Data-Centric Engineering (2022), 3: e11
doi:10.1017/dce.2022.4

RESEARCH ARTICLE

Managing streamed sensor data for mobile equipment prognostics

Toby Griffiths[1], Débora Corrêa[2,3,*], Melinda Hodkiewicz[1,3] and Adriano Polpo[1,2,3]

[1]System Health Lab, Department of Mechanical Engineering, University of Western Australia, Crawley, Western Australia 6009, Australia
[2]Department of Computer Science and Software Engineering, University of Western Australia, Crawley, Western Australia 6009, Australia
[3]ARC Industrial Transformation Training Centre (Transforming Maintenance through Data Science), University of Western Australia, Crawley, Western Australia 6009, Australia
*Corresponding author. E-mail: debora.correa@uwa.edu.au

Received: 07 October 2021; **Revised:** 26 January 2022; **Accepted:** 02 February 2022

Keywords: Asset health; LASSO logistic regression; mobile fleet prognostics; sensor data; Streaming data; time-series data

Abstract

The ability to wirelessly stream data from sensors on heavy mobile equipment provides opportunities to proactively assess asset condition. However, data analysis methods are challenging to apply due to the size and structure of the data, which contain inconsistent and asynchronous entries, and large periods of missing data. Current methods usually require expertise from site engineers to inform variable selection. In this work, we develop a data preparation method to clean and arrange this streaming data for analysis, including a data-driven variable selection. Data are drawn from a mining industry case study, with sensor data from a primary production excavator over a period of 9 months. Variables include 58 numerical sensors and 40 binary indicators captured in 45-million rows of data describing the conditions and status of different subsystems of the machine. A total of 57% of time stamps contain missing values for at least one sensor. The response variable is drawn from fault codes selected by the operator and stored in the fleet management system. Application to the hydraulic system, for 21 failure events identified by the operator, shows that the data-driven selection contains variables consistent with subject matter expert expectations, as well as some sensors on other systems on the excavator that are less easy to explain from an engineering perspective. Our contribution is to demonstrate a compressed data representation using open-high-low-close and variable selection to visualize data and support identification of potential indicators of failure events from multivariate streamed data.

Impact Statement

The industrial Internet of Things is providing an avalanche of data on our physical assets, but using this data to assess asset condition is still largely done manually by engineers and operators. Dealing with the velocity and volume of sensor data requires new approaches, and in this work, we demonstrate the use of the open-high-low-close (OHLC) to efficiently compress and visualize streamed data. The OHLC method results in a 96% reduction in data storage requirements while retaining important features of the underlying time series. A logistic regression model using L1 penalization is then used to identify sensor combinations related to historical failure events. The pipeline provides a data-driven basis for raising questions of the Heavy Mobile Equipment industry status quo and practices.

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

1. Introduction

With the rise of wireless streaming data, asset health visibility is greater than ever before. There are numerous sensors transmitting data on many different classes of equipment, particularly on mobile assets in high value industries such as mining. These data are revolutionizing remote health monitoring, enabling the transition to autonomy (Loureiro et al., 2014).

Production excavators are primary production units in fleets of heavy mobile equipment (HME) used in mining projects. Excavators sit at the start of a continuously operating supply chain, and any loss of availability has knock-on impacts to down-stream production. Any downtime associated with production excavators is costly to mining companies, and therefore effective scheduling of maintenance is key to maximizing availability. Excavators and other HME are equipped with a large number of sensors that wirelessly transmit time-series data to a central control station. With the ability to efficiently capture the condition of the asset given by this sensor data, maintenance can be scheduled more effectively to ensure that parts are replaced before failure. Waiting until failure incurs a greater loss in production time. Our goal is to see if the streamed data available to asset owners and exportable as comma-separated values can be postprocessed to inform maintenance through a compact and manageable representation of the actual condition of the asset.

Our contributions are to (a) present a transparent and replicable method that can be used to clean and prepare streamed sensor data using open-high-low-close (OHLC) to summarize large numerical sensor datasets retaining their important time-domain characteristics and (b) demonstrate a data-driven variable selection model based on the LASSO regularization (Hastie et al., 2019) with a lagged data frame to identify covariates related to faults on a specific subsystem (the hydraulic system). The manual selection of such variables to monitor requires subject matter expertise, and automation is required when there are many HME, each with multiple subsystems. Each HME asset has hundreds of sensors of which a subset (~ 100) is available to the asset owners as a streamed dataset.

The paper proceeds as follows: Section 2 provides an overview of the related work, Section 3 describes the data sources and their limitations, whereas Section 4 presents the steps conducted for preparing the data for the methodology presented in Section 5. Section 6 assesses the logistic regression for variable selection model. Finally, Section 7 explores the assumptions underlying the processing and results, highlighting areas for future work.

2. Related Literature and Current Limitations

The availability of time-series datasets from many sensors, each measuring the condition of an asset from a different perspective, has resulted in an explosion of interest in the application of machine learning models for decision-making support in predictive maintenance (Angelopoulos et al., 2020). However, unbalanced and poorly annotated failure events remain a key challenge (Unsworth et al., 2011; Dalzochio et al., 2020; Correa et al., 2022). Many assets have high reliability, and therefore critical failures are rare. An assessment of damage extent and failure mode is only determined when the machine is removed from service and “opened up” for repair by the maintenance technicians. Chronic failures and maintenance work resulting from on-condition maintenance work notifications are also poorly recorded in maintenance texts, making it difficult to match patterns in the data to actual damage on the machine. As a result, the availability of well-annotated failure datasets based on real industry case studies is very limited (Sikorska et al., 2016), and there is considerable uncertainty over what is ground truth for the response variable in many real industrial datasets. With mobile machines, the industry relies on fault codes, selected by the machines’ operator and stored in the fleet management system database, as the response variable for predictive analytics work. As with all manual diagnostic processes, the quality and consistency of the codes selected depend in part on the training and experience of the operator as well as their motivation toward the work (Unsworth et al., 2011).

Other open issues include the quality of data for explanatory variables with streamed data often having missing values and ill-conceived data structures (Hegedűs et al., 2018). Modern machines are starting to use

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

edge computing, allowing for preprocessing data at the sensor or locally on the machine (Kwon et al., 2016). However, HME are long life, and there is significant investment in existing fleets of older machines that predate modern edge computing developments. Existing machines are not fitted with edge capability; instead, data from sensors are sampled, processed, and streamed to a centralized data storage unit of the machine.

While there is considerable interest in time-series modeling for prediction, and the interested reader is referred to surveys on these (Angelopoulos et al., 2020; Dalzochio et al., 2020), our interest is in the manipulation of the streaming data and variable selection ahead of any downstream diagnostics or prognostics process. We are also aware of the reluctance of many engineers to trust black box models (Phillips et al., 2015), and therefore providing ways of visualizing the data in an efficient way is a stepping stone to using it in decision support models. This need to visualize large volumes of streamed data in an efficient way has also been of interest to analysts in financial markets. Candlestick charts are intuitive and widely used to understand and widely used for monitoring price movements of financial products (Lee and Jo, 1999). There is a significant literature on methods to analyze the OHLC data in candlestick charts in finance (Romeo et al., 2015), but only limited references to the use of OHLC data have been located in the condition monitoring and Internet-of-Things literature (Aleman et al., 2018).

In time-series data, observations from different sensors are often taken from many units (e.g., in a health data, we can have a number of patients). This is not an adequate assumption for streamed data on mobile machines where sensors are all on the same unit, albeit some on different subsystems. Taking into account the temporal relation of the sensors to the response variable requires further transformation, which has been done, for instance, with the use of distributed lag models (Bentzen and Engsted, 2001; Belloumi, 2014). Distributed lag models include previous values of explanatory variables in time-series data to account for temporal dependence (Haugh and Box, 1977). Lag models have been previously used in the prediction of turbocharger failures (Moura et al., 2011), and for prediction of the current state of the *same* variable given its previous values.

3. Data

The dataset used in this project was sourced from industry, and is typical of streamed sensor data from mobile machines. To build the dataset, we used two data sources: streamed sensor data and codes from the fleet management system. The dataset covers the period of 9 months, from January 2019 to September 2019, and contains over 45-million rows of raw data available for one excavator across the 9 months. The sensor data contains 98 variables wirelessly streamed from the unit and stored on a structured query language (SQL) database within the company. Of the variables, 58 are represented numerically (e.g., pilot pump pressure), and the remaining 40 are binary variables, referred to as indicators (e.g., tension indicators). These indicators give a value of 0 when not active and 1 when their criteria are met. The other data source is the fleet management system. This is used by the machine operator to report the status of their machine using a set of codes, for example, loading, idling, tramming, and so on. Others of these status entries correspond to maintenance events, showing the start and end times when the unit is unavailable due to scheduled and unscheduled maintenance work. Of particular interest to this work are the fault codes entered by the machine operators such as “Hydraulics,” representing an unscheduled hydraulic maintenance event. The selection of a particular code involves the synthesis of observations by the operator from both the monitoring data available in the cab and how the machine is performing. Code selection is therefore very dependent on the training and experience of each individual operator. Fault codes influence the follow-up action, and serious faults need the attention of maintenance personnel.

3.1. Data structure and challenges

The structure of the sensor data in its original form is known as long-form data (Everitt and Hothorn, 2011). There is a column displaying time stamps, a “key” column to identify the sensor each row refers to, and another to specify the value for that sensor at the specified time. An extract of the data is presented in Table 1. The sensor-recording intervals differ between sensors, with some polling values as often as every

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

Table 1. *An extract from the raw sensor data showing the structure and asynchronous sensor recording.*

Index	Time stamp	Sensor	Value
1	19/01/2019 01:31:25	1	38
2	19/01/2019 01:31:25	2	27.41
3	19/01/2019 01:31:26	3	64
4	19/01/2019 01:31:55	1	38
5	19/01/2019 01:31:55	2	27.36
6	19/01/2019 01:31:56	3	65
7	19/01/2019 01:32:12	2	27.36
8	19/01/2019 01:32:25	1	38
9	19/01/2019 01:32:26	3	65
10	19/01/2019 01:32:55	1	38
11	19/01/2019 01:32:56	3	66
12	19/01/2019 01:33:25	1	38
13	19/01/2019 01:33:25	2	27.41
14	19/01/2019 01:33:26	3	66

5 s, and others every 30 s or 1 min. Furthermore, the polling times are not synchronized between sensors, even if they are polling at the same interval. For example, two sensors may both be set to record at 30-s intervals, and whereas one may record at 25 and 55 s past each minute, the other may record after 26 and 56 s. Table 1 shows an extract of the raw data where asynchronous recording is evident. Two minutes of data from three sensors are presented, each recording at 30-s intervals. From rows 1 to 6, it is clear that Sensors 1 and 2 record at the same time each minute, and Sensor 3 records a second later. Sensors 1 and 3 consistently record at a 30-s interval throughout the extract; however, Sensor 2 records after only 17 s in row 7, and does not record again for 73 s.

Another complication of this sensor data is in periods where no data are recorded for certain sensors. For one machine, there are four gaps of greater than 60 days in the sensor values, presumably where the machine has been powered down or unable to send data to the SQL database for other reasons. There are also many shorter periods of missing data, with a total of 57% of time stamps across the 9 months for one excavator containing missing values for at least one sensor. An example of a period of missing data is highlighted in red in Figure 1.

Examples of states that may be recorded are waiting, loading, scheduled maintenance, unscheduled maintenance, and hydraulics-related events. Each event or change of state has an associated time stamp. These time stamps must be matched between the fleet management records and the streamed sensor data. This matching of time stamps between the data sources is not straightforward. The time stamps do not explicitly match, but rather, the time of sensor value recording often falls in an interval between the start time and end time of an operating state recorded.

4. Data Preparation

In this section, we describe the steps required to achieve the data frame presented in Table 2 that will be used as input to the logistic regression model. The preparation is performed in six steps: (a) data wrangling (Wickham and Grolemund, 2017), (b) integration of event data, (c) data summary by OHLC, (d) data reduction of highly correlated sensors and events (events happening in a [predefined] short-time interval), (e) preparing lagged data, and (f) removal of missing data. We describe each of these steps below.

Before the data preparation, the dataset contained 58 numerical data (sensors) and 40 binary indicators (alarms). Two sensors and 20 alarms were removed from the analysis on the advice of engineers as they

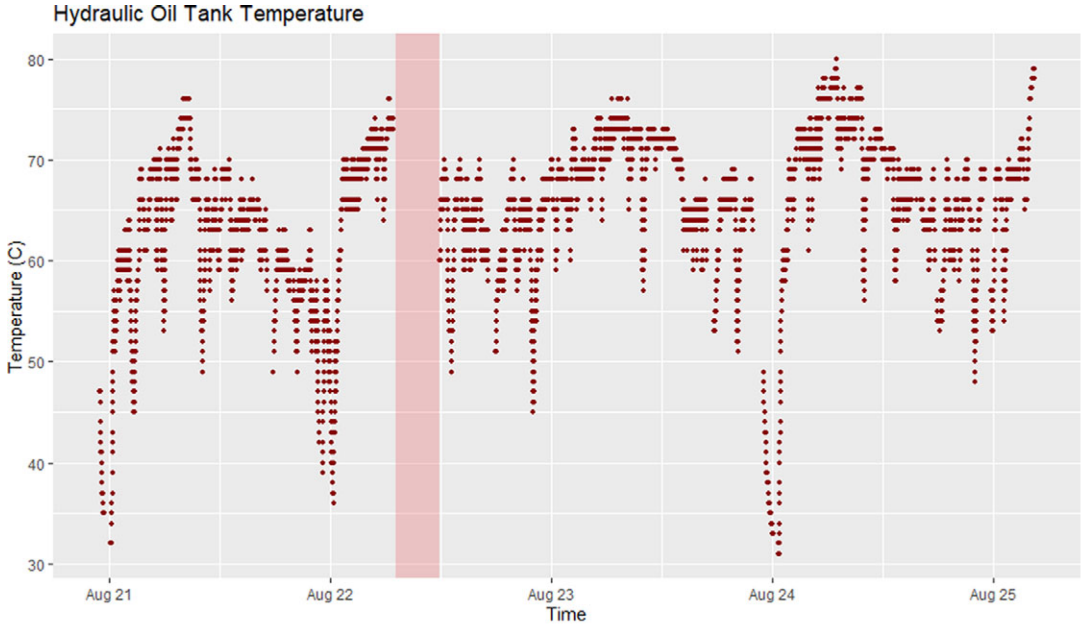

Figure 1. *An extract of the raw data, with a region of missing data highlighted by the vertical red line.*

Table 2. *The proposed baseline data frame. After removing rows containing Not a Number (NAN) values, the final dimension of the data is 3,591 × 35. The total number of events is 31 for the response variable given by the fleet management data.*

Time	S1	S2	…	S6:9(mean)	S10	…	A1	A2	…	A40	Event
2019-01-09 11:03:00	47.0	27.32	…	52.5	557.0	…	1	1	…	0	0
2019-01-09 12:03:00	NAN	NAN	…	NAN	NAN	…	NAN	NAN	…	NAN	0
2019-01-09 13:03:00	NAN	NAN	…	NAN	NAN	…	NAN	NAN	…	NAN	0
2019-01-09 14:03:00	47.0	27.31	…	53.00	556.0	…	0	0	…	0	0
2019-01-09 15:03:00	48.0	27.36	…	66.75	496.0	…	0	1	…	0	0
2019-01-09 16:03:00	43.0	27.26	…	76.75	504.0	…	0	1	…	0	0
2019-01-09 17:03:00	41.0	27.31	…	71.50	508.0	…	1	1	…	0	0
2019-01-09 18:03:00	42.0	27.36	…	65.50	504.0	…	0	0	…	0	0
2019-01-09 19:03:00	44.0	27.41	…	66.25	508.0	…	0	0	…	1	1
2019-01-09 20:46:00	37.0	27.31	…	57.75	256.0	…	1	1	…	0	0
2019-01-09 21:46:00	NAN	NAN	…	NAN	NAN	…	NAN	NAN	…	NAN	0
…	…	…	…	…	…	…	…	…	…	…	…
2019-01-10 02:46:00	49.0	27.21	…	76.00	504.0	…	1	0	…	0	0
2019-01-10 03:46:00	50.0	27.26	…	65.50	508.0	…	1	0	…	0	1
2019-01-10 10:57:00	43.0	27.36	…	62.00	504.0	…	0	0	…	1	0
2019-01-10 11:57:00	50.0	27.31	…	69.00	492.0	…	0	0	…	1	0
…	…	…	…	…	…	…	…	…	…	…	…

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

are used for operational purposes, such as fuel level, and make no contribution to asset health estimation. Other examples include auto-idle switch status and stairway position indicator. Another six alarms, such as valve status and engine overrun, were never triggered in this data (they only contained zeros), and therefore were also removed from the analysis.

4.1. Data wrangling

The first step of the data wrangling is to prepare the data in a format known as wide form, where each variable/sensor is separated into unique columns, with a time stamp column identifying each row. Wide format data facilitate the aggregation or manipulation of variables, and it is a good practice in data science (Kotu and Deshpande, 2018). The inconsistencies between recording times of different sensors and consecutive recording times of the same sensor make it difficult to convert to wide form data. When rearranging the data into a wide format, there are many empty values because each row corresponds to a particular time that only a few sensors have recorded values for. The wide form of the extract from Table 1 is shown in Table 3, with many missing values evident. The missing data unnecessarily increase the size of the data when converted. The dataset is already very large, with over 45-million rows of raw data available for one machine over the 9 months. Therefore, any unnecessary data size should be avoided to ensure that any storage and analysis can be performed efficiently.

To wrangle the sensor data into a wide form data frame, we first need to truncate all time stamps to the nearest minute in order to align them across all variables. In cases where the sensor is recorded multiple times within the minute, the first result is taken. This method conserves the real data, as each entry still corresponds to an actual measurement taken, rather than taking mean values and changing the original data. The data can then be pivoted to a wide format. The results of applying this process to the sensor data extract from Tables 1 and 3 are shown in Table 4. The sensor recordings are now aligned at the same time stamps, truncated to each minute.

Table 3. *The sensor data extract presented in Table 1 after pivoting, showing several unnecessary missing values.*

Index	Time stamp	Sensor 1	Sensor 2	Sensor 3
1	19/01/2019 01:31:25	38	27.41	NA
2	19/01/2019 01:31:26	NA	NA	64
3	19/01/2019 01:31:55	38	27.36	NA
4	19/01/2019 01:31:56	NA	NA	65
5	19/01/2019 01:32:12	NA	27.36	NA
6	19/01/2019 01:32:25	38	NA	NA
7	19/01/2019 01:32:26	NA	NA	65
8	19/01/2019 01:32:55	38	NA	NA
9	19/01/2019 01:32:56	NA	NA	66
10	19/01/2019 01:33:25	38	27.41	NA
11	19/01/2019 01:33:26	NA	NA	66

Table 4. *The sensor data extract after preprocessing steps and aligned with event data.*

Index	Time stamp	Sensor 1	Sensor 2	Sensor 3	Event
1	19/01/2019 01:31:00	38	27.41	64	0
2	19/01/2019 01:32:00	38	27.36	65	0
3	19/01/2019 01:33:00	38	27.41	66	1

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

4.2. Integrating event data

To relate the sensor data to the event of interest (hydraulics failure, in our case), columns corresponding to the status in the fleet management data are appended. These columns contain values of 1 when the machine is in that status, and 0 otherwise. The start time of each event is truncated to the minute level, and the data are padded to include a row for each minute between the first and last entries. This creates many empty rows, which are filled by extending the status columns down. The result is a continuous value of 1 while that status is active. Next, the data are grouped by time stamp and the maximum value of each column taken. This ensures that there is only a single row for each minute of time, and that in cases where two statuses are active in 1 min (as the machine moves between statuses), they both show a value of 1 in that row. The manipulated fleet management data are joined to the cleaned and pivoted sensor data based on matching the time stamp columns as represented by the column "Event" in Table 4, which have all been truncated to the minute in which the observations occur.

4.3. Data summary with OHLC

The sensor data are summarized by adapting the method behind financial OHLC charts. These charts are used to show price variations of financial measures, such as stock prices, in a specified time window. The open and close values together show the direction of the price movement within the window, and the high and low values give an indication of volatility (Gregory, 2006). The sensor data are manipulated by taking the first and last values of each sensor, as well as the maximum and the minimum for each interval period, similar to the manner in which price data are manipulated to create OHLC charts. This allows the key trends of the data to be represented with less entries. A period of 1 hr is chosen as it is expected that a failure will show symptoms multiple hours before it occurs. Thus, for each hour, each sensor is described by four values, given the first and last values of each sensor, as well as the maximum and the minimum for each 1-hr period. Then, another decision in the process is which OHLC information we will use. We have chosen to represent the value of the sensor as the *close* information in OHLC values, as it captures the signature of the data in the last hour.

The raw data, due to being recorded at high frequency (as little as 5-s increments between recordings), take a large amount of storage (2.5 GB for a single machine). Converting the data to OHLC values over a longer interval retains important features of the data while reducing the storage required. Figure 2 shows a snapshot of 5 days of data, consisting of 14,402 data entries. The raw data are shown on the bottom, and the OHLC data are shown in a candlestick plot on the top. A candlestick plot represents OHLC data with the open and close values at the ends of the candle "body" and the high and low values indicated by the "wick." The color of the candle is red if the close value is lower than the open, and green if the close value is higher. The figure shows that by using candlestick plots, the OHLC data can represent the important features of the raw data. The OHLC data of the same period only require 600 entries, saving 96% of space.

Candlestick plots allow the data to be quickly interpreted visually and ensure that although some resolution may have been lost in the conversion, the key trends remain clear. Furthermore, if there are missing data within the interval, the OHLC method will ignore it in the summary, resulting in clearer visibility of some events.

As mentioned before, the variables in this analysis are a combination of readings from sensors, presented as continuous variables, and indicators, which are binary variables. In the preprocessing of the data, the numerical sensors are manipulated using the OHLC method; however, this is not applied to the binary indicators. Summarizing them in the same way would not add value, as the range of possible values only includes two possibilities. Instead, the maximum value for the interval is taken. We use a value of 1 for the specified OHLC interval if the alarm was active at any time within the interval. The same is applied for the event data inside that hour. The same process for the alarms was used to define the response variable. Then, the response variable represents if an event has occurred in that OHLC interval. There may be instances where it is valuable to see how long an indicator has been active for within the interval. In this way, a fraction of time in the interval that the indicator was active may be used in future analysis.

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

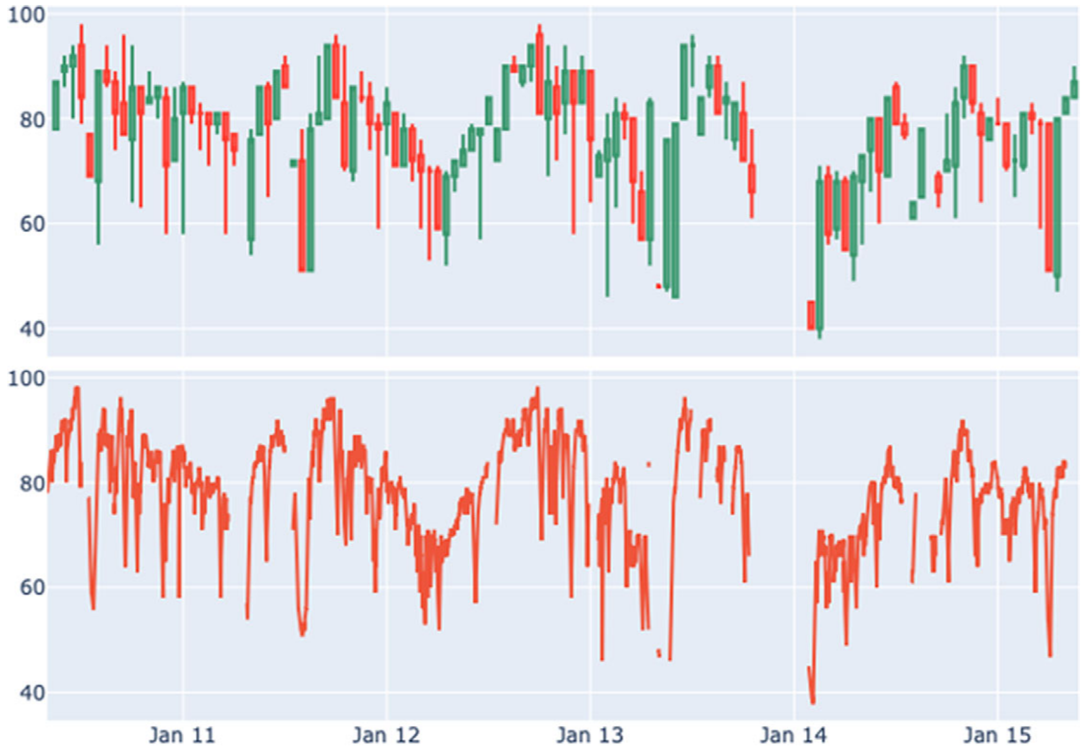

Figure 2. *The open-high-low-close data (top) retain the important features of the raw data (bottom).*

4.4. Correlated sensors and correlated events

We noticed that some sensors are highly correlated. For instance, there are 16 sensors measuring exhaust temperature of the left engine, and 16 sensors measuring exhaust temperature of the right engine. There are four sensors measuring intake manifold temperature of the left engine, and four sensors measuring intake manifold temperature of the right engine. We computed the Pearson correlation of these sensors, and we found a correlation greater than 90% for all pairs in a subsystem. We decided, together with the Engineers, that a good strategy would be to further reduce the data dimension by taking the average value of each of these groups of sensors.

We also need to make an assumption about when to consider consecutive events as part of one unique event or as part of different events. We established an interval of 8 hr to be indicative of different events, which was informed by an engineer. Therefore, if the time difference of two consecutive events is lower than 8 hr, these events are grouped together as the same event. We keep the information of the first event in the group and use its calendar time to define when the event occurred (value of 1 in the column *Event* of Table 2).

4.5. Preparing lagged data

To consider the effect of the dependence between rows of the time-series sensor data, data from multiple hours before each event are considered. To do this, lagged variables are created from each sensor including values of this sensor from previous time steps leading up to a response variable event (Haugh and Box, 1977). One of the company aims was to identify the features that could predict the event. In this analysis, it was considered a period of 5 hr before the event, but not including the features at the event time. The aim is to select features that are useful indicators for predicting failure. A value of $\ell = 5$ (lag)

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

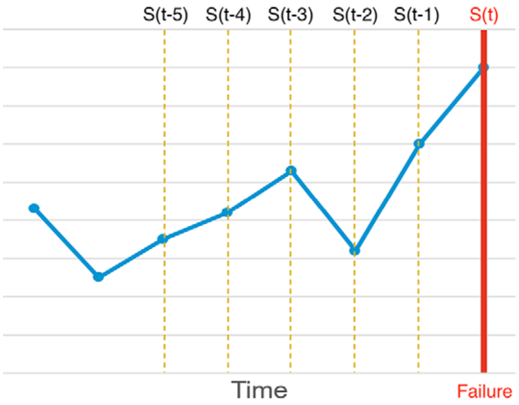

Figure 3. *A representation of a lag data model for one sensor, using a window of 5 hr. The red line indicates the time of failure. The last 5 hr of each sensor before the failure are used as variables in the lagged model. As we do not use overlap between the windows, this process results in a data frame with dimension 1,093 × 107, which is further reduced to 581 × 107 after removing missing rows.*

will already generate 107 covariates (19 sensors × 5 plus 11 alarms plus the "Hour Meter"), so we avoided larger ℓ values to minimize the effect of overfitting. Figure 3 shows a representation of how an observation is created in the lag model from the values of one sensor considering $\ell = 5$ hr. The lag model allows the five most recent instances of each sensor to be considered as input variables that can be used to model the binary failure state. The lagged data are considered for the sensor features (continuous). For the case of alarm features (binary), we have considered a window of ℓ hours (the same ℓ as before). The value of the alarm in this window is equal to 1 if the alarm was equal to 1 at any moment during the window time.

4.6. Missing data

The next step consists of removing any row of the data frame containing missing data. There are many approaches to dealing with missing data, some of which may be applied to this problem in the future. Replacing missing data may be possible: for the continuous variables, interpolation or extrapolation of values is an option. The missing data in this use case do not occur randomly in different variables, but often across many variables simultaneously for a significant amount of time. This is due to the machine powering down in some instances and could be replicated on other areas of missing data where the machine is idling or powered down, using a measure of machine running time rather than a date and time as was used in this analysis. About 35% of the rows in Table 2 contained missing data.

5. Logistic Regression Model

In this section, we define the logistic regression model used, including the mathematical details on how we handled with the lagged data and the high-dimensional problem.

Consider $t \in \{1,2,3,\ldots\}$, in hours, as a time index, $Y(t)$ as binary random variable, with $Pr(Y(t)=1|\boldsymbol{x}_{\boldsymbol{\ell}}(\boldsymbol{t})) = p(\boldsymbol{x}_{\boldsymbol{\ell}}(\boldsymbol{t})|\boldsymbol{\theta})$, the lag $\ell \in \{1,2,\ldots\}$, the parameter vector $\boldsymbol{\theta} = \{\beta_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{\boldsymbol{\ell}}\}$ in dimension of $(k \times \ell + 1)$, $\boldsymbol{\beta}_{\boldsymbol{j}} = \left\{\beta_{j1}, \ldots, \beta_{jk}\right\}$, $j = 1, \ldots, \ell$, $\boldsymbol{x}_{\boldsymbol{\ell}}(\boldsymbol{t}) = \{\boldsymbol{x}(\boldsymbol{t}-\mathbf{1}), \ldots, \boldsymbol{x}(\boldsymbol{t}-\boldsymbol{\ell})\}$, and $\boldsymbol{x}(\boldsymbol{t}) = \{x_1(t), \ldots, x_k(t)\}$ as a vector of k features at time t, for each t. Note that, $\boldsymbol{x}(\boldsymbol{t})$ are known values (fixed/given) $\forall t$. To handle with the imbalanced data, we consider the weights $w_1 = n/(2n_1)$ and $w_0 = n/(2n_0)$, that are the contributions in the model to each observed $y(t)$ equal to 1, and 0, respectively. In this case, $n = n_1 + n_0$ is the total number of samples, n_1 is the number of events, $n_1 = \sum_t y(t)$, and n_0 is the number of nonevents.

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

Define

$$p(\boldsymbol{x_\ell}(\boldsymbol{t})|\boldsymbol{\theta}) = \frac{\exp\left(\beta_0 + \sum_{j=1}^{\ell} \boldsymbol{\beta_j} \boldsymbol{x}(\boldsymbol{t}-\boldsymbol{j})'\right)}{1+\exp\left(\beta_0 + \sum_{j=1}^{\ell} \boldsymbol{\beta_j} \boldsymbol{x}(\boldsymbol{t}-\boldsymbol{j})'\right)}$$

and the loss function

$$\text{Loss}(\boldsymbol{\theta}|\mathcal{D}) = -\sum_t \{w_1 y(t)\log[p(\boldsymbol{x_\ell}(\boldsymbol{t})|\boldsymbol{\theta})] + w_0(1-y(t))\log[1-p(\boldsymbol{x_\ell}(\boldsymbol{t})|\boldsymbol{\theta})]\}/n, \tag{1}$$

where $\mathcal{D}$ is the available data. That is, the observed values $y(t)$ of $Y(t)$ and $\boldsymbol{x}(\boldsymbol{t})$, $\forall t$. Under this construction, we have $(k\times\ell+1)$-dimensional parameter space, which in our study case we have 98 features (sensors/alarms), leading us to a high-dimensional problem. To handle with this, we adopted the LASSO logistic regression (LASSO-LR). The LASSO-LR is a popular log-linear classifier that incorporates L1 penalization. Then, considering the L1 penalization, our loss function is

$$\text{Loss}_\lambda(\boldsymbol{\theta}|\mathcal{D}) = -\sum_t \{w_1 y(t)\log[p(\boldsymbol{x_\ell}(\boldsymbol{t})|\boldsymbol{\theta})] + w_0(1-y(t))\log[1-p(\boldsymbol{x_\ell}(\boldsymbol{t})|\boldsymbol{\theta})]\}/n + \lambda\sum_{j=1}^{\ell}\sum_{h=1}^{k}|\beta_{jh}|.$$

To obtain the estimates of βs, we find the values of θ that minimizes the loss function $\text{Loss}_\lambda(\boldsymbol{\theta}|\mathcal{D})$.

LASSO-LR is applied to our data frame in order to select the most influential variables to describe hydraulic failure events. This uses the fleet management for the response variable, and the sensor data as explanatory variables. Using different values of λ will result in a different number of β values returning as 0. With $\lambda = 0$, the expression is traditional logistic regression (with no penalization), fitting β values for all variables. Increasing values of lambda will decrease the number of variables, returning other β values as 0. We tune this parameter using the training set as described in Section 5.1.

To compute a confidence interval for the parameters, we use the nonparametric bootstrap technique. However, when using a lag model ($\ell = 5$), for instance, we have a binary response variable where the number of observed events (total of times that $Y(t) = 1$) is only 3.75% of the event data for the fleet management. That is, we have an imbalanced dataset, and in this case, we have used a block bootstrap technique, where we consider two blocks: (a) when $Y(t) = 0$, we take a sample with replacement of the same size of the block, and (b) when $Y(t) = 1$, we take a sample with replacement of the same size of the block. Then, we aggregate these two blocks to have a resample corresponding to the entire data frame. Our bootstrap samples will keep the property of original data, with 3.75% for the fleet management being from the class $Y(t) = 1$. The confidence interval of 95% is taken as usual from a bootstrap, where we considered the 2.5% percentile and the 97.5% percentile of the estimated distributions of the estimators of the parameters.

5.1. Model evaluation

The primary reason for conducting this regression analysis is for variable selection, so we need to define a way to select the regularization parameter. We used a Grid-Search strategy with $k-$fold cross-validation to estimate the regularization parameter λ. For each value of λ, the Grid-Search fits the model using data in the $k-1$ sets and calculates the performance of the model for the validation dataset given by the remaining set. We adopted $k = 5$. The value of λ is chosen as the one providing better performance for the validation set.

Finally, given the estimates of the parameters, we compute the odds ratio of a feature. Since we are using a logistic regression model, the odds ratio are easily computed by taking the exponential of the βs, that is, $\exp(\beta_{jh})$ is the odds ratio of the feature associated with the parameter β_{jh}, $j = 1,\dots,\ell$, and $h = 1,\dots,k$. For a binary feature (alarm), the odds ratio represents the chance of an event (hydraulic failure) occurs given that the alarm is 1 (active) relative to when the alarm is 0. For instance, consider the alarm “Pump Contamination Indicator,” and that the odds ratio related to this alarm is 0.5. This implies

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

that the chance of a failure occurring when the alarm is on is half of the chance of the failure occurring when the alarm is off. On the other hand, if the odds ratio is equal to 2, then the chance of failure when the alarm is on is twice the chance when the alarm is off. For continuous features (sensors), the odds ratio is measured in the difference of one unit. Similarly to the previous example, if the odds ratio is 0.5, then for each increase of one unit in the sensor value, this reduces the chance of failure by 50%. When the odds ratio is 2, then it will double the chance of failure (for each unit increase in the sensor).

6. Results

In this section, we explore the performance of the model using the fleet management data as the response variable. When the fleet management system displays the code "Hydraulics," it is assumed to have experienced a failure in the hydraulic system at that time or that the hydraulic system needs maintenance. These status assignments originate from the machine operator's diagnosis of a fault, so there is a risk of human error attached to the code selection.

We apply data standardization using the strategy of Z-score normalization (sensors values are rescaled to have zero mean and unit standard deviation). The β values and odds ratios resulting from LASSO-LR for feature selection are presented in Table 5 using the fleet management response variable.

The results presented in Table 5 need some care in interpretation as follows.

- Only variables for which the odds ratio confidence interval does *not* include the value 1 are regarded as significant. This includes Ambient Air Temperature at (Lag 4: 4 hr before the event), Hydraulic Oil Tank Temperature (Lag 1: 1 hr before the event), and so on.
- Variables considered significant and increase the chance of a hydraulic fault are shown in blue, and those that decrease the chance of a hydraulic failure are shown in red.
- Although the OHLC can compress and keep some important information about the data, we have used only the close (C) value to build our model. Adding another measure as high (H) or low (L) would add 19 features for each lag in the model, which already has 107 features for a sample size of 581 observations.

We asked the site and reliability engineers and maintenance planners to select features they consider as important indicators of a hydraulic failure. The features selected, shown in bold on the Table 5, are four sensors (ambient air temperature, hydraulic oil tank temperature, pilot pump pressure, and the two pump transmission oil level alarms) and two alarms (hydraulic oil overheat indicator and pump contamination indicator). To better understand the engineers' selection, consider that the hydraulic system has a functional system and physical system. The functional system comprises all components required to deliver hydraulic oil to various elements on the excavator (see Figure 4). The physical system shown in Figure 5 illustrates how the hydraulic pumps are physically connected to the engines (which provide the torque) through gearboxes in the pump transmission system.

The results suggest the following:

- As ambient air temperature starts to rise, we would expect an increase in the chance of a hydraulic system failure as oil lubrication properties are adversely affected by high and low temperatures. Note that while Lag 4 is shown in blue, the lower bound of the 95% odds ratio confidence interval for Lag 1 and Lag 2 is quite close to 1. Elevated ambient air temperature is also indicated as important by the engineers.
- The engineers identified an increase in hydraulic oil tank temperature as being important. Paradoxically, while the model also identifies this as being important, it suggests that an increase in temperature is likely to cause a decrease in the chance of hydraulic failure. However, as in any model, we must be careful with the analysis of the parameters individually. The model is built with all features together, and the estimated values of the parameters are not independent. That is, the hydraulic tank oil temperature is showing a decrease on the chance of hydraulic failure in the

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

Table 5. *βs, odds ratio, and 95% confidence interval for the odds ratio from LASSO logistic regression variable selection with the fleet management response variable.* $\mathrm{Loss}(\boldsymbol{\theta}|\mathcal{D}) = 211.85,\ \lambda = 10.$

	Lag 1		Lag 2		Lag 3		Lag 4		Lag 5	
Sensor	β	Odds ratio	β	Odds ratio	β	Odds ratio	β	Odds ratio	β	Odds ratio
Ambient air temperature	0.29	1.34 (0.9,1.89)	0.18	1.19 (0.8, 1.81)	0.04	1.04 (0.74, 1.45)	0.77	2.16 (1.28, 2.77)	−0.03	0.97 (0.64, 1.65)
Hydraulic oil tank temperature	−0.69	0.50 (0.36, 0.88)	−0.45	0.64 (0.48, 1.02)	0.33	1.39 (0.95, 1.98)	−0.07	0.93 (0.59, 1.61)	0.25	1.29 (0.78, 1.96)
Engine coolant temperature of the left engine	0.03	1.03 (0.75, 1.33)	0.25	1.29 (0.80, 1.8)	0.30	1.36 (0.86, 1.79)	−0.09	0.91 (0.68, 1.20)	−0.22	0.81 (0.68, 1.12)
Intake manifold temperature (left)	0.04	1.04 (0.69, 1.44)	0.35	1.42 (0.96, 1.76)	0.20	1.22 (0.87, 1.64)	−0.11	0.90 (0.70, 1.19)	0.07	1.07 (0.82, 1.42)
Engine oil pressure of the left engine	−0.05	0.96 (0.75, 1.22)	0.10	1.11 (0.93, 1.28)	−0.09	0.91 (0.75, 1.04)	0.28	1.32 (1.00, 1.49)	0.18	1.19 (0.94, 1.43)
Engine oil temperature of the left engine	0.01	1.01 (0.97, 1.06)	0.03	1.03 (0.99, 1.08)	0.11	1.11 (1.05, 1.16)	0.04	1.04 (0.99, 1.10)	0.03	1.03 (0.96, 1.13)
Engine speed of the left engine	0.22	1.24 (0.88, 1.50)	0.17	1.18 (0.99, 1.33)	0.02	1.02 (0.75, 1.24)	0.28	1.32 (1.04, 1.46)	0.38	1.46 (1.08, 1.62)
Boost pressure of the left engine (left bank)	0.33	1.39 (1.09, 1.60)	−0.02	0.98 (0.83, 1.20)	0.01	1.01 (0.82, 1.29)	−0.11	0.89 (0.68, 1.17)	−0.31	0.73 (0.63, 0.95)
Boost pressure of the left engine (right bank)	0.11	1.11 (0.92, 1.34)	0.03	1.03 (0.86, 1.22)	0.11	1.11 (0.90, 1.35)	−0.16	0.85 (0.71, 1.10)	−0.36	0.69 (0.61, 0.93)
Exhaust temperature of the left engine	−0.23	0.79 (0.62, 1.06)	0.04	1.04 (0.82, 1.30)	0.13	1.14 (0.93, 1.30)	−0.30	0.74 (0.61, 0.99)	−0.07	0.93 (0.73, 1.27)
Engine coolant temperature of the right engine	−0.01	0.99 (0.78, 1.28)	0.21	1.23 (0.89, 1.65)	0.22	1.25 (0.97, 1.52)	0.15	1.17 (0.92, 1.48)	0.05	1.05 (0.83, 1.38)
Intake manifold temperature (right)	−0.18	0.84 (0.74, 1.10)	0.23	1.26 (0.92, 1.60)	−0.08	0.93 (0.77, 1.36)	−0.26	0.77 (0.68, 1.16)	0.29	1.34 (1.00, 1.80)

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

Table 5. *(Continued)*

	Lag 1		Lag 2		Lag 3		Lag 4		Lag 5	
Sensor	*β*	Odds ratio	*β*	Odds ratio	*β*	Odds ratio	*β*	Odds ratio	*β*	Odds ratio
Engine oil pressure on the right engine	−0.33	0.72 (0.59, 0.99)	0.31	1.36 (1.02, 1.55)	−0.24	0.79 (0.63, 1.00)	0.17	1.18 (0.93, 1.39)	0.06	1.06 (0.84, 1.33)
Engine oil temperature of the right engine	−0.07	0.93 (0.84, 1.11)	0.04	1.04 (1.01, 1.12)	−0.10	0.91 (0.82, 1.14)	−0.08	0.93 (0.85, 1.10)	0.05	1.05 (1.03, 1.11)
Engine speed of the right engine	0.04	1.04 (0.81, 1.29)	0.01	1.01 (0.91, 1.17)	−0.12	0.88 (0.73, 1.08)	0.06	1.06 (0.93, 1.23)	0.30	1.35 (1.09, 1.48)
Boost pressure of the right engine left bank	0.01	1.01 (0.91, 1.21)	−0.10	0.90 (0.75, 1.13)	−0.22	0.81 (0.69, 1.08)	−0.11	0.90 (0.76, 1.07)	−0.11	0.89 (0.76, 1.12)
Boost pressure of the right engine right bank	0.13	1.14 (0.98, 1.32)	0.06	1.06 (0.86, 1.26)	0.10	1.11 (0.89, 1.41)	−0.03	0.97 (0.81, 1.15)	−0.12	0.89 (0.73, 1.13)
Exhaust temperature of the right engine	0.11	1.12 (0.86, 1.34)	0.06	1.06 (0.87, 1.3)	0.12	1.12 (0.96, 1.31)	−0.01	0.99 (0.80, 1.22)	0.41	1.50 (1.11, 1.74)
Pilot pump pressure	0.23	1.26 (0.76, 1.78)	−0.72	0.49 (0.40, 0.89)	0.17	1.18 (0.8, 1.62)	−0.20	0.81 (0.63, 1.27)	−0.50	0.61 (0.47, 0.98)
Hour meter	0.12	1.13 (0.55, 1.69)								
Alarm			*β*	Odds ratio						
Prelub function status (L)			−0.08	0.92 (0.87, 1.00)						
Prelub function status (R)			−0.03	0.97 (0.94, 1.00)						
Auto lubrication alarm			0.22	1.25 (0.76, 1.79)						
Fast-filling indicator			−0.34	0.71 (0.56, 0.94)						
Emergency engine (motor) stop indicator			−0.35	0.71 (0.63, 0.82)						
Hydraulic oil overheat indicator			0.21	1.23 (0.92, 1.71)						
Pump contamination indicator			0.10	1.11 (0.60, 1.56)						
Pump transmission oil level alarm of left unit			−0.54	0.58 (0.53, 0.71)						
Engine warning indicator (L)			−0.16	0.85 (0.61, 1.12)						
Pump transmission oil level alarm of right unit			−0.49	0.61 (0.57, 0.74)						
Engine warning indicator (R)			−0.28	0.75 (0.47, 1.12)						

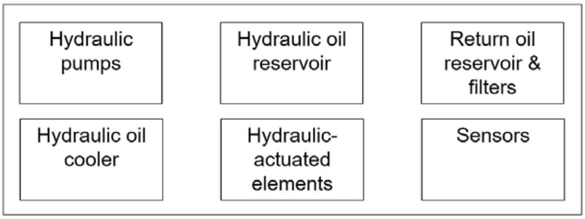

Figure 4. *Functional blocks in the hydraulic system.*

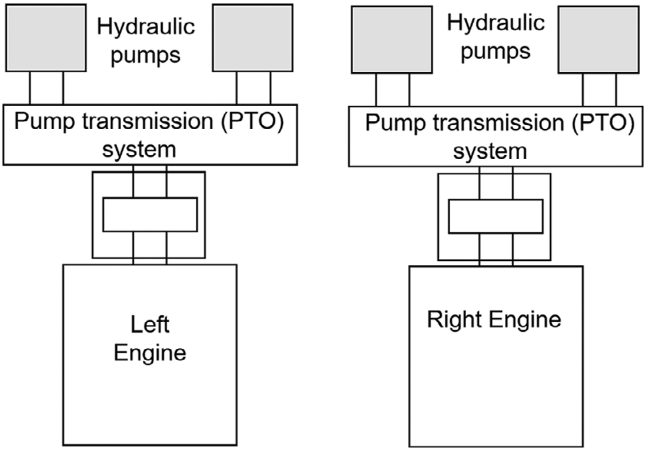
Hydraulic pumps
Pump transmission (PTO) system
Left Engine
Hydraulic pumps
Pump transmission (PTO) system
Right Engine

Figure 5. *Physical layout of the hydraulic pumps, pump transmission system, and engines.*

presence of the other features. We have performed an analysis considering only the hydraulic oil tank temperature, which results in a positive value of the parameter, and then, the higher the hydraulic oil tank temperature, the higher is the chance of failure. This is a common problem when dealing with high-dimensional data, there are many factors working together, and the data represent what happened with the machine, which (strangely) may not be its expected behavior. Moreover, any environmental condition, operator procedure, namely external factors, can lead to a registered data that are not the fact that is happening with the asset. In addition to that, it is also possible that we have confounding variables, which are hard to be analyzed in a high-dimensional dataset. For instance, a potential problem in data analysis is Simpson's paradox (Sprenger and Weinberger, 2021), where we add/remove a feature in a model that can completely change the value of the estimated parameter, changing its interpretation.

- The identification of the engine performance indicators as significant in the probability of hydraulic failure may be due to the physical connections between the engines and the hydraulic pumps. The hydraulic pumps are driven by torque supplied by the engines.
- The model suggests that an increase in pilot pump pressure will reduce the chance of failure. In practice, any issues with the pilot system have the potential to cause issues with the hydraulic system.
- The model shows that both the fast-filling indicator and the emergency engine motor stop indicator when activated are likely to decrease the chance of hydraulic failure. This makes technical sense as the asset is being filled or is stationary at these times.
- While the model does not show the hydraulic oil overheat indicator as being significant, the Odds Ratio lower than 95% confidence limit of 0.92 does not preclude this.
- The pump contamination indicator is triggered by a sensor that is made of two contacts that can be bridged by a solid contaminant in the hydraulic oil. If the contacts are bridged, the contamination

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

indicator will turn on. A common response is to acknowledge the alarm and continue operating to see if the alarm reoccurs. Only if it continues to reoccur, will the operator generate a work order to have the filters checked. This might explain why the Pump Contamination Indicator was not selected by the model.

- The model identifies the oil level alarm in the pump transmission system as reducing the chance of a hydraulic failure, going against the engineers' selection of the alarm as indicating an increase in chance of hydraulic failure. This is difficult to explain from an engineering perspective, as problems with the pump transmission resulting from low oil levels in the gearbox are likely to have knock-on effects on the hydraulic pumps, therefore increasing chance of hydraulic failure.

7. Discussion

This paper presents a data processing method that is applied to streamed sensor values. The output is a wide form data frame containing separate columns for each variable and columns for possible response variables in hydraulic failure events from the fleet management system. This data frame is reduced in size by adapting the OHLC summary method. We have used the OHLC mainly for data visualization. Due to the high-dimensionality of the data, we have considered only the close (C) value in the analysis. However, future work could explore other measures from the OHLC compress data in the modeling. Also presented is a method to transform the data in order to better apply statistical models, accounting for the dependence in the data by applying a lag model. The model's use is demonstrated by performing variable selection using LASSO-LR and bootstrap procedure.

We suggest this OHLC method described can support Original Equipment Manufacturer (OEM) and in-company solutions for health monitoring of HME, when the data are streamed and contain missing data, as so often happens with mobile units. The resulting data frame allows more flexibility in manipulation of individual variables and the capability of setting up data-driven predictive methods from the lag model. This approach also allows visualization of a vast amount of data in a compressed format, allowing for questioning of the sensor data and dependent variables. While this analysis is demonstrated on failures within the hydraulic system, the methods described are applicable to other subsystems on the machine. The processing of data for this analysis provides a universal method for dealing with data presented in this format, and can be applied to other classes of equipment with high volume streamed sensor data.

The results of the model cannot be sensibly compared to other traditional time-series multivariate methods due to the very large number of missing data and very small (31) number of events. We suggest three factors may be at play affecting the agreement (or lack of) for variables of importance identified by the model and the engineers. First, the model is using as a dependent variable the selection of "hydraulic fault" by different operators. There is no quality control over this process, and with a large dataset (45-million rows) and only 31 response variables, one or two incorrect response variable events will significantly affect the model. Second, the hydraulic system is related to other systems on the excavator in both functional and physical ways, and there may be correlations and dependencies that are not obvious yet to engineers who have not traditionally been able to look at data analysis at a system level over time as we are doing here. Finally, we are making the assumption that the sensors are all calibrated and functioning correctly over the 9-month period.

Another feature of the status quo is that although the excavator owner has access to streamed data from the 58 variables and 40 binary indicators, there are many more sensors on the HME with restricted access due to proprietary protocols put in place by the OEMs. In addition to having only a partial view of the entire dataset, we are also not privy to data sampling and other preparation methods such as averaging, nor to the limit settings for the binary variables. By preventing access to all the raw data, OEMs encourage the use of their own commercial software-as-service solutions (Kwon et al., 2016). This is a similar situation to modern cars where the car driver can see a limited number of alarms and values, but the driver needs to go to the OEM repair shop for the mechanic to assess the car's raw data.

Because of our suspicions about the credibility of the dependent variable, we looked into the maintenance work order (MWO) database. All work involving maintainers, at this operation, requires

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

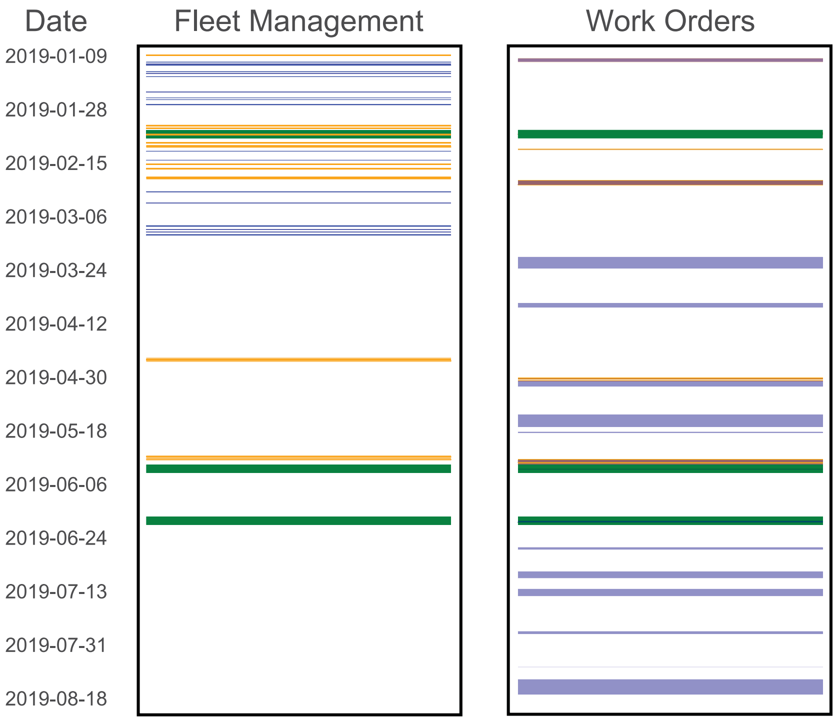

Figure 6. *A comparison over time of hydraulic faults recorded in the fleet management system with corrective maintenance work orders. The green lines indicate dates for which the events match in both systems. The blue lines indicate event dates for the individual systems. The orange lines indicate when an event in the fleet management system had a work order in the next 7 days. The data correspond to hourly open-high-low-close.*

a work order record. Therefore, a hydraulic fault identified by an operator in the fleet management system should trigger the generation of an MWO record. Given that some issues with hydraulic systems are often critical to production, we would expect the work order to be generated within a few hours, and certainly a day, of the fault occurring.

Consider Figure 6; ideally, we expect to see a fault recorded in the Fleet Management System at the same time as a corrective work order in the MWO system. This is clearly only the case in a few events. The data for this plot are very time-consuming to produce requiring manual examination unstructured text containing abbreviations, misspellings, and jargon in hundreds of MWO records. The results though do suggest that we should be cautious of human-generated fault codes on complex machines when it comes to assessing model performance.

8. Final Remarks

The sensor data used in this study are large (45-million rows for one machine) and consist of many variables (98 sensors and indicators), making it difficult to manually monitor the health of mobile machines, especially across an entire fleet, where there can be upward of 50 machines in many operations. Extensive knowledge of each machine is required to select variables to monitor, and it is not feasible to continually monitor every available piece of data. When an entire fleet of mobile machines is considered, this number can be over 200 when excavators, loaders, trucks and graders are included. It is impossible to conceive an efficient monitoring technique that involves manually looking over the data.

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

The analysis outlined in this report is focused on a particular set of streaming data from HME with complementary data from other sources, and shows how specific data processing techniques including OHLC charts and lag models can be used to harness streamed sensor data. The techniques explored are applicable to streaming data on all assets across many industries where problems such as asynchronous recording, missing data, and processing difficulties due to data size exist in time series. The methods outlined enable future work in applying different prediction models to these datasets, such as neural networks or support vector machines. Further experimentation is also suggested to explore the effects of varying the method including the representation of binary variables, varying the window size in the lagged model, using substitute values or otherwise integrating the periods of missing data, and applying models to other subsystems.

Funding Statement. The authors acknowledge funding from the Australian Research Council Centre for Transforming Maintenance through Data Science (Industrial Transformation Research Program Grant No. IC180100030). In addition, Melinda Hodkiewicz acknowledges funding from the BHP Fellowship for Engineering for Remote Operations.

Competing Interests. The authors declare no competing interests exist.

Data Availability Statement. These are confidential data from the company that cannot be shared.

Author Contributions. Conceptualization: all authors; Investigation: all authors; Methodology: all authors; Validation: all authors; Visualization: all authors; Writing—original draft: all authors; Writing—review and editing: all authors.

References

Aleman CS, **Pissinou N**, **Alemany S and Kamhoua GA** (2018) Using candlestick charting and dynamic time warping for data behavior modeling and trend prediction for MWSN in IoT. In *2018 IEEE International Conference on Big Data (Big Data)*. Seattle, WA, USA: IEEE, pp. 2884–2889.

Angelopoulos A, **Michailidis ET**, **Nomikos N**, **Trakadas P**, **Hatziefremidis A**, **Voliotis S and Zahariadis T** (2020) Tackling faults in the industry 4.0 era—A survey of machine-learning solutions and key aspects. *Sensors 20*(1), 109.

Belloumi M (2014) The relationship between trade, FDI and economic growth in Tunisia: An application of the autoregressive distributed lag model. *Economic Systems 38*(2), 269–287.

Bentzen J and Engsted T (2001) A revival of the autoregressive distributed lag model in estimating energy demand relationships. *Energy 26*(1), 45–55.

Correa D, **Polpo A**, **Small M**, **Srikanth S**, **Hollins K and Hodkiewicz M** (2022) Data-driven approach for labelling process plant event data. *International Journal of Prognostics and Health Management 13*(1). https://doi.org/10.36001/ijphm.2022.v13i1.3045.

Dalzochio J, **Kunst R**, **Pignaton E**, **Binotto A**, **Sanyal S**, **Favilla J and Barbosa J** (2020) Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges. *Computers in Industry 123*, 103298.

Everitt B and Hothorn T (2011) The analysis of repeated measures data. In *An Introduction to Applied Multivariate Analysis with R*. Berlin–Heidelberg: Springer, pp. 225–257.

Gregory MM (2006) *Candlestick Charting Explained: Timeless Techniques for Trading Stocks and Futures*. New York: McGraw-Hill.

Hastie T, **Tibshirani R and Wainwright M** (2019) *Statistical Learning with Sparsity: The Lasso and Generalizations*. New York: Chapman and Hall/CRC.

Haugh LD and Box GEP (1977) Identification of dynamic regression (distributed lag) models connecting two time series. *Journal of the American Statistical Association 72*(357), 121–130.

Hegedűs C, **Varga P and Moldován I** (2018) The MANTIS architecture for proactive maintenance. In *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*. Thessaloniki, Greece: IEEE, pp. 719–724.

Kotu V and Deshpande B (2018) *Data Science: Concepts and Practice*. Burlington, MA: Morgan Kaufmann.

Kwon D, **Hodkiewicz MR**, **Fan J**, **Shibutani T and Pecht MG** (2016) IoT-based prognostics and systems health management for industrial applications. *IEEE Access 4*, 3659–3670.

Lee K and Jo G (1999) Expert system for predicting stock market timing using a candlestick chart. *Expert Systems with Applications 16*(4), 357–364.

Loureiro R, **Benmoussa S**, **Touati Y**, **Merzouki R and Bouamama BO** (2014) Integration of fault diagnosis and fault-tolerant control for health monitoring of a class of MIMO intelligent autonomous vehicles. *IEEE Transactions on Vehicular Technology 63*(1), 30–39.

Moura MdC, **Zio E**, **Lins ID and Droguett E** (2011) Failure and reliability prediction by support vector machines regression of time series data. *Reliability Engineering and System Safety 96*(11), 1527–1534.

https://doi.org/10.1017/dce.2022.4 Published online by Cambridge University Press

Phillips J, **Cripps E**, **Lau JW and Hodkiewicz M** (2015) Classifying machinery condition using oil samples and binary logistic regression. *Mechanical Systems and Signal Processing 60*(61), 316–325.

Romeo A, **Joseph G and Elizabeth DT** (2015) A study on the formation of candlestick patterns with reference to Nifty index for the past five years. *International Journal of Management Research and Reviews 5*(2), 67.

Sikorska J, **Hodkiewicz M**, **D'Cruz A**, **Astfalck L and Keating A** (2016) A collaborative data library for testing prognostic models. *Proceedings of the European Conference of the PHM Society 3*(1). https://doi.org/10.36001/phme.2016.v3i1.1579

Sprenger J and Weinberger N (2021). Simpson's paradox. In Zalta EN (ed.), *The Stanford Encyclopedia of Philosophy.* Metaphysics Research Lab, Stanford University.

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

Wickham H and Grolemund G (2017) *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. Newton, MA: O'Reilly Media.

Cite this article: Griffiths T, Corrêa D, Hodkiewicz M and Polpo A (2022). Managing streamed sensor data for mobile equipment prognostics. *Data-Centric Engineering*, 3: e11. doi:10.1017/dce.2022.4