Data-driven approach for labelling process plant event data

Débora Corrêa[1,2], Adriano Polpo[2,3], Michael Small[1,2,4], Shreyas Srikanth[5], Kylie Hollins[2,5] and Melinda Hodkiewicz[2,6]

[1] *Complex Systems Group, Department of Mathematics and Statistics, The University of Western Australia, Crawley, WA 6009, Australia*
debora.correa@uwa.edu.au, michael.small@uwa.edu.au

[2] *ARC Industrial Transformation Training Centre (Transforming Maintenance through Data Science), The University of Western Australia, Crawley, WA 6009, Australia*

[3] *Department of Mathematics and Statistics, The University of Western Australia, Crawley, WA 6009, Australia*
adriano.polpo@uwa.edu.au

[4] *Mineral Resources, Commonwealth Scientific and Industrial Research Organisation, Kensington, WA 6151, Australia*

[5] *Continuous Improvement Centre of Excellence, Alcoa of Australia, Booragoon, WA 6154*
Shreyas.Srikanth@alcoa.com, Kylie.Hollins@alcoa.com

[6] *School of Engineering, The University of Western Australia, Crawley, WA 6009, Australia*
melinda.hodkiewicz@uwa.edu.au

ABSTRACT

An essential requirement in any data analysis is to have a response variable representing the aim of the analysis. Much academic work is based on laboratory or simulated data, where the experiment is controlled, and the ground truth clearly defined. This is seldom the reality for equipment performance in an industrial environment and it is common to find issues with the response variable in industry situations. We discuss this matter using a case study where the problem is to detect an asset event (failure) using data available but for which no ground truth is available from historical records. Our data frame contains measurements of 14 sensors recorded every minute from a process control system and 4 current motors on the asset of interest over a three year period. In this situation the "how to" label the event of interest is of fundamental importance. Different labelling strategies will generate different models with direct impact on the in-service fault detection efficacy of the resulting model. We discuss a data-driven approach to label a binary response variable (fault/anomaly detection) and compare it to a *rule-based* approach. Labelling of the time series was performed using dynamic time warping followed by agglomerative hierarchical clustering to group events with similar event dynamics. Both data sets have significant imbalance with 1,200,000 non-event data but only 150 events in the *rule-based* data set and 64 events in the *data-driven* data set. We study the performance of the models based on these two different labelling strategies, treating each data set independently. We describe decisions made in window-size selection, managing imbalance, hyper-parameter tuning, training and test selection, and use two models, logistic regression and random forest for event detection. We estimate useful models for both data sets. By useful, we understand that we could detect events for the first four months in the test set. However as the months progressed the performance of both models deteriorated, with an increasing number of false positives, reflecting possible changes in dynamics of the system. This work raises questions such as "what are we detecting?" and "is there a right way to label?" and presents a data driven approach to support labelling of historical events in process plant data for event detection in the absence of ground truth data.

1. INTRODUCTION

The drive for data-driven algorithms and models to support predictive maintenance programs continues apace — a recent review of the literature on this topic is Carvalho et al. (2019). However, there has been little discussion on uncertainties as-

https://doi.org/10.36001/IJPHM.2022.v13i1.3045

sociated with the assumed response variable (event of interest). When it comes to equipment performance, identifying an 'event' such as process upsets and equipment failures, such identification usually involves a human in the loop. This is often an operator or maintainer who records the time of the event and provides a label. This label can be a fault code and/or unstructured text describing the event. But what if the historical records for these events are no longer available or considered unreliable? We cannot wait for future events in order to get a labelled data set with ground truth data. The practical alternative is to select a response variable from the available data. We discuss in our work the strategy to label a binary response variable to handle a fault detection scenario (a single type of fault).

Industrial context. The modern day mineral processing plant is highly instrumented and can be run by a small number of operators in a control room. Historically their role has been to use the information on their computer displays to keep the process in control, to anticipate loss of control events and to take proactive or reactive measures to meet production and recovery targets. A consequence of this investment in instrumentation is that there is a wealth of historical process data available and the potential to leverage it for modern data science-driven efforts to predict upsets causing equipment outages. In a number of cases the root cause of a process-related equipment outage is loss of control of key process parameters upstream of the equipment of interest. These process upsets cause deviations in the chemical and physical compositions of the product that then result in downstream equipment having problems with processing the product. One example is a higher-than-desired density product that causes bogging in equipment such as rakes and thickeners, loss of cyclone cut-points, blinding of screens and so on. Early detection of these potential events would allow for proactive actions to be taken to prevent unwanted downstream equipment outages. However, while historical records of the process plant instrumentation data can be found in the databases, records of the outage events are not always readily available. These outages may be planned events to conduct maintenance or unplanned events associated with equipment damage or equipment failure due to process upsets. Historically these outage events will have been captured in hand written operator logs and even today, while a computer may be used, the data is unstructured and the data quality may not be reliable for analysis purposes. So the challenge for data scientists is to determine how to label these events so that they have a response variable for their predictive model. When the event is associated with a single piece of the equipment, some simple rules such as the loss of motor current may be used. However mineral processing plants often have sub-systems made of multiple units in parallel and series configurations and these produce a range of different outage signatures. In this paper we look at one such system on a real industry case study and discuss two methods of labelling the response variable, one using a *rule-based* method and the other using a *data-driven* approach.

Unsupervised techniques for predictive analytics. It remains unclear in the literature how the application of data science methods can be beneficial to enhance the quality of labelled data, particularly for predictive analytics on equipment. Here, we call attention to this question and show how such techniques can be used. We understand that *data-driven* labelling can enhance the definition of events of interest and improve the quality of labelled data.

We suggest that unsupervised techniques have potential value in this scenario: they do not require manual intervention and only use the data structure to group samples with similar patterns. In general, approaches dealing with *data-driven* predictive maintenance have focused on using clustering techniques from two perspectives: as anomaly detection algorithms, and as an intermediate step in data pre-processing. One exception to these two perspectives is the use of clustering techniques to classify systems using key performance/features of subsystems as parameters that are further used in the reliability analysis (Cai, Zhao, & Zhu, 2020).

Clustering techniques have been widely used in the context of anomaly detection algorithms to identify an asset's healthy and unhealthy condition (see Erhan et al. (2021) for a review on the topic). In Kim et al. (2011), for instance, an auto-associative neural network that works as a nonlinear Principal Component Analysis is trained on labelled data to find partitions in the feature space related to normal and abnormal asset conditions. An unsupervised algorithm is applied to project new data into the learned partitions, therefore working as a soft alarm along with metrics from the trained neural network. Another use of clustering techniques is in the pre-processing step in *data-driven* frameworks. In Listou Ellefsen, Bjørlykhaug, Æsøy, Ushakov, and Zhang (2019), a combination of neural network models and genetic algorithms are used to estimate remaining useful life (RUL) annotated in the dataset. The unsupervised pre-training is used to initialise the weights of the lower layers of the neural network model to provide an optimised initialisation of the network when there is a reduced number of labelled data. The authors conclude that using an unsupervised pre-training stage provides better results in contrast with random initialisation of neural network weights when the availability of labelled data is an issue.

Another example of the use of clustering techniques in a pre-processing step is the work of Reder, Yürüşen, and Melero (2018). They study the relationship between wind turbine failures and environmental variables and the degradation process, using five wind turbine systems: gearbox, generator, pitch system, yaw system and frequency converter. Labelled response variables are the exact times and duration of the fail-

ures for each turbine. They compare manually supervised and unsupervised approaches for the labelling strategy of the input parameters (not the response variable). In the supervised case, expert opinions and information from the literature are used to manually define the categories representing the parameters used as input to the association rule mining algorithm used in their work. For instance, wind speed can be classified as calm, low, high, etc.; or relative humidity that can be classified according to corrosiveness. In the unsupervised case, a K-means clustering is used to group the parameter values, so there is no expert input. Each combination of the parameters is associated with a cluster/category, and that category is used as input to the rule mining algorithm. Next they compare the performance of both approaches for the association with failure. They found that performance depended on the amount of available information and the number of obtained rules (higher number of rules increases the complexity of interpretation of the results). Both labelling approaches generated a higher or lower number of rules for different scenarios. For instance, the clustering algorithms found three labels for severity, while the supervised labelling only used two. On the other hand, the expert labelling used eight categories for temperature, while the clustering found three.

Our research goals. We can find many prognostics papers in the academic literature where the problem is clearly stated, and the data is available, labelled and organised. Often these works are laboratory experiments where everything is controlled, and hence the data is labelled clearly and well behaved. An extraordinary proportion of published prognostic models are tested on just four laboratory and simulated data sets (Ferreira & de Sousa, 2020; Lei et al., 2018): Turbofan engine degradation simulation data set (Saxena & Goebel, 2008), FEMTO Bearing Data Set (Nectoux et al., 2012), Bearing data sets (Lee, Qiu, Yu, & Lin, 2007), and the Milling data set (Agogino & Goebel, 2007). In these data sets, there is no lack of clarity around the labelling of the dependent variable. However, giving our experience, there are real challenges when working with real industry problems and data (Astfalck, Hodkiewicz, Keating, Cripps, & Pecht, 2016). The data analysis process depends on many decisions that come before the model selection (Sambasivan et al., 2021). However, all of this is done conditioned on trust that the response variable is labelled 'correctly' and the data is related to the problem of concern. Under this scenario our research goals are 1) explore an unsupervised data-driven labelling of a response variable, and 2) explore the impact of different labelling strategies on an end-to-end modelling process.

The paper is organised as follows. In Section 2, we describe the case study context, data and the organisation's motivation for the project. Section 3 describes data preparation and shows how definitions of events affect the number and type of events labelled in the data set. Section 4 describes the development of two models on each of the labelled data sets with the results described in Section 5. Finally, Section 6 discusses the lessons learned and implications for the prognostics community.

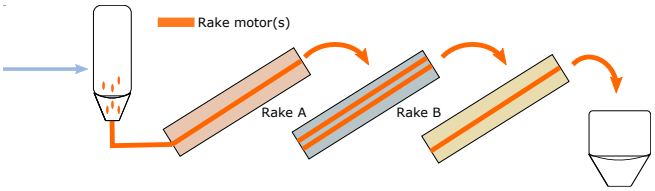


Figure 1. Simplified diagram showing how the rakes classifiers operate. Sand separation is performed by mechanically ranking the sand up an incline.

2. CASE STUDY

We consider a mineral processing system in which there are three mechanical classifiers called sand rakes. The purpose of the rakes is to separate solids (sand) from liquid (liquor). Upstream of the sand rakes is a complex processing plant involving chemical reactions and physical changes to the product. The sand rakes are a set of three units arranged in series as shown in Figure 1. Units A and C have one motor, and unit B has two motors. Slurry (a mix of solids and liquor) flows from rake A, B to C. The rake classifier uses rakes actuated by an eccentric motion causing them to dip into the settled material and to move it up the incline for a short distance. The rakes are then withdrawn, and return to the starting-point, where the cycle is repeated. The settled material is thus slowly moved up the incline to the top of the classifier where it is discharged. For a illustration of a rake classifier see Michaud (2015).

The eccentric motion of each rake is driven by four motors A, B1, B2 and C, where B1 and B2 are the two motors of rake B, as illustrated in Figure 1. The motor current is monitored and is the key indicator of the operating status of the rake. Changes in motor current are indicative of operating problems with the rakes. A loss of motor current indicates an outage, however this can be due to either desirable events (such as planned maintenance) or undesirable events. Our interest here is in an event called "bogging". This is an unplanned event caused by higher than desired load on the rakes, often associated with higher density and higher tonnage slurry. This additional load causes the rakes to 'bog' or stop moving. The remedy is for operators to free the rakes which can involve having to manually enter the rake and dig out material. This can create both an unwanted safety exposure for the operator and flow loss.

Our data frame contained measurements of 14 sensors from the process control system and 4 current motors over a three year (2016-2018) period. Variables are collected at a frequency of once per minute. Other than the rake motor current data that is specific to that rake, all other variables are relevant

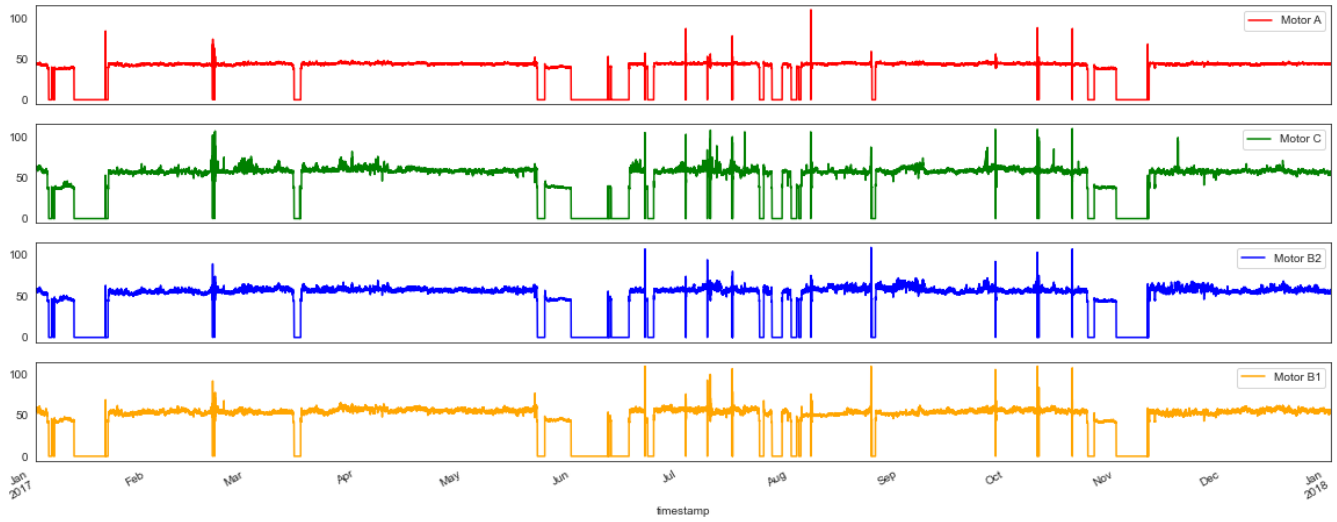


Figure 2. Illustration of the motors currents behaviours over one year.

to the entire system. There was no supplementary information from operator or maintenance logs provided. As a result there are no labelled records as to which events are actual bogging events. If labels could be generated and proved useful in predicting unplanned events, the value to the company in terms of avoiding process upsets is considerable, since these process upsets can result in operators having to enter and clean the sand rake units.

3. TRANSFORMING DATA TO A MODEL INPUT

An essential part of creating a model for detecting an event of interest is to prepare the process control data from the 18 sensors in a format to be used for modelling. Decisions made in this stage affect interpretations of the model and suitability of its performance. In the following subsections we describe the data preparation.

3.1. Identifying bogging events

Initially, we applied the following definition: "a bogging event occurred when the rake was offline for a period between 1 and 60 minutes". We label the response variable resulting from this definition as the *rule-based* response variable.

The primary means of determining the status of the system is the motor current reading. In Figure 2 we illustrate the behavior of the motor current of each rake in a system for a period of one year. Overall, the four time series are synchronized, presenting common periods of offline (zero current) and periods where the currents operate in a mean operational point at about 50% FLA (Full Load Amps). Some of these offline events do not relate to bogging.

First we had to construct a data set using the the *rule-based* response definition. We use a subset of sensor data representing the motor currents to find the event times, as per the *rule-based* definition above. We perform the following two steps:

1. For each motor current, we find patterns of continuous "offline" periods falling in the range of one to sixty minutes. We consider that an event has occurred whenever any one of the motors go offline. This step provides a set S_1 of event times with 249 events.
2. Group events that happened *exactly* at the same time in two or more motors regardless of the period they persist in the offline status. For instance, if motor A goes offline at 15:01 and stays offline for 20 minutes, and motor B2 goes offline also at 15:01 and stays offline for 10 minutes, they are considered as one event with a duration of 20 minutes (we consider the entire system is offline for a given minute if one or more motors are offline together in that minute). This step reduces the number of events and generate a subset of S_2 with 150 events.

Figure 3 shows six examples of some challenges with data labelling. For each example in Figure 3 there are one or more bogging events under the *rule-based* definition. We indicate the start time (the time when the first motor goes offline) of the event with a vertical dashed red line. Different colors represent the motors of the four rakes as detailed in the legend of Figure 3 (f). We complement the information of Figure 3 with Table 1 which shows the times and associated motor(s) for each event in each of the six examples. We can see that the behaviour of the four motors are quite different in each case. In Figure 3 (a) there are six events in about 70 minutes. We now start to recognise possible problems in the definition of the event. The detection of a sequence of events that are close together would be challenging for any model as there

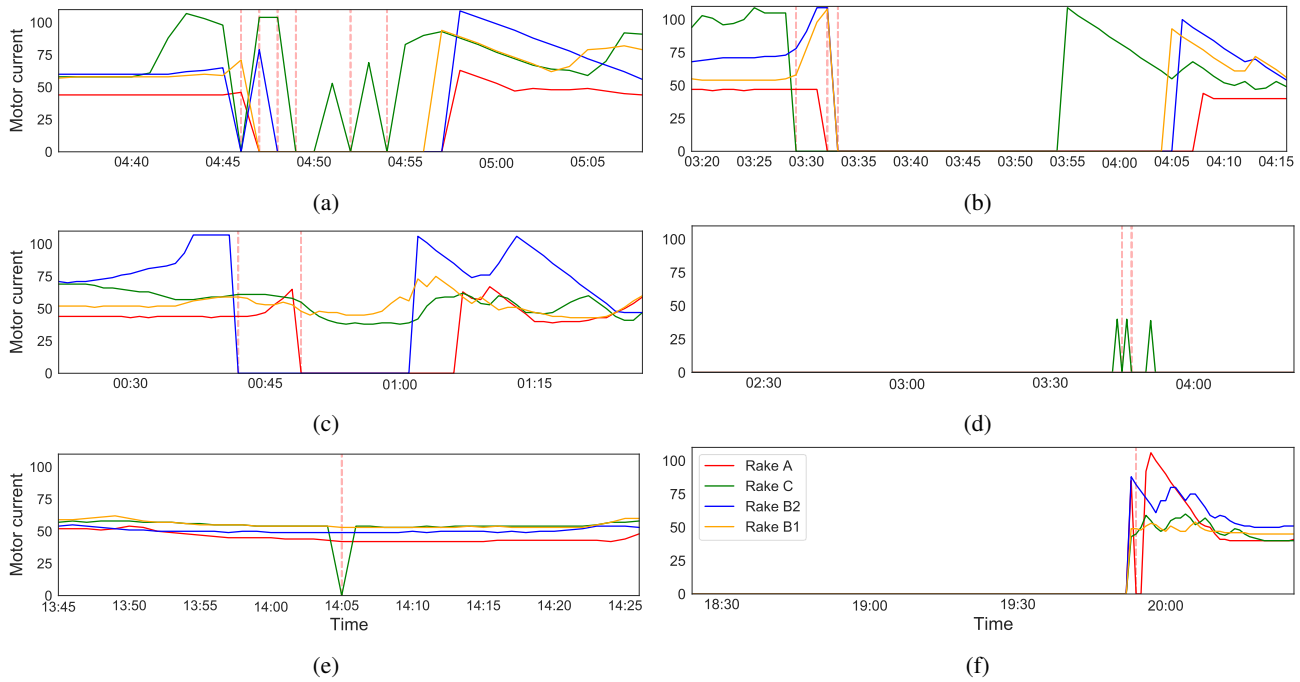


Figure 3. Examples of possible problems in the definition of a bogging event. In (a), six events (according to the *rule-based* definition) happens in a short time scale (see Table 1 for details about the event times and corresponding motor(s)). In (b), three events show synchronized behavior of motors with a short delay in time. In (c), only two motors go offline while the other two remain online. In (d), motors were offline for a longer period than 60 minutes except that, the current of one or other motor presented a quick online-offline-online pattern that falls into the specified interval of a bogging event. In (d), one motor went offline for a minute without any dynamics before this event. In (f), motors were offline for longer than 60 minutes except for a quick moment of online-offline observed for one motor.

are chances they are related. Moreover, how can the time series of upstream process variables — which are common to all the motors — be used to differentiate such behaviour? In Figure 3 (b) there are three events according to the *rule-based* definition, but all motors are offline together after less then 5 minutes. In Figure 3 (c), only two of the motors go offline, does this characterise an event? In Figure 3 (d) motors were all offline, but the current of one motor presented two quick online-offline-online patterns that fall into the specified interval of a bogging event. Would this likely indicate some maintenance intervention? In Figure 3(e) all motors seemed to be operating normally. Still, one motor went offline for a minute without any clear dynamics before this happened. Could this indicate a sensor measurement error? In Figure 3 (f) motors come from a long offline period and one motor has a quick moment of online-offline that again falls into the specified time interval of a bogging event. Would this indicate a maintenance period?

The result of step 2 applied in Figure 3 is a set of event times as presented in Table 1. We performed this process for all data sets. The resulting event times are used in Section 3.3 for feature engineering and data set labelling.

Table 1. Event times from examples presented in Figure 3.

	Time	Motor(s)	Description
(a)	04:46 04:47 04:48 04:49 04:52 04:54	C,B2 A,B1 B2 C C C	Motor currents go to zero, but one of them is varying.
(b)	03:29 03:32 03:33	C A B2,B1	All motors have a period of motor current zero.
(c)	00:42 00:49	B2 A	Two motors went to motor current zero.
(d)	03:45 03:47	C C	Zero motor currents most of the time, but only three minutes with motor current different from zero.
(e)	14:06	C	A zero motor current for a single minute.
(f)	19:54	A	Long period of zero motor current, then only one motor (after a few minutes after motor current changes to non-zero) get zero again.

3.2. Data-driven event definition

The problems identified with the *rule-based* definition of the response variable motivated us to construct *data-driven* alter-

natives based on the dynamics of the data. The main idea is to automate pattern matching of these dynamics in order to facilitate the selection of suitable representatives of the event signature by an expert. We use a time series clustering approach to filter event signatures with similar dynamics and create a new dependent variable for our problem. Clustering analysis of time series is performed using Dynamic Time Warping (DTW) (Muda, Begam, & Elamvazuthi, 2010) as a measure of similarity between two time-series. Then, an agglomerative hierarchical clustering is used to group similar event dynamics. The proposed clustering approach for creating the new response variable follows the steps below.

1. Use the current data and event times to determine the time series to be used as input of the clustering algorithm.
2. Use the time series resulting from 1) to compute a distance matrix using DTW.
3. Fit Linkage Tree (Agglomerative Hierarchical Clustering).
4. Plot dendrogram and select cut-level to get clusters.
5. Apply cut-level and use expert/engineer input to discard inconsistencies in the *rule-based* definition of the response variable possibly related to maintenance, controlled tests or sensor errors.

Determining the time-series to be used as input of the clustering algorithm. Our starting point is the four time series corresponding to the motor currents and the *rule-based* event times computed from the *rule-based* definition – examples are given in Figure 1. We apply three steps to prepare the time series for the clustering algorithm. First, the event times indicate the start of the events, but, to group similar dynamics that are representative of pre-event and post-event, we need to consider an interval of time before and after the event start, respectively. For example, if we want to group the situations as in in Figure 3 (d) we need to capture the long offline patterns of the motor currents before the event, and we need to know that they also stayed offline after the quick online-offline-online pattern. Therefore, the first step is to determine what the time interval will be. Our second step group events that happen in a short time scale. For instance, if we use a window of 10 minutes, all the events presented in Figure 3 (a) would be considered a unique event. This step maximises the number of events that can be detected by considering a cluster of possibly events as representative of one event.

The time interval window. The time interval should be long enough to capture important dynamics anticipating and proceeding the event, but it shouldn't be too long to bring information that is related to past or future events. Our purpose here is to group offline-online patterns of the time series that relate or not to an event of interest. However, we checked the number of resulting groups identified by the clustering strategy according to time intervals of one and two hours and window sizes of 5, 10 and 15 minutes to group similar events. This information is presented in Table **??**. We argue that the temporal changes in the current dynamics that are signature of the pre-event and post-events can be adequately represented in these periods of one and two hours as the number of groups for both situations was exactly the same. For the strategy of grouping close events, the number of resulting clusters change only marginally for different window sizes and should not have an significant impact in our analysis.

Given the four time series representing the current of the motors are synchronised most of the time as illustrated in Figure 2, our third step is to use the mean of all four motor currents to get one single representation of the shape of those time series (a prototype) to be used as input for the clustering. For the situations where they are not synchronised all time (for instance, Figure 3 (d), the resulting shape of the prototype is still representative of the dynamics.

Using DTW to measure distance between time-series. In time series analysis, Dynamic Time Warping (DTW) is a popular technique used to measure the similarity (or distance) between two time series (Sakoe & Chiba, 1978). It uses the dynamic programming method to find the best alignment (minimum cumulative distance) between two temporal sequences. DTW has been used in several pattern recognition applications as it tends to better capture points with similar geometric shapes (Li, 2015). Let $X = \{x_1, x_2, ..., x_N\}$ and $Y = \{y_1, y_2, ..., y_M\}$ denote two time series with lengths N and M, respectively. The first step of the technique is to build a $N \times M$ cost matrix C_{XY}, where the (ith,jth) element corresponds to the cumulative pairwise distance between points X_i and Y_j:

$$C_{XY}[i,j] = d(x_i, y_j) - \min\big(C_{XY}[i-1,j], C_{XY}[i,j-1], C_{XY}[i-1,j-1]\big) \quad (1)$$

The Euclidean distance is often used for computing $d(x_i, y_j)$, but other distances can be used. $C_{XY}[N, M]$ will contain the distance according to the best alignment between the two time series. DTW works by aligning the time-series to find the minimum cumulative distance $C_{XY}[i, j]$ between $X[1 : i]$ and $Y[1 : j]$ as indicated in Eq. (1). This distance represents the optimal alignment between the two time-series.

Applying Agglomerative Hierarchical Clustering. The basic principle of clustering techniques is to group time series according to a similarity criterion. The idea is to maximise the similarity between time series in the same cluster and minimise the similarity between time series in different clusters. While partitional or non-hierarchical clustering methods use a fixed number of clusters and a single partition of the data, hierarchical methods use a series of partitions that are taken progressively. If the hierarchical clustering is ag-

glomerative, the procedure starts with N objects (time series) as N clusters and then successively merges the clusters until all time series are joined into a single cluster. Conversely, divisive hierarchical clustering starts with all the time series as a single cluster and splits it into progressively finer subclusters. Our starting conditions are the individuals events as defined by the *rule-based* response variable, and our aim is to find similarities between them, motivating the use of the agglomerative approach.

To perform hierarchical cluster analysis it is necessary to define three main parameters: the similarity criterion to quantify the similarity between every pair of time series in the dataset (DTW in our case); the linkage method, used to measure the distance between two clusters, and number of desired clusters, an issue that is directly related to where to cut the dendrogram resulting from the clustering. We adopted the single linkage method that assigns the distance between two clusters as the closest distance between all pair of points across the two clusters (Murtagh & Contreras, 2012).

Selecting cut-level of the Dendogram. Hierarchical clustering algorithms such as the one used here can be represented by a dendrogram. To find suitable clusters, we need to cut the dendrogram at a specific level. Different cut-levels will result in different final clusters. To date, there is no clear-cut solution to automatically find the cut-level of the dendrogram, as clustering is essentially an exploratory analysis. Hence, the interpretation of quality of the obtained clusters will depend on context. However, there is an extensive literature proposing different criteria that can be used for this matter (Jung, Park, Du, & Drake, 2003; Steinley & Brusco, 2007; Charrad, Ghazzali, Boiteau, & Niknafs, 2014). Examples include the silhouette method (Rousseeuw, 1987), the Dunn's validity index (Dunn, 1973), and the gap statistic method (Tibshirani, Walther, & Hastie, 2001). Here, we selected the cut-level according to a visual inspection of the clusters, since the primary objective is to separate relevant failure modes from inconsistencies presented in the data (and not to find the best partition of the clustering).

Figure 4 presents examples of events grouped in four clusters. Motor current dynamics represented by the first two clusters in Figure 4 could be indicative of maintenance scheduled events or controlled tests. On the other hand, there is an increase in the motor currents before the event on the last two groups in Figure 4. We can also observe that such events follow a chain of offline patterns. In the *rule-based* definition of our response variable, these events in a chain are considered distinct.

Using expert/engineer input to discard inconsistencies in the rule-based definition of events. The clusters are useful to separate events with specific dynamics. However, after applying the clustering, a visual inspection by an expert was performed to filter the events in the clusters that are unlike to be representative of real bogging events (like the situations presented in Figure 3) and/or the first two groups of Figure 4.

3.3. Feature engineering and data labelling

A standard way to proceed with the data labelling is to use event times to create a minute-based binary response variable (transforming the task into a classification problem). For each minute, a value of zero represents non-event times, and a value of one represents event times. There are other possible procedures to analyse/model our data that are based on signal process or time series analysis, for instance. However, we aim to discuss the impact of different choices in data acquisition and preparation, showing possible effects of decisions through data analysis steps that occur before and after the model. That is, we focus here on the whole process and not on specific techniques.

Figure 5 presents details of the labelling process. For each minute of data, we have the sensor data (18 values) and the binary response variable associate to it (we exemplify the approach using only the motor current values). First, we can see that when there are two events, only one minute apart, the *rule-based* definition will consider them as separate. This is another indication that the *rule-based* definition poses challenges for a model; either there is not enough data to detect the second event or would it be two different events? The training process will, therefore, discard this event. Second, we can also observe another decision in the process: we discard the data related to the event's duration (yellow are in Figure 5). In the example, the duration of the event of motor A (red line) is the longest and therefore determines the area to be discarded. Third, a period after the event might also be discarded, or we can decide to use all available data, as illustrated in the figure. We note that the feature engineering process utilising a sliding window approach will need to wait for the first k minutes after the event to have enough data to process the features (similar to not being able to detect the second event), where k is the size of the sliding window. Under a *data-driven* process we have many decisions to take when preparing the data.

After this process, we have a data frame of minute-scale data, with 18 columns indicating the sensor data and one column showing the binary response variable. Finally, we discard data when the motors currents are zero for longer than sixty minutes. There are, however, situations when only one or a combination of the motors are offline together. Therefore, we have considered that a regular operation condition of the asset is when all motors currents are not offline, and then we opted to discard the data when one or more motors are offline.

After the data pre-processing step, we can proceed with feature extraction strategies. The usual approach for feature engineering when dealing with time series data is to use an overlapping sliding window of size k to extract statistics descrip-

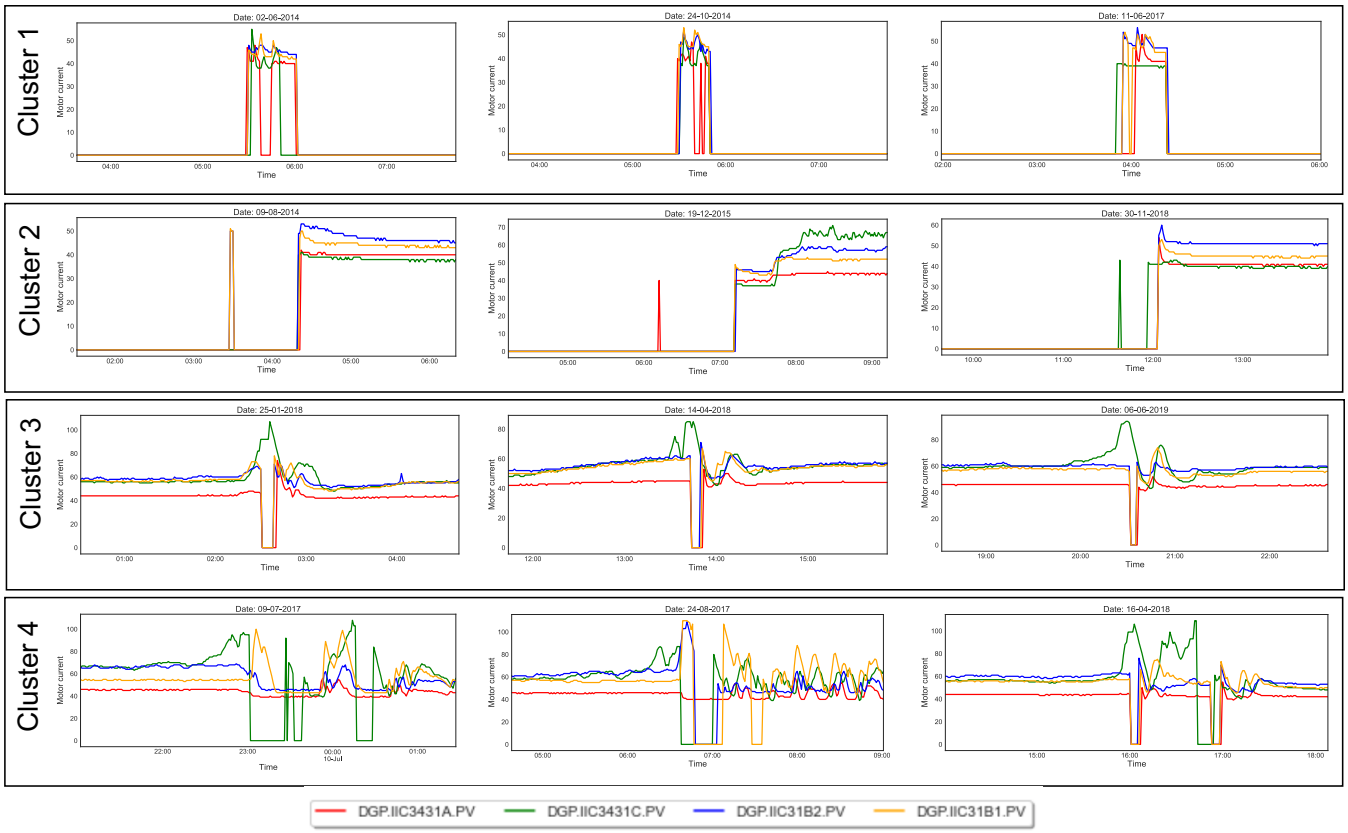


Figure 4. Time series plots of rake motor current dynamics according to four clusters. In Cluster 1, the motors stay offline except for short moments of current peaks and falls. In Cluster 2, the rakes go online after a long offline period. In the Cluster 3, the rakes were operating, and an increase in the rake currents (most prominent in C rake) can be observed in all cases before an event. Motor current levels are unstable after the event. Finally, in Cluster 4, the current rises similar to Cluster 3 however there is an extended period in which motor current levels are unstable and this is accompanied by a number of events within a time interval of about 1.5 hours, where one or more rakes go offline again.

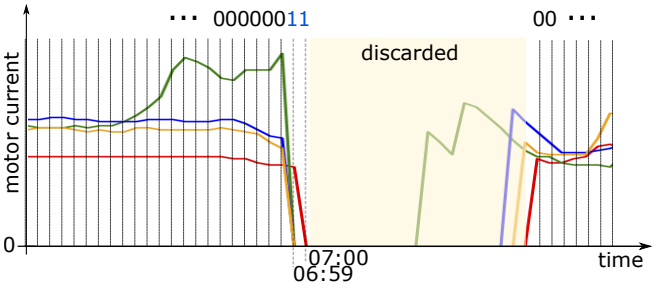


Figure 5. The process of creating a binary response variable for the training set. For each minute, we consider the value of the sensor data (here we show the four motor current values to simplify) and associate a label to it. A label of 1 indicates the event (blue 1's indicate an event at 06:59 and another event at 07:00), while a label 0 indicates no event. We discard the data during the event as represented by the yellow are in the figure.

tors to summarise the dynamic information of the time series in that window. A step size s defines how we move the window forward. If $s = k$ the window do not overlap. If $s = 1$ the window overlap in $k - 1$ elements. A simplified diagram of this process is presented in Figure 6. The label associated to each widow is given by the value of the binary response value at each time k. The features we used include eight statistic descriptors of each window: the mean, the standard deviation, the maximum value, the minimum value, and the 10%, 25%, 50%, 75%, and 90% percentiles.

4. MODEL

We used the features extraction processing described in the previous section as input to two models: a Logistic Regression and a Random Forest. We used a Grid-search strategy with cross-validation to estimate the hyper-parameters of each model. We also normalised the features to have zero mean and unit standard deviation. We have substantial imbalanced data, with over 1,200,000 non-event data and less than 150 events in, for example, the *rule-based* labelled data set.

We have two strategies to handle the imbalanced data: 1) run the model as the data is (imbalanced data (ID) strategy); and

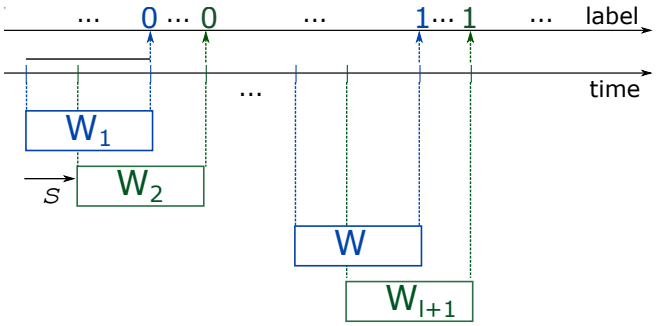


Figure 6. Feature engineering in a sliding window approach. Several features are extracted from the time series inside each window. The step size s define how we extract features from consecutive windows of size k. The label of each window is determined as the corresponding value of the binary response at time k of the window.

Table 2. Hyper-parameter range for fine-tuning grid search with cross-validation.

Model	Hyper-Parameter	Grid
LR	λ (penalisation parameter)	$\in (0.01, 10)$
RF	max depth	None, 5, 10, 20, 30
	number of estimators	20, 50, 100
	max features	'auto', 'sqrt', 'log2'
Both	threshold	$\in (0, 1)$

2) consider a balanced data set based on a sub-sampling (balanced data (BD) strategy). We chose the balanced accuracy (BA) as a metric to select the best model on the training data. The balanced accuracy (BA) is defined as (TPR+TNR)/2, where TPR= $tp/(tp + fn)$, TNR= $tn/(tn + fp)$, tp is the number of true positives, fp is the number of false positives, tn is the number of true negatives, and fn is the number of false negatives. This metric considers the number of correct event detections in both categories (event and non-event). We discuss the model selection in Section 5. The best model in the training stage selected according to each strategy is used on the (imbalanced/full) test set, that is left untouched. The balanced data strategy (2) works as follows:

1. We select all the p events samples and sample (at random) without replacement p non-events samples. This gives us a balanced version of the training dataset, with p non-event samples and p event samples.
2. We apply Grid-Search (see Table 2 for the hyper-parameters considered) with 10-fold cross-validation on this balanced dataset.
3. We compute the balanced accuracy (BA, defined in the sequel) on this training set and save the model hyper-parameters.
4. We repeat steps 1-3 m times.
5. We choose the best model (highest BA) from all m repetitions.

The Grid-Search and cross-validation techniques also require the specification of an evaluation strategy when seeking the best combination of parameters of the models. Here, we adopted the BA metric. Given the limitations of the accuracy for imbalanced data, the balanced accuracy is considered preferred as an overall performance metric for a model.

A summary of the main decisions in the whole data analysis process discussed so far is presented in Table 3.

We emphasise that while we believe these design choices to be appropriate, the modelling building challenge is not the focus of our paper. Many machine learning approaches could be applied here without affecting our primary conclusion.

5. RESULTS

We split our data in the following way. For the training set, we used data in the period 01/Mar/2016 – 31/Dec/2017, and, for the test set, we used the period 01/Jan/2018 – 15/Dec/2018. There are around 1,200,000 nonevent data. After we separate the training and test sets, we verified a total of 81 events in the training set and 69 events in the test set for the *rule-based* response variable, while there are 29-31 events in the training set and 31-33 events in the test set for the *data-driven* based response variable. We now check how many of these events can be detected in reality according to different window sizes for each of the response variable definition. This information is presented in Table 4. We recap that we can detect an event only if we have at least k data samples preceding it, where k is the window size.

The results from the *data-driven* labelling for the different values of t_g (time, in minutes, used to group events as a unique event) have produced similar results in terms of which events are identified. Similarly the results obtained for different window sizes k identified similar events. For the sake of brevity, we present the results considering $t_g = 10$ for the *data-driven* labelling of the response variable, and window sizes of 1 min and 30 min for both response variables.

Our analysis is a multi-step process, described below. In this description we necessarily select/define many aspects of the project and hyper-parameters (see Tables 3 2). We have considered for the Logistic Regression model an L1-penalty (LASSO) for features selection. Again, the specifics of these choices is not the main message of this paper. The process is summarised as:

1. We have two data sets with different response variables: the *rule-based* data, and the *data-driven* data. We performed an independent analysis of each data set.
2. For each data sets:
   (a) Our data can have current data per minute ($k = 1$) or features in a sliding window of 30 minutes. The same applies for the other sensor data. We decided

Table 3. Hyper-parameters used in the data preparation, feature extraction, and modelling.

Step	Hyper-Parameter	Values used	Pertinent to
Defining the events	Periods of offline	1-60 minutes	*rule-based*, *data-driven*
	Period to consider subsequent events as a unique event	15,20 and 30 minutes	*data-driven*
	Time before and after the event to create time series used as input to clustering	1 and 2 hours	*data-driven*
Feature extraction	Window size	30, 50 minutes	*rule-based*, *data-driven*
	Step size	1 minute	*rule-based*, *data-driven*
Modelling	Model	Logistic Regression and Random Forest	*rule-based*, *data-driven*
	Grid-Search evaluation metric	balanced accuracy	*rule-based*, *data-driven*
	Dealing with imbalanced data	Sub-sampling	*rule-based*, *data-driven*

Table 4. Number of events that *can* be detected for different window sizes k and labelling strategies.

			k (window sizes)							
	# Events		**1**		**30**		60		120	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Rule-based	**81**	**69**	**32 (40%)**	**36 (52%)**	**26 (32%)**	**26 (38%)**	20 (25%)	20 (29%)	20 (25%)	19 (26%)
Clust. ($t_g = 5$)	31	33	31 (100%)	33 (100%)	25 (81%)	26 (79%)	19 (61%)	19 (58%)	19 (61%)	18 (55%)
Clust. ($t_g = 10$)	**29**	**33**	**29 (100%)**	**33 (100%)**	**25 (86%)**	**25 (76%)**	19 (66%)	19 (58%)	19 (66%)	18 (55%)
Clust. ($t_g = 15$)	29	31	29 (100%)	31 (100%)	25 (86%)	25 (81%)	19 (66%)	19 (61%)	19 (66%)	18 (58%)

to investigate the performance of the models with the data as it is, per minute, without any featuring extraction. Our motivation was to see if the strategies of feature extraction (always assumed as better than using the data itself) can indeed improve the results.

(b) Imbalanced and balanced data strategies: as discussed in section 4, we have few events when compared to the total entries in the data (*rule-based* 150, *data-driven* from 60 to 64 events, from around of 1,200,000 samples).

(c) We performed the analysis considering the data as it is (imbalanced data) and with the balanced data strategy in the previous section.

(d) Split the data into training and testing sets. We are using the first two years of data to train the model and the last year (test data) to evaluate the quality of the model for event detection. The test data is not used for any model selection nor for hyper-parameter definition.

(e) We are considering two different class of models: Logistic Regression and Random Forest.

(f) We used Grid-Search with k-fold ($k = 10$) cross-validation to tuning (define) the hyper-parameters values to find the best tuning for the model.

(g) The model output is a score between 0 and 1, and we need a threshold to define if the final label will 0 (non-event) or 1 (event). For that, we optimise the threshold ($\in (0, 1)$) to find the one that gave us the best BA (the adopted metric) in the training data and apply the same threshold in the testing data.

Rule-based data set. Results from Tables 5 and 7 suggest that both LR and RF had similar results. We have organised the data in different ways to estimate the LR and the RF. We considered balanced or imbalanced data; features from current data, or sensor data (excluding the current data); no feature extraction (window size of 1 min), or featuring extraction (window size of 30 min), as described in Section 3.3. Table 5 shows the BA metric for all scenarios, the best result for the LR was considering balanced data, current data as features, and window size of 30 min, achieving a BA of 0.995. For the RF, we had two different scenarios achieving a BA of value 1. Then, we selected the one with only current data to allow a better comparison with the LR scenario: the imbalanced data (no need for sub-sampling), with current data features and window size of 30 min. However, test results may be indicating that the model over-fitted.

Table 5. The best BA considering the grid-search results on the training set and the *rule-based* definition of the response variable.

		Window size (min)			
		1		30	
		LR	RF	LR	RF
BD	Current data	0.983	0.975	**0.995**	0.993
	Sensor data	0.872	0.926	0.915	0.953
ID	Current data	0.843	0.999	0.923	**1.000**
	Sensor data	0.577	**1.000**	0.672	0.923

*BD: Balanced Data; ID: Imbalanced Data; LR: Logistic Regression; RF: Random Forest

The selected hyper-parameters of the best scenarios for each model and the corresponding results on the training and test sets are given in Tables 6 and 7, respectively. In Table 7 we present, together with the metric BA, the metrics TNR, F1-score, Recall, and Precision, for both training and test data

sets. Precision is defined as $tp/(tp + fp)$. In our case precision informs the ability of our model to identify (*only* the true events. Recall is also referred as TPR. See Section 4 for the definitions of TPR, TNR, tp, fp, tn, and fn. The $F1$-score computes an harmonic mean of precision and recall as $F1 = 2 \times \frac{precision \times recall}{precision + recall}$. This calculates the metrics for the positive class only. On the other hand, the $F1$-weighted is calculated for both the positive and negative class and the number of samples of each class is used to balance the final score.

Table 7 shows that the metrics for best LR are slightly worse in the test data than in the training data. However, for RF, the metrics in the test data are much worse than for the training data. This particular situation is common when we have the problem of overfitting. The model was able to fit perfectly the train data, but generalised badly. In this case, we consider that the RF model is not suitable to detect the events in this data set.

An important fact is that, for a window size of 30 min, we have originally 69 events in the test data. Nevertheless, we have only 26 events (38% of the total) in the test data to try to detect (see Table 4). For the other 43 events, we do not have enough data, and we cannot produce any detection for them. In Figure 7, we present the labelled events (those that the labelling identified) in the test data and the detected events, considering the LR. As we can see in the figure, the model detected well the events in the first four months of the test data. However, after this, we have many wrong detections that are false positives, reflecting the poor Precision and F1 results on both training and test sets as indicated by Table 7.

Moreover, Figure 7 is suggesting that the operating pattern of system may have changed. There is a significant deterioration in the model performance after a period of about four months after the models were built. To further test this, we extended the training set to 2.5 years and reducing the test set to the last 0.5 year. The results did not improve and presented similar poor results on the 0.5 year test data – further supporting our presumption of a local or global change in the system, for instance, different operating profiles, different sand characteristics, etc. Investigating this was beyond the scope of this data analysis project.

Data-driven data set. Results obtained for the *data-driven* approach are presented in Tables 8–10, and Figure 7b. The results are similar to those obtained for the *rule-based* data. The best results of the models were achieved in similar scenarios. We had similar results in the set of the hyper-parameters. The RF model presented the overfitting problem too. However, we have an important difference: the total number of events in the *data-driven* test data is 33 (against 69 for the *rule-based* data), and we have enough data to provide a model output for 25 (76%) of them (against 38% for the *rule-based* data). Furthermore, the total number of events that we can be detected

Table 6. Scenario and hyper-parameters for the best models of the *rule-based* data.

LR	Balanced data Current data features Window size of 30 min $\lambda = 0.1$ threshold = 0.53
RF	Imbalanced data Current data features Window size of 30 min max depth = None number of estimators = 100 max. features = auto threshold = 0.26

Table 7. Results of best model for the *rule-based* definition of the response variable.

	LR		RF	
	Train	Test	Train	Test
BA	0.995	0.879	1.000	0.570
TNR	0.990	0.950	1.000	0.999
F1-score	0.006	0.002	1.000	0.108
F1-score (weighted)	0.995	0.974	1.000	0.997
Recall	1.000	0.807	1.000	0.076
Precision	0.003	0.001	1.000	0.181

with the *original* definition was 26 (only one more than for the *data-drive* strategy). That is, when we compared the set of events from both definitions, we verified that many of them are the same in both data sets, and, consequently, the features values are similar for both approaches. Then, it was expected that the models would have similar performance. However, this confirms that the cluster analysis performed before any modelling to build the labelling of the events has grouped the *original* events in a reasonable way. However, in theory, both approaches have a different way of labelling the set of events, and it is not correct to compare the metrics of these methods.

Table 8. The best balanced accuracy considering the grid-search results on the training set and the *data-driven* definition of the response variable.

		Window size (min)			
		1		30	
		LR	RF	LR	RF
BD	Current data	0.980	0.997	**0.997**	0.999
	Sensor data	0.870	0.955	0.916	0.957
ID	Current data	0.844	0.965	0.919	**1.000**
	Sensor data	0.517	**1.000**	0.639	0.959

*BD: Balanced Data; ID: Imbalanced Data; LR: Logistic Regression; RF: Random Forest

Both *rule-based* and *data-driven* data sets resulted in lower precision and F1-score (see tables 7 and 10. This is due to the many false positives compared to the number of true positives. However, we recall that we have a model output for each minute. When we compare the quantity of data in the test set (around 420,000 minutes) to the total of false positives

Table 9. Scenario and hyper-parameters for the best models of the *data-driven* data.

LR	Balanced data Current data features Window size of 30 min $\lambda = 0.1$ threshold = 0.84
RF	Imbalanced data Current data features Window size of 30 min max depth = None number of estimators = 100 max. features = auto threshold = 0.21

Table 10. Results of best model for the *data-driven* definition of the response variable.

	LR		RF	
	Train	Test	Train	Test
BA	0.997	0.925	1.000	0.6
TNR	0.994	0.970	1.000	0.999
F1-score	0.011	0.003	1.000	0.0.167
F1-score (weighted)	0.995	0.984	1.000	0.996
Recall	1.000	0.880	1.000	0.200
Precision	0.005	0.001	1.000	0.142

(around 15,000), we have around 3.5% of false positives. As can be viewed in Figure 7a and Figure 7b, we can see that most of the false positives are after the four months for both data sets. Also, the F1-score weighted, that is a measure more appropriated when we have imbalanced data, is presenting values close to 1, for all cases (including test data), which does not reflect the results presented in Figures 7a and 7b. The weighted F1-score is the average of the F1-score for class 0 (nonevent) and the F1-score for class 1 (event), proportional to the number of samples in each class. Here, the F1-score of class 0 is close to 1, and this class corresponds to more than 99.9% of the data, leading to a weighted F1-score close to 1.

6. DISCUSSION AND CONCLUSION

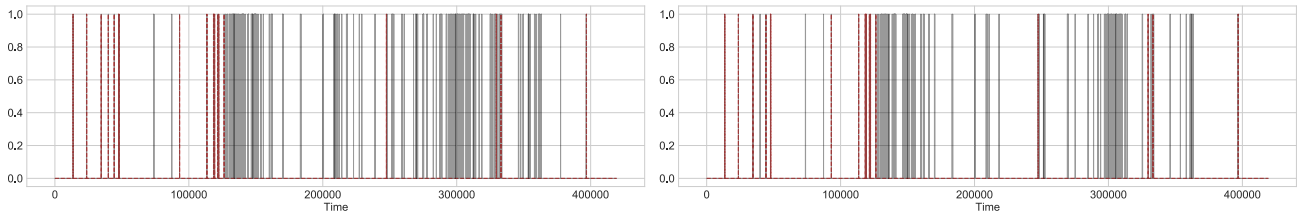
This case study was motivated by attempts to improve the predictive efficiency of a model developed to identify a disruptive and costly event in a mineral process plant. The data for the model is drawn from the process control system. An independent and trusted ground truth for the event is not available. This is not uncommon in industry situations today as the operators and maintainers have not necessarily worked in environments where data collection and quality are paramount considerations. However, today's asset managers cannot afford to wait for failure events to occur in order to build good models, so there is considerable interest in using available process control data to assist with predicting failures now and in the future. A key question is how to label events. This work considers two alternatives, the first is using a *rule-based* approach suggested by the company. This identifies a bogging event as "occurring when the rake was offline for a period between 1 and 60 minutes". The primary means of determining if a rake is offline is the motor current reading. Motor current and other sensors are recorded every minute. This approach identifies 150 events in the three year period. We demonstrate a data-driven approach using Dynamic Time Warping followed by Agglomerative Hierarchical Clustering. This groups events with similar event dynamics producing data set with about 64 events, depending on decisions associated with time windowing and other processing considerations. Both data sets are used as inputs to two models: a Logistic Regression and a Random Forrest. Grid-search strategy with cross-validation is used to estimate the hyper-parameters of each model. Two different strategies are used to handle the imbalance in the data.

Both models produce 'good' results for the test data in four month period after the model is built as shown in Figures 7a and 7b. However the model performance deteriorates as we move further away in time from when the model was built. This is not unexpected as mineral processing plants are subject to changes in ore type that impact process dynamics.

Results on the test sets from both models illustrate the challenges with the imbalanced data and the need to explicitly consider, and document, the scenarios and hyper-parameters used. This is illustrated in Tables 7-11. We suggest much greater transparency when publishing the steps taken in the data wrangling process to manage this risk. These issues are seldom discussed in the literature (Ferreira & de Sousa, 2020).

Increasingly in our work with industry data, we are finding ambiguity associated with labelling, hence the focus of this paper. This should not be a surprise as it is a human dependent task and the quality of the labels is dependent on many factors such as training, motivation, and the design and ease of use of data capture systems (Unsworth, Adriasola, Johnston-Billings, Dmitrieva, & Hodkiewicz, 2011; Molina, Unsworth, Hodkiewicz, & Adriasola, 2013). We note the challenge for data scientists, who are focused on finding a good model, in trying to assess how well the data and its labels are representative of the actual events. Problems with incorrect labelling quickly become apparent when models are deployed in the field on operating assets, but by then, it is too late and much time, and money has been wasted.

For instance, if we have incorrect labelling of the events in the anomaly detection scenario, we still can find a good model to predict these (incorrect) events, but are they really what we are looking for? The data analysis approaches are powerful tools, but we must consider the aim and nature of the problem. Therefore, the data scientists must work together with the engineers, as well the engineers should work together with the data scientists. We understand that this interaction is the way to find proper solutions to the problems. We discussed



(a) (b)

Figure 7. Event label (dashed red) vs model predictions (gray) in the test set for (a) *rule-based* response variable, (b) *data-driven* response variable.

in the paper the fault/anomaly detection problem. However, the extension to a multi-class response variable (multi-fault cases) could be achieved by defining more than two classes (fault/working) in the clustering step. For instance, the modelling could be performed using multinomial logistic regression.

The demonstration of a data-driven approach to data labelling as a means to both identify events and group events with similar dynamics is a step-forward in labelling. It allows for alternate set of response variables to be considered and supports an event-based discussion about what could be in or out of the data frame. This data-driven approach provides an alternative solution for industry situations where ground truth data is not available for model estimation, and yet these models are still necessary to better support today's operations. One of the wider aims of this paper is to support discussion on the general topic of labelling of industry data.

ACKNOWLEDGMENT

The authors acknowledge funding from the Australian Research Council Centre for Transforming Maintenance through Data Science (Industrial Transformation Research Program Grant No. IC180100030). Additionally, Melinda Hodkiewicz acknowledges funding from the BHP Fellowship for Engineering for Remote Operations. The authors acknowledge the support of Alcoa of Australia for assistance in providing working definitions of maintenance events and for making available the data utilised in this manuscript.

NOMENCLATURE

BA	Balanced Accuracy
BD	Balanced Data
DTW	Dynamic Time Warping
F1	F1-score
fn	number of false-negative
fp	number of false-positive
ID	Imbalanced Data
k	sliding window size
λ	penalisation parameter
LR	Logistic Regression
RF	Random Forest
s	step size
TNR	True-Negative Ratio
TPR	True-Positive Ratio
tn	number of true-negative
tp	number of true-positive

REFERENCES

Agogino, A., & Goebel, K. (2007). *Milling data set.* `http://ti.arc.nasa.gov/project/prognostic-data-repository`. (BEST lab, UC Berkeley. NASA Ames Prognostics Data Repository)

Astfalck, L., Hodkiewicz, M., Keating, A., Cripps, E., & Pecht, M. (2016). A modelling ecosystem for prognostics. In D. Larsen & K. Reichard (Eds.), *Proceedings of the annual conference of the prognostics and health management society 2016* (pp. 273–281). Prognostics and Health Management Society. (Annual Conference of the Prognostics and Health Management Society 2016 ; Conference date: 03-10-2016 Through 08-10-2016)

Cai, W., Zhao, J., & Zhu, M. (2020). A real time methodology of cluster-system theory-based reliability estimation using k-means clustering. *Reliability Engineering & System Safety*, *202*, 107045. doi: https://doi.org/10.1016/j.ress.2020.107045

Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., da P. Francisco, R., Basto, J. P., & Alcala, S. G. S. (2019). A sys-

tematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, *137*, 106024. doi: https://doi.org/10.1016/j.cie.2019.106024

Charrad, M., Ghazzali, N., Boiteau, V., & Niknafs, A. (2014). Nbclust: An R package for determining the relevant number of clusters in a data set. *Journal of Statistical Software, Articles*, *61*(6), 1–36. doi: 10.18637/jss.v061.i06

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, *3*(3), 32-57. doi: 10.1080/01969727308546046

Erhan, L., Ndubuaku, M., Di Mauro, M., Song, W., Chen, M., Fortino, G., ... Liotta, A. (2021). Smart anomaly detection in sensor systems: A multi-perspective review. *Information Fusion*, *67*, 64-79. doi: https://doi.org/10.1016/j.inffus.2020.10.001

Ferreira, H. M., & de Sousa, A. C. (2020). Remaining useful life estimation of bearings: Meta-analysis of experimental procedure. *International Journal of Prognostics and Health Management*, *11*(2). doi: https://doi.org/10.36001/ijphm.2020.v11i2.2922

Jung, Y., Park, H., Du, D.-Z., & Drake, B. L. (2003). A decision criterion for the optimal number of clusters in hierarchical clustering. *Journal of Global Optimization*, *25*(1), 91–111. doi: 10.1023/A:1021394316112

Kim, K., Parthasarathy, G., Uluyol, O., Foslien, W., Sheng, S., & Fleming, P. (2011, 08). *Use of SCADA Data for Failure Detection in Wind Turbines* (Vols. ASME 2011 5th International Conference on Energy Sustainability, Parts A, B, and C). doi: 10.1115/ES2011-54243

Lee, J., Qiu, H., Yu, G., & Lin, J. (2007). *Bearing data set.* https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository. (NASA Ames Prognostics Data Repository)

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, *104*, 799-834. doi: https://doi.org/10.1016/j.ymssp.2017.11.016

Li, H. (2015). On-line and dynamic time warping for time series data mining. *International Journal of Machine Learning and Cybernetics*, *6*(1), 145–153. doi: 10.1007/s13042-014-0254-0

Listou Ellefsen, A., Bjørlykhaug, E., Æsøy, V., Ushakov, S., & Zhang, H. (2019). Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety*, *183*, 240-251. doi: https://doi.org/10.1016/j.ress.2018.11.027

Michaud, D. (2015). *Rake classifier.* www.911metallurgist.com/blog/rake-classifier. 911 Metallurgist. (Accessed: 2021-Apr-08)

Molina, R., Unsworth, K., Hodkiewicz, M., & Adriasola, E. (2013). Are managerial pressure, technological control and intrinsic motivation effective in improving data quality? *Reliability Engineering & System Safety*, *119*, 26–34.

Muda, L., Begam, M., & Elamvazuthi, I. (2010). Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *Journal of Computing*, *2*(3). (arXiv:1003.4083v1)

Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. *WIREs Data Mining and Knowledge Discovery*, *2*(1), 86–97. doi: https://doi.org/10.1002/widm.53

Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., & Varnier, C. (2012). Pronostia: An experimental platform for bearings accelerated degradation tests. In *Ieee international conference on prognostics and health management, phm'12*. (pp. 1–8).

Reder, M., Yürüşen, N. Y., & Melero, J. J. (2018). Data-driven learning framework for associating weather conditions and wind turbine failures. *Reliability Engineering & System Safety*, *169*, 554–569. doi: https://doi.org/10.1016/j.ress.2017.10.004

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65. doi: https://doi.org/10.1016/0377-0427(87)90125-7

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on Acoustics, Speech, and Signal Processing*, *26*(1), 43–49. doi: 10.1109/TASSP.1978.1163055

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P. K., & Aroyo, L. M. (2021). "everyone wants to do the model work, not the data work: Data cascades in high-stakes ai". In *Human factors in computing systems (chi)* (pp. 1–15).

Saxena, A., & Goebel, K. (2008). Turbofan engine degradation simulation data set. *NASA Ames Prognostics Data Repository*, 1551–3203.

Steinley, D., & Brusco, M. J. (2007). Selection of variables in cluster analysis: An empirical comparison of eight procedures. *Psychometrika*, *73*(1), 125. doi: 10.1007/s11336-007-9019-y

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(2), 411–423. doi: https://doi.org/10.1111/1467-9868.00293

Unsworth, K., Adriasola, E., Johnston-Billings, A., Dmitrieva, A., & Hodkiewicz, M. (2011). Goal hierarchy: Improving asset data quality by improving motivation. *Reliability Engineering & System Safety*, *96*(11), 1474–1481.