smallest outlierness degree, within the set composed by the first x% rows there is the y% of the total failures. Within the graph, the green line shows the optimal performance, while the red line shows the random performance.

The first graph of Figure 8 shows the chart associated with the entire dataset. Here, the area under the ROC curve of 0.781. We can notice a steep initial behavior of the curve: a clear sign that all tuples with higher rank correspond to actual failures.

The other three graphs show a zoom at 5% of the dataset, and they focus on different threshold values. In particular, the second graph highlights that all first 16 top outlier-ranked rows correspond to failures. Also, according to the other graphs, the top 20 ranked tuples correspond to 19 actual failures, and the 24 top rows show only 2 false positives.

The above experiments assume that the ranking is accomplished by exploiting all 2,958 tuples for building the model. However, we can devise a different scenario, where the model is built periodically, and then it is exploited to compute the probabilities of incoming tuples and then to rank them according to the previous model.

A second set of experiments was then accomplished by randomly splitting the initial dataset with a proportion of 80% and 20%. The splits represent a training and a test set, respectively. In particular, the training set was

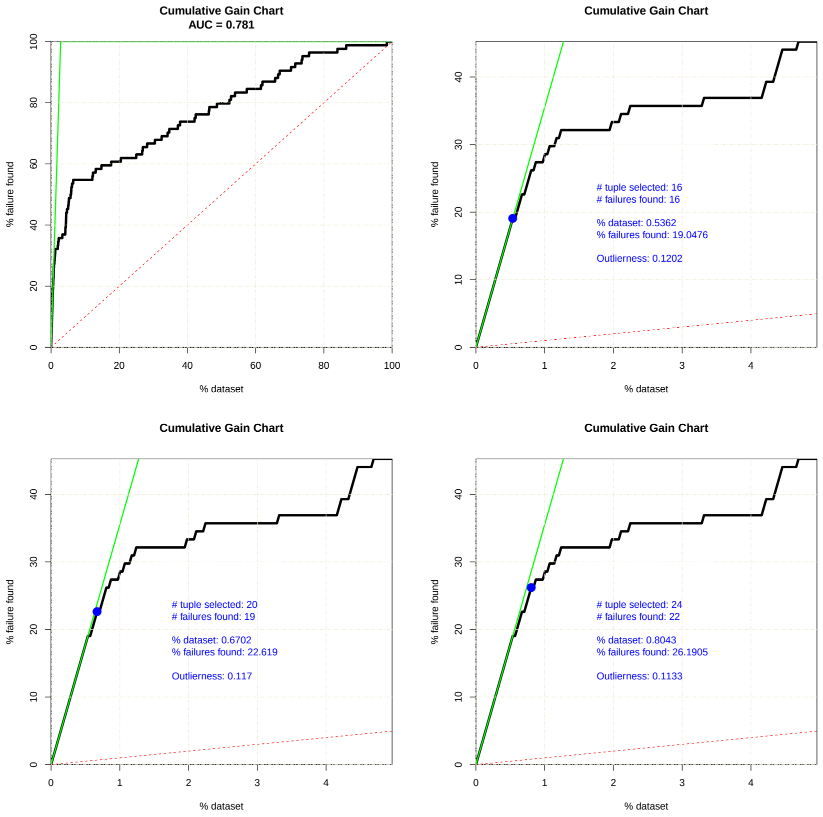

Figure 8: Cumulative gains chart (on the left side) and its zooms on the whole dataset

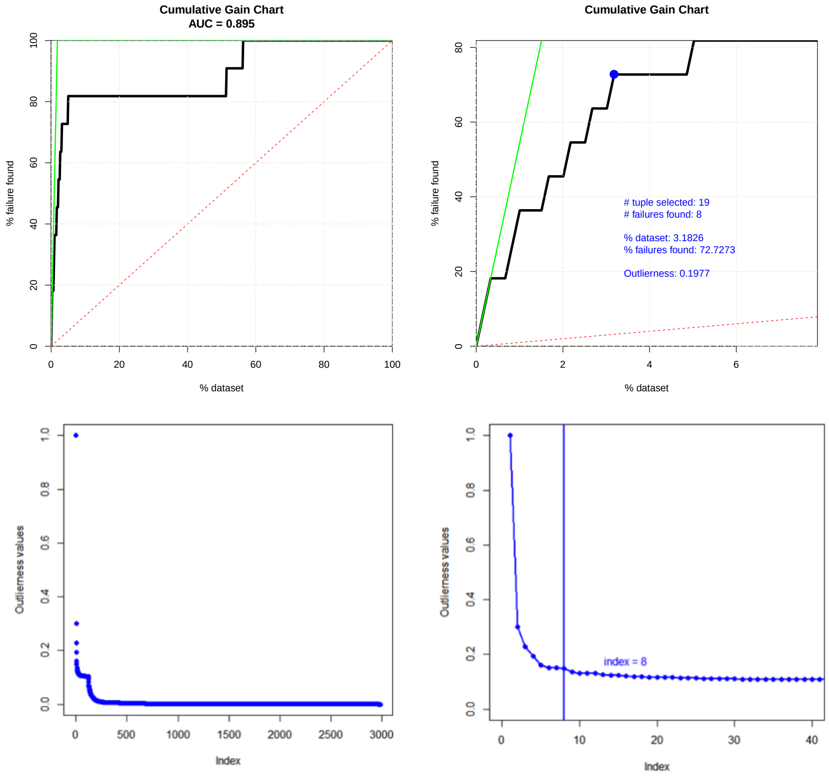

Figure 9: Cumulative gains charts and corresponding outlierness degrees (in descending order) on the test set

composed by 2387 tuples with 73 failures, while the test set had 571 rows and 11 failures. The results shown in Figure 9 exhibit an even higher AUC (0.895 in the picture) and, by selecting only the first 19 rows (out of 571 in the test set) we are able to detect 8 failures representing 73% of the total. The figure also shows a sorting of the outlierness values. In general, a cutting threshold strategy can be devised based on the study of such a curve. Peaks within such a curve highlight abnormal values. Thus, choosing a threshold on this curve based on the capabilities of actually monitoring can provide a suitable strategy. The last graph shows that a threshold focused on 8 tuples out of 571, allows us to detect 4 actual failures, on a total of 11.

4.3. Explanation

The explanations of the outlierness degrees were obtained according to the methodology proposed in Section 3.5: for each outlier, the *ci-distance* was computed and drawn. Figures 10 and 11 plot the values of *ti-difference* for the two top outliers (red line in the plots) and compares it to the values for the non-outlier tuples (grey thick line in the plot). From that figure one can notice that each outlier has several features exhibiting abnormal values (the most ten relevant features of the two outliers are shown as box plots in the figure). For the selected attributes, we can observe that the value exhibited by the outlier (depicted in red) departs significantly from the values of the normal tuples.

Thus, the general strategy for explaining the attributes can be devised as follows. For each outlier:

- Select relevant attributes by separation
- for each selected attribute compute the *ti-difference*
- plot values of *ti-difference* for the outlier tuple and for normal tuples
- select the attritues with most significant deviation in the *ti-difference* and inspect them.

The resulting outlier visualization technique looks extremely promising, since it allows for looking at once all unusual attribute behaviours and gathering,

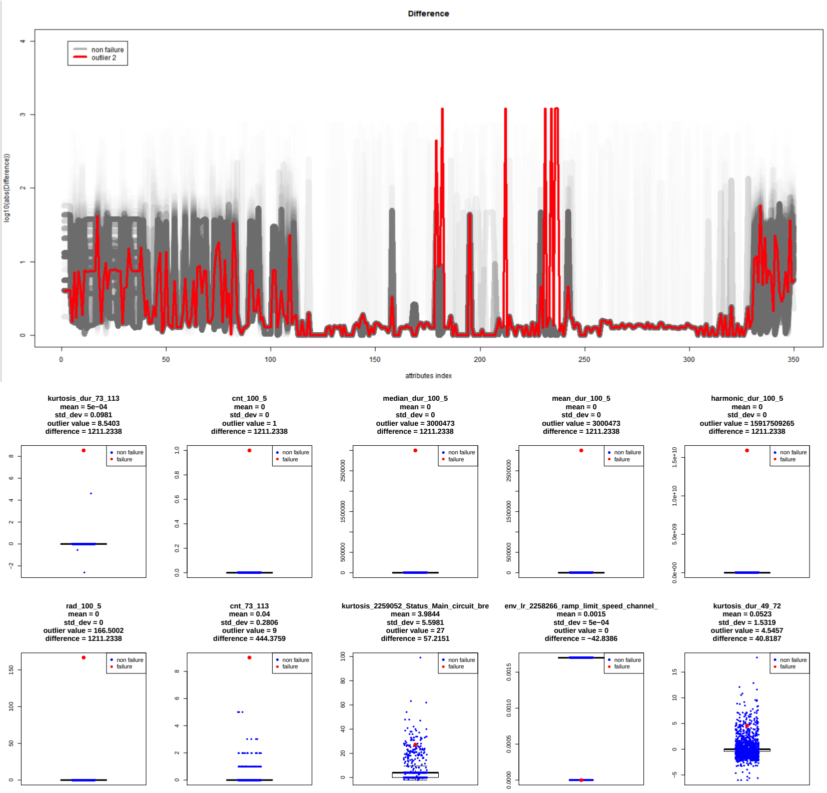

Figure 10: Plot of the *ti-difference* for the top outlier and boxplots of the outlying attributes according to *ti-difference*.

as such, a larger choice of candidate causes for outlierness, while still retaining focus on attributes having peculiar statistical behaviour.

5. System Architecture

The implementation of the methodology relies on an architecture for data manipulation, which was set up specifically for the purposes of the project. Figure 12 depicts the layers and tools of the architecture which are briefly described below:

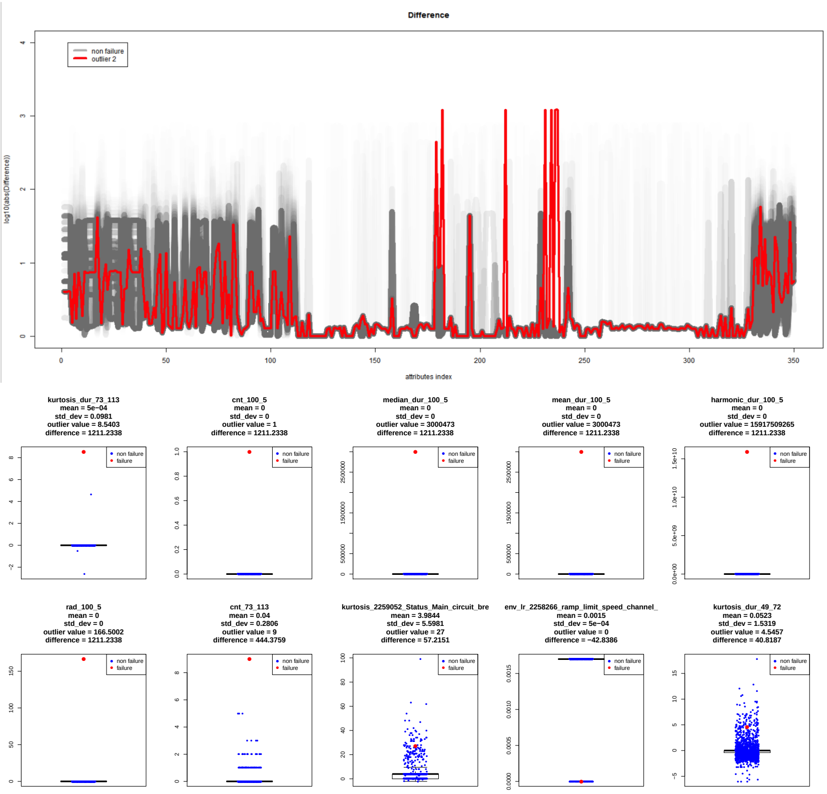

Figure 11: Plot of the *ti-difference* for the runner-up top outlier and boxplots of the outlying attributes according to *ti-difference*.

- The Apache Hadoop framework which comprises (1) the Hadoop Distributed File System for supporting high-throughput access to application data; (2) HBase and Hive, for efficient data storing, summarization and *ad hoc* querying. Within this layer, tables containing events and environmental measures data are stored using an Hadoop cluster running an HBase database accessed through the Hive relational interface.

- A data processing layer which exploits the Apache Pig dataflow framework for creating MapReduce programs used with Hadoop. Within this layer, data stored within the HBase are manipulated by means of Pig scripts in order to obtain a condensed (flattened) representation of the information concerning devices according to the aggregate features devised in sec. 3.3. The phases which involve this layer are the most time consuming in the whole process, due to high demand of computational resources. In this respect, a salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turn enables Pig to scale out to many machines and handle very large data sets in a reasonable execution time.

- An analytical layer which exploits the analytical tools R (a free software environment for statistical computing and graphics (R Core Team, 2014)) and Rialto (an extensible Business Analytics platform based on Machine Learning techniques for models induction (Manco et al., 2016)). Within this layer, the models are devised in two phases: first, by performing statistical analysis, data cleaning, manipulation and partitioning on the flattened table; second, by feeding cleansed partitions

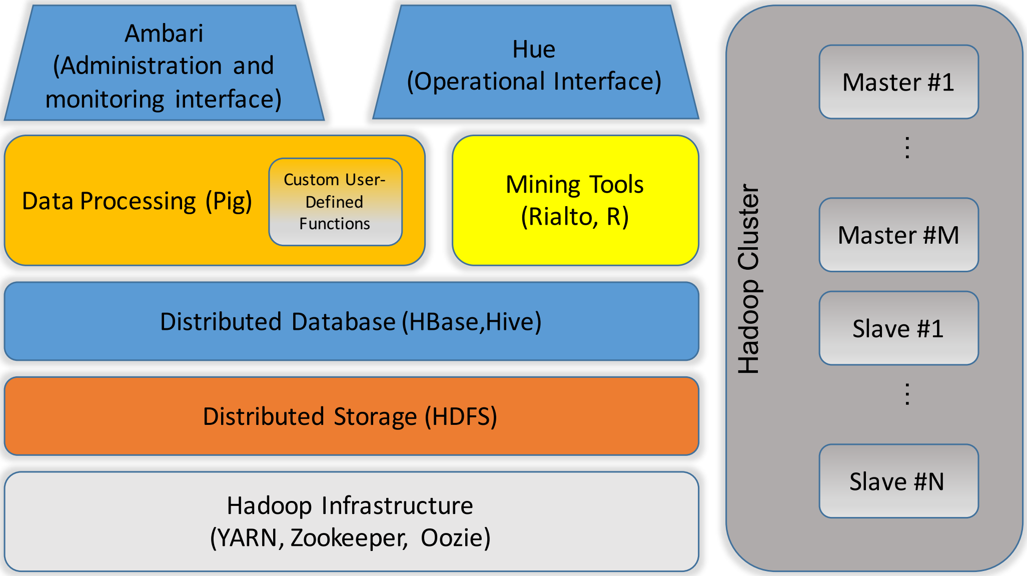

Figure 12: System architecture.

are fed to the anomaly detection and algorithms which are implemented in this layer as well.

For the case study described in this paper, the architecture was deployed on a cluster of 8 virtual machines:

- A cluster management machine with Ambari and Hue.
- Two master nodes with HBase and Hive servers and other infrastructure servers (Oozie, Yarn, Zookeper).
- Five slave nodes containing actual data in their HDFS and performing computations.

6. Related Work

Traditional maintenance is usually accomplished by devising a mathematical model of faults which can be exploited in order to diagnose asset status, predict the asset abnormality and execute suitable maintenance actions. By contrast, machine learning and data mining techniques which exploit logging data for maintenance systems are becoming increasingly important, since they can strengthen (Peng et al., 2010) or even replace model-based approaches to detect faults and malfunctions (Katipamula and Brambley, 2005; Shin and Jun, 2015; Kauschke et al., 2015b; Peng et al., 2010). The current literature mainly focuses on two aspects, namely data manipulation(Pereira et al., 2014; Kauschke et al., 2015a,c) or modeling (Rabatel et al., 2011; Holst et al., 2012; Kauschke et al., 2015a). In the former, the approaches proposed are essentially focused on issues such as feature selection and engineering, filtering and cleaning which are capable of highlighting discriminating properties characterizing faults and anomalies. In the latter, the approaches proposed are essentially based on classification and regression (Kauschke et al., 2015a; Petsche et al., 1995; Fink et al., 2013; Huang et al., 2015), outlier detection (Holst et al., 2012), pattern discovery (Rabatel et al., 2011) and time series analysis (Pereira et al., 2014; Ulanova et al., 2015).

In general, methods based on classification (either based on neural networks (Petsche et al., 1995; Fink et al., 2013) or support vector machines (Huang et al., 2015; Sipos et al., 2014)) are not well suited for the problem at hand, due essentially to the strong inbalance that the problem exhibits. This is also witnessed in (Pereira et al., 2014), where advanced techniques based on one-class classification or semisupervised learning are explored. The paper deals failures on train doors and develops an alerting system focusing on the deterioration of door systems. Similarly to our initial statements, the authors observe that warning events about abnormal working operations are not good predictors for system failures. Further, events are not i.i.d. observation and consequently their correlation has to be taken into account when making predictions about failures. Based on these premises, the authors develop a system for classifying anomalous open/close cycles within trains, based on the difference between the inlet and outlet pressure in specific intervals of the cycle. Cycles and cycle sequences are then classified as anomalous and as potential forthcoming failures. Interestingly, the evaluation points out that simple methods based on interquantile range work better than semisupervised and unsupervised methods. Compared to our approach here, the authors only focus on sensors measuring pressure, whereas we devise a general strategy for extracting features from a general set of sensors and exploiting these features for fault detection.

The approach (Holst et al., 2012) builds on the same context of our approach. The focus is on visualizing and detecting anomalous events relative to sensors from train devices. The approach relies on the capability of characterizing each feature by means of an nomaly score. The latter can be roughly expressed as the amount of instances for which the value exhibited by a given feature is more likely than the value exhibited by the instance at hand. A homogeneous poisson process is exploited, which characterizes the rate of occurrence of a given event: thus anomalous events are those events whose frequency is significantly different from the expected one, according to the estimated rate. The approach only considers event frequencies as features, whereas it has been shown (Kauschke et al., 2015c) that several aggregate features can be exploited which better characterize faults. In addition, it only provides insight about anomalous features, and does not explore correlation of such features with actual faults. Similar ideas for outlier explanation were investigated independently in (Angiulli et al., 2016).

One can also think of directly analysing time series from signals and spot aging issues which characterize a degredation in performance and ultimately a failure (Ulanova et al., 2015). Although degradation in performance can be effectively exploited in defining maintenance policies, the general issue of fault detection does not generally depends on degradation and aging issues. For example, the effect of passenger load can cause faults which are not necessarily triggered by deteriorated systems.

Anomaly detection methods were explored also by exploiting patterns of co-occurrence relationships that characterize sensors. (Wang et al., 2015; Rabatel et al., 2011; Ao et al., 2015). In (Rabatel et al., 2011) for example, the authors work on discretized values coming from sensor measurements and other contextual information (such as itinerary, weather conditions, etc.). Normal behavior is modeled by a set of sequential patterns which characterize the status of a journey. Then normal/anomalous behavior is represented by the presence of patterns which comply with/contradict such a behavior. Relating the notion of anomaly to the presence of patterns easese the task of outlier explanation which is also the core of our approach. However, compared to our approach, pattern mining is extremely sensitive to tuning based on hyperparameters. In this respect, our approach is fully automatic, in that clusters of normal behaviors are detected and explanation is obtained by scoring and separability.

7. Conclusion

In this paper we have addressed the problem of fault prediction and explanation from diagnostic data. The research was conducted within a project financed by Bombardier Transportation - one of the world's largest rail-equipment manufacturing companies, and carried out by Exeura with the purpose of preventing and explaining door failures on metro trains. These failures result in several service inefficiencies, such as delays or trip cancelations, so that a successful fault prediction may significantly reduce both operating and maintenance costs.

Diagnostic data were collected from a number of data sources storing historical events and sensor values concerning a one-year period. These data suffered from a number of issues, like size, sparsity, bias, burst effect and trust. Thus, suitable pre-processing techniques were applied to mitigate their effect. In particular, diagnostic time series were compressed in such a way that the behavior of a device in a given time frame was effectively summarized through a number of suitable statistics.

Then, failure prediction was performed by using outlier detection. The reason was that, in the given application scenario, failures do not share common patterns and, thus, traditional classification techniques perform poorly. On the contrary, we observed that failures are well characterized as non-normal devices behavior.

Fault explanation was in turn achieved by providing a snapshot of the device features exhibiting abnormal values. Such a snapshot is a description where features that exhibit abnormal behavior are highlighted and ranked, with respect to the average activity of the normal working operation. A snapshot is then an ordered list of abnormal features.

The implementation of the proposed approach relies on an architecture for data manipulation based on the Apache Hadoop framework and its ecosystem. System R (a free software environment for statistical computing and graphics) and Rialto (an extensible Business Analytics platform based on Machine Learning techniques for models induction) were both used as analytical tools.

An experimental evaluation was finally performed in order to assess the goodness of the devised approach. Results show that high-degree outliers are effective indicators of incipient failures. Also, explanation in terms of abnormal features values (responsible for outlierness) seems to be rather efficacious.

References

Aggarwal, C., 2013. Outlier Analysis. Springer.

Angiulli, F., Fassetti, F., Manco, G., Palopoli, L., 2016. Outlying property detection with numerical attributes. Data Mining and Knowledge Discovery.

Ao, X., P.L., Li, C., Zhuang, F., He, Q., 2015. Online frequent episode mining. Proceedings of the31st IEEE International Conference on Data Engineering (ICDE'15), 891–902.

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A survey. ACM Computing Surveys 41 (3), 15:1–15:58.

Dempster, A., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society, Series B 39 (1), 1–38.

Figueiredo, M. A. T., Jain, A. K., 2002. Unsupervised learning of finite mixture models. IEEE Transations on Pattern Analysis and Machine Intelligence 24 (3), 381–396.

Fink, O., Zio, E., Weidmann, U., 2013. Extreme learning machines for predicting operation disruption events in railway systems. Proceedings of the European Safety and Reliability Conference, 1–8.

Holst, A., Bohlin, M., Ekman, J., Sellin, O., Lindström, B., Larsen, S., 2012. Statistical anomaly detection for train fleets. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence AAAI'12. pp. 2217–2223.

Huang, H., Wang, H., Li, Y., Zhang, L., Liu, Z., 2015. Support vector machine based estimation of remaining useful life: current research status and future trends. Journal of Mechanical Science and Technology 29 (1), 151–163.

Katipamula, S., Brambley, M. R., 2005. Methods for fault detection, diagnostics, and prognostics for building systemsa review, part i. HVAC&R Research 11 (1), 3–25.

Kauschke, S., Janssen, F., Schweizer, I., 2015a. Advances in predictive maintenance for a railway scenario - project techlok. Tech. rep., Knowledge

Engineering Group, University of Darmstadt.
URL http://tubiblio.ulb.tu-darmstadt.de/76467/

Kauschke, S., Janssen, F., Schweizer, I., Oktober 2015b. On the challenges of real world data in predictive maintenance scenarios: A railway application. In: Proceedings of the LWA 2015 Workshop on Knowledge Discovery, Data Mining and Machine Learning.
URL http://tubiblio.ulb.tu-darmstadt.de/76093/

Kauschke, S., Schweizer, I., Fiebrig, M., Janssen, F., 2015c. Learning to predict component failures in trains. In: Proceedings of the LWA 2015 Workshop on Knowledge Discovery, Data Mining and Machine Learning.
URL http://ceur-ws.org/Vol-1226/paper13.pdf

Lazarevic, A., Kumar, V., 2005. Feature bagging for outlier detection. In: Proceedings of the 11th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'05). pp. 157–166.

Manco, G., Rullo, P., Gallucci, L., Paturzo, M., 2016. Rialto: A knowledge discovery suite for data analysis. Expert Systems with Applications 59, 145–164.

Micenková, B., Ng, R. T., Dang, X. H., Assent, I., 2013. Explaining outliers by subspace separability. In: 2013 IEEE 13th International Conference on Data Mining. pp. 518–527.

Mladenic, D., Grobelnik, M., 1999. Feature selection for unbalanced class distribution and naive bayes. In: Proceedings of the Sixteenth International Conference on Machine Learning. ICML '99. pp. 258–267.

Peng, Y., Dong, M., Zuo, M., 2010. Current status of machine prognostics in condition-based maintenance: a review. The International Journal of Advanced Manufacturing Technology 50 (1), 297–313.

Pereira, P., Ribeiro, R. P., Gama, J., 2014. Failure prediction - an application in the railway industry. In: Proceedings of the 17th International Conference on Discovery Science (DS 2014). pp. 264–275.

Petsche, T., Marcantonio, A., Darken, C., Hanson, S., Kuhn, G. M., Santoso, I., 1995. A neural network autoassociator for induction motor failure prediction. In: Advances in Neural Information Processing Systems NIPS 1995. pp. 924–930.

R Core Team, 2014. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
URL `http://www.R-project.org/`

Rabatel, J., Bringay, S., Poncelet, P., 2011. Anomaly detection in monitoring sensor data for preventive maintenance. Expert Systems with Applications 38 (6), 7003 – 7015.

Shin, J.-H., Jun, H.-B., 2015. On condition based maintenance policy. Journal of Computational Design and Engineering 2 (2), 119 – 127.

Sipos, R., Fradkin, D., Moerchen, F., Wang, Z., 2014. Log-based predictive maintenance. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14). pp. 1867–1876.

Tang, L., Liu, H., 2005. Bias analysis in text classification for highly skewed data. In: Proceedings of the Fifth IEEE International Conference on Data Mining. ICDM '05. pp. 781–784.

Ulanova, L., Yan, T., Chen, H., Jiang, G., Keogh, E., Zhang, K., 2015. Efficient long-term degradation profiling in time series for complex physical systems. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15). pp. 2167–2176.

Wang, C., Vo, H. T., Ni, P., 2015. An IoT application for fault diagnosis and prediction. Proceedings of the IEEE International Conference on Data Science and Data Intensive Systems, 726–731.

Zheng, Z., Wu, X., Srihari, R., 2004. Feature selection for text categorization on imbalanced data. SIGKDD Explorations 6 (1), 80–89.