Article:

Yeardley, A.S. orcid.org/0000-0001-7996-0589, Ejeh, J.O. orcid.org/0000-0003-2542-1496, Allen, L. orcid.org/0000-0001-7669-3534 et al. (2 more authors) (2022) Integrating machine learning techniques into optimal maintenance scheduling. Computers & Chemical Engineering, 166. 107958. ISSN: 0098-1354

https://doi.org/10.1016/j.compchemeng.2022.107958

Computers and Chemical Engineering 166 (2022) 107958

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/cace

Integrating machine learning techniques into optimal maintenance scheduling

Aaron S. Yeardley, Jude O. Ejeh, Louis Allen, Solomon F. Brown *, Joan Cordiner

Department of Chemical and Biological Engineering, The University of Sheffield, Sheffield, S10 2TN, United Kingdom

ARTICLE INFO

Keywords:
Predictive maintenance
Optimisation
Maintenance scheduling
Machine learning
Time estimation model

ABSTRACT

Poor maintenance regimes often contribute to unplanned downtimes, quality defects and accidents; thus it is crucial to apply an effective maintenance strategy to achieve efficient and safe processes. Industry 4.0 has brought about a proliferation of digital data and with it new opportunities to advance and improve the way maintenance activities are planned. Here, we propose a novel methodology that utilises machine learning to predict both machine faults and repair time, and uses this data to underpin the scheduling of maintenance activities. This can be used to plan maintenance, and optimise the schedule with a cost objective within the constraints of labour availability and plant layout. When applied to a dataset obtained using a simulated Fischertechnik (FT) model, this methodology reduced the overall plant maintenance costs by decreasing unplanned downtimes and increasing maintenance efficiency. This work provides a promising first step towards improving the way maintenance tasks are approached in Industry 4.0.

1. Introduction

Machine maintenance is paramount to the process industry with regards to both safety and effectiveness. A poor maintenance system has a direct impact on costs, deadlines, quality, and accidents making it catastrophic to an organisation in terms of both operational performance and process safety. Unplanned downtime due to emergency repairs resulting from poor maintenance is estimated to have led to £13.1 billion in discrete manufacturing in 2016 (Thomas and Weiss, 2020). Given the important nature of maintenance, it must be conducted in parallel with a plant's normal operations to avoid compromising the plant's productivity levels (Kobbacy and Murthy, 2008).

Currently, industry is in a process of transformation towards Industry 4.0 where process automation and digitisation are becoming the norm (Gilchrist, 2016). One of the key factors that Industry 4.0 brings with it is the abundance of digital data, which can be used in the control and operation of a plant, improving production efficiency and managing process safety. Technologies using the Internet of Things provide the ability to measure and store large amounts of data from many sensors enriched by the control commands of actuators, transforming manufacturing environments into complex cyber–physical production systems (Gunes et al., 2014). The proliferation of data resulting from wide spread digitisation of manufacturing processes is opening up new opportunities to advance the way maintenance tasks are scheduled. These opportunities promise to improve process safety and reduced maintenance costs.

This manuscript aims to improve the standard smart maintenance scheduling by considering both predictive maintenance and maintenance time estimation modelling. The standard smart maintenance scheduling approach uses predictive maintenance and optimisation to schedule maintenance tasks. However, the authors believe this concept can be improved by additionally considering the time required to complete each maintenance task. This new framework predicts when maintenance is required and how long each task will take before optimising the schedule. Thus, the integration of machine learning techniques creates an optimal maintenance schedule that is robust and reliable.

The most popular method that utilises machine learning and digital data is predictive maintenance (Zonta et al., 2020; Carvalho et al., 2019). Predictive maintenance utilises machine learning on real time sensor data to provide estimations of when maintenance is required on a machine (Yan et al., 2017). The most common predictive maintenance techniques use machine learning classification to predict a fault or failure occurring (Susto et al., 2015). The other techniques use machine learning regression to predict Remaining Useful Life of machines (Van Horenbeek and Pintelon, 2013) and forecast industrial ageing processes (Bogojeski et al., 2021). A systematic literature review by Carvalho et al. (2019) showed that the most common machine learning algorithms used are Random Forest, Neural Networks, Support Vector Machine and k-means clustering. Additionally, they found each

* Corresponding author.
E-mail address: s.f.brown@sheffield.ac.uk (S.F. Brown).

https://doi.org/10.1016/j.compchemeng.2022.107958
Received 5 January 2022; Received in revised form 29 June 2022; Accepted 10 August 2022
Available online 22 August 2022

machine learning method proposed was applied to a specific piece of equipment, for example turbines (Kumar et al., 2018), motors (Dos Santos et al., 2017) and compressors (Prytz et al., 2015). For this reason, it becomes difficult to compare various machine learning algorithms as each study uses vastly different data for validation (Carvalho et al., 2019). Typically, predictive maintenance is employed on single machine systems. This approach lacks applicability to large industrial sites as it fails to consider the causal sequence of fault occurrence in process manufacturing. As such, further research that focuses on the application of predictive maintenance, integrating it within industrial sites to help develop maintenance workflow strategies instead of deriving novel machine learning algorithms is necessary.

Failure to properly consider time to complete maintenance tasks leads to prolonged downtime, increased technician time, and poorly executed jobs causing process incidents (Palmer, 2013). Estimating the time a maintenance task takes is a difficult task in large industrial settings, but it is essential to allow maintenance tasks to be accomplished more efficiently, leading to lower costs (Nyman and Levitt, 2006). Various methods are available to help estimate the time required including time study (Duffuaa and Raouf, 2015), predetermined motion time series (Alkan et al., 2016), or estimations based on past experience. However, these methods often lead to inaccurate errors leading to expensive overtime and rushed fixes.

Machine learning has been proven to be a valuable tool for time estimation models to aid the prediction of product manufacturing times (Liu and Jiang, 2005; Lingitz et al., 2018). Therefore, maintenance time estimation models using machine learning algorithms are possible. To the authors knowledge, Khalid et al. (2020) developed the only maintenance time estimation model. In their work, the historical work orders, functional locations and equipment related variables were used in machine learning algorithms to create better estimations of work hours for preventative maintenance tasks in an Oil Company. The work compared nine machine learning algorithms and found the Random Forest algorithm performed the best, decreasing the mean absolute error from 4.57 h (when using estimation based on experience) to 3.83 h. A decrease of 0.74 h from using a maintenance time estimation model is 16% better than from using estimation. Fully realised, this saving is significant; for example in 2016 census data estimates that the US spent $50 billion on maintenance and repair, with a 16% saving on this representing around $8 billion (Thomas, 2018). While this number is somewhat inflated by the inclusion of building maintenance and other internal expenditures, it does do well to capture the gravity of the potential savings that predictive maintenance could contribute.

The knock-on impact of a failure to consider maintenance time is a poor maintenance schedule. This is to the detriment of profitability (Vassiliadis and Pistikopoulos, 2001). Careful consideration of scheduling is required since performing more preventative maintenance will prevent serious failures but can cause unnecessary downtime and incur high maintenance costs. On the other hand, too little maintenance leads to corrective maintenance where tasks are performed due to failures occurring, thus, leading to process downtime and increased expenses. Traditional scheduling approaches rely on frequent periods of plant shutdown to perform maintenance tasks. Maintenance schedule optimisation is now a popular research topic due to its capabilities in increasing plant profits. It has been conducted for short-term scheduling of a multipurpose plant (Dedopoulos and Shah, 1995), long-term chemical plant turnarounds (Amaran et al., 2015), cleaning schedules in a furnace (Jain and Grossmann, 1998). These maintenance scheduling techniques are optimised to reduce costs focusing on when to schedule periodic maintenance. This, however, is not an optimal approach as it does not consider the relationship between maintenance and machine degradation. Thus, combining maintenance schedule optimisation with predictive maintenance offers opportunities to improve current practices. In literature, adaptive process scheduled have been developed to select acceptable process conditions based on predicted anomalies (Görür et al., 2021). Alternatively, condition-based maintenance scheduling enables dynamic maintenance scheduling based on the estimations from predictive maintenance (Mobley, 2002) to help create optimisation schedules (Jardine et al., 2006). Recently, preventative maintenance and optimisation were combined to create a schedule for a biomass boiler (Macek et al., 2017), a building heating ventilation and air conditioning system (Wu et al., 2021), and an ethylene cracking furnace system (Feng et al., 2021). Research studies have illuminated the combination of predictive maintenance and developed complex mathematical optimisation techniques, yet to the authors knowledge, no study to date has examined the optimisation of condition-based scheduling for a full industrial process.

The work presented within this paper develops a novel maintenance framework and then investigates its efficacy through its application to a full industrial process as a case study. The standard smart maintenance process involves using predictive maintenance with optimisation to build a maintenance schedule. This schedule however cannot be accurately constructed without an appreciation for the time required to conduct maintenance. Using machine learning to accurately forecast this time enables a more accurate schedule. Thus, the novel methodology developed in this work integrates machine learning techniques to optimise the way maintenance is conducted and hence drive savings not only through reduced equipment downtime but also through reduced labour costs.

Here, we present a novel methodology that can analyse the collection of machine sensor data to provide an optimum maintenance schedule through the combination of multiple machine learning techniques. We aim to provide a data-driven approach that automates the learning of models for predictive maintenance and maintenance time estimation. The main objective of this work, therefore, is to build such a workflow that implements machine learning and optimisation in an approach that produces an optimum maintenance schedule. To do this, we first perform an investigation into the classification algorithms readily available for predictive maintenance. Predicting whether a fault has occurred in each machine can be used as the first tool in a promising application to develop robust maintenance scheduling in an industrial plant. We then seek to improve condition-based maintenance schedules by implementing a maintenance time estimation model that offers greater accuracy compared to physical observation. This work will build on the time estimation models previously created by Khalid et al. (2020) by estimating the maintenance time required to fix a predicted fault using live sensor readings as input variables. Finally, the workflow is completed by using the plant layout data, the predicted faults and the estimated maintenance time to optimise a full industrial process.

The remainder of the paper is structured as follows. First, Section 2 presents the novel workflow created. Then a case study is described in Section 3 to which each stage of the workflow is tested on. Finally, Section 4 presents the analysis from each of the investigations and the results produced from the overall workflow.

2. Maintenance policy method

To create a robust maintenance schedule, we propose the workflow shown in Fig. 1 that combines three techniques to enable accurate scheduling of maintenance tasks. The novel maintenance framework builds on standard smart maintenance policies by incorporating maintenance time estimation to ensure the maintenance schedule produced is robust. Here, we outline the methodology focusing on the implementation of the algorithm withing industry to create a maintenance policy.

Fig. 1 shows historical maintenance records and sensor readings require feature engineering and separation to train and validate machine learning models. This training process is necessary to ensure the machine learning models can accurately predict whether a fault has occurred on each machine in the plant and how long the maintenance

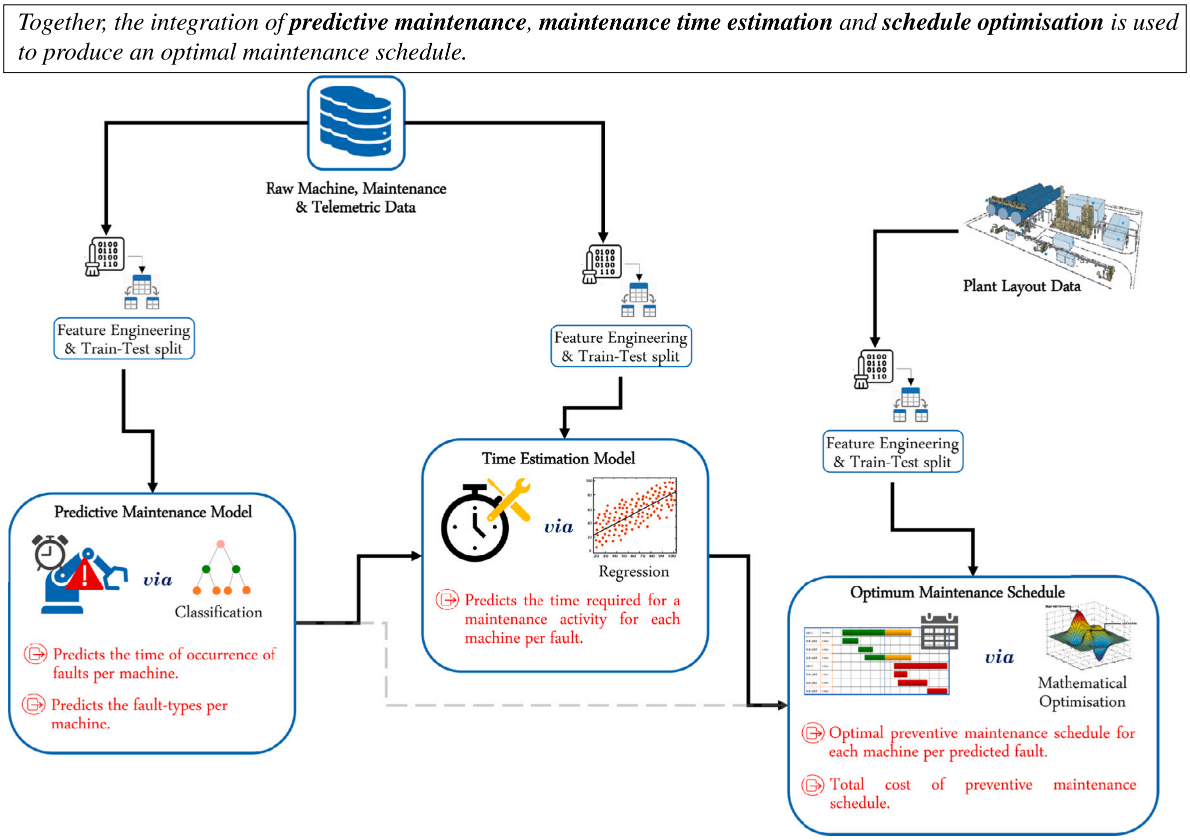

Fig. 1. A flowchart of the ensemble of machine learning techniques used to produce an optimum maintenance schedule.

task will take to complete. Validation is necessary to prevent any occurrence of overfitting of the model (i.e. learning the noise of the data, not the trend). The novel methodology uses raw machine maintenance and telemetric data to make predictions regarding the state of each machine. For this case study, the training and prediction were conducted in an offline mode, owing to the lack of availability of real-time plant data. Once both the classification model and the regression model are trained and validated to ensure they are achieving accurate predictions, the novel maintenance workflow proceeds as follows.

The first element of the workflow uses a classification model for predictive maintenance to classify which machine requires maintenance to fix a fault based on sensor readings. Please note, that these faulty machines have not yet failed but are indicating faults, or signs of failure, which will need intervention before complete machine failure is realised. The output from the predictive maintenance model is a machine number that requires maintenance.

Maintenance time estimation is then performed using a further regression model to accurately predict the maintenance time required to fix a fault using the sensor readings as input variables. The training of this model uses historic sensor readings to learn a blackbox function that maps the sensor readings to previous times required to fix a fault.

The results from both predictive models are passed to the maintenance schedule optimisation. The maintenance schedule optimisation seeks to determine the most cost-effective maintenance plan. Here, the mathematical model is explained in detail for the maintenance schedule optimisation with the objective to minimise the total cost comprising downtime costs, technician costs and the cost of part for replacement under corrective and/or preventative maintenance scenarios.

3. Case study

The following section describes the case study to which the methodology described in the previous section is applied.

3.1. Data

The application of the method outlines in Section 2 requires the availability of historical maintenance and complex sensor reading data related to an industrial plant. However, the availability of real-data from industry is extremely limited due to confidentiality issues. For this reason, data provided by a simulation, developed by Klein and Bergmann (2019), of a cyber–physical production system using a Fischertechnik (FT) factory model was used. The production plant consists of five workstations as described in an ontological knowledge base (Klein et al., 2019).

The large FT plant provides a realistic and challenging case study for detecting faults using a simulation of an industrial production plant. The FT simulation consists of 14 machines. The sensor readings indirectly relate to each machine as some machines had more sensors fitted than others. The cyber–physical system contains 61 different sensors and actuators, measuring the telemetry of every machine to monitor their current condition. The sensor readings measure various telemetrics such as the differential pressure sensor on the pneumatic lift. The simulation generated 28 faults in total by running multiple run-to-failure simulations where the sensor readings and the fault were recorded with time. Altogether, the simulation created 27073 readings for each of 61 sensors. These sensor readings form the inputs for the models described in Section 2.

Besides the sensor readings, machine states are also required for each time so that if a fault has occurred maintenance can be conducted before failure occurs. Primarily, it is important to know at any given point during the simulation if a machine is in fault mode, and if so which machine. To this end, 27073 readings is coupled with an integer identifier for a machine in a fault condition or 0 where no fault was observed.

In addition to this, the actual nature of the fault should be noted and, for the FT plant simulation, 28 unique faults are observed. Therefore, at any given time during the simulation, if the system is at fault the nature of that fault was given its own integer identifier. When the system was not at fault, an additional 'no-fault' class was recorded. This results in 29 unique fault-type classes representing the system fault name. Knowing the nature of the fault is crucial to predicting the required time to perform maintenance.

Predictive maintenance in this case is posed as a classification problem, where at each time the 61 sensor readings are used to predict which of the 14 machines is in a state of fault, if any. The training data set can therefore be represented by Eq. (1) where $\mathcal{X}$ is the training data set consisting of pairwise relationships between inputs, x_t, the sensor readings at a given time period t, and r_t the number of the faulting machine.

$$\mathcal{X} = \{x_t, r_t\}_{t=1}^{N} \tag{1}$$

The classification problem can therefore be posed as Eq. (2), where h is the classification model, C_i is the set of times where no fault has occurred, C_j is the set of times where a fault has occurred somewhere in the system and $\mathcal{N}$ is the machine which has caused the fault state.

$$h(x_t) = \begin{cases} 0 \text{ if } x_t \in C_i,\ i \neq j \\ \mathcal{N} \text{ if } x_t \in C_j \text{ where } \mathcal{N} \in 1 \dots 14 \end{cases} \tag{2}$$

In addition to these datum, the time used to fix each fault would provide the output that could then be used for training and testing. Given that the FT plant is a model simulation, this is not possible. Therefore, we have combined expert knowledge with that from the Ontology (Klein et al., 2019) to generate maintenance task times for each of the 28 faults that occur and the failures that each fault could lead to. Additionally, Gaussian noise with zero mean and unit STD was added on to account for the variance between each maintenance task (for example, fixing a low wear fault may take 53 minutes previously but 62 minutes the next time). The average time for each maintenance task obtained in this manner is shown in Table 2.

The plant layout data (provided by Klein et al. (2019)) and the following assumptions were used to provide a basis for the maintenance scheduling:

- The plant operates at a sold out supply chain. When the plant is shutdown, it is losing profits.
- The plant is operating at a just in time manufacturing rate and overtime is not incurred.
- The plant is sequential so that a shutdown machine shutdowns the entire plant.
- Middle value products are packaged so that every hour of down-time is worth £10,000.
- A maintenance engineer can only work on one machine at a time.
- Multiple engineers are available.
- Each maintenance engineer is highly skilled in all departments and costs include planning time and overheads. This costs the plant £32.53 per hour of maintenance (Glassdoor Inc., 2021).
- Faults can be fixed without replacing parts.
- Failures are fixed by replacing parts that have a cost informed by an expert as shown in Table 2.

The objective of the maintenance scheduling is assumed to minimise cost under the assumptions mentioned. For the data given, we want to determine the maintenance schedule for each fault that occurs on a machine in the plant. This can be obtained using mathematical formulation described in Section 3.4.1.

3.2. Predictive maintenance

Here, we compare five classification techniques; Decision Tree, Random Forest, Neural Network, AdaBoost and Quadratic Discriminant Analysis. The implementation used here was taken from the Python library, Scikit Learn 1.0 (Pedregosa et al., 2011). Each classification technique used gradient descent to minimise the cross-entropy loss function and train the predictive maintenance models. The techniques are applied to the data to provide and insight into the promising tools readily available.

Table 1
A summary of the data used to train and test predictive maintenance techniques.

Machine no.	Train	Test
No Fault	22,763	3,233
1	9	2
2	14	24
3	207	234
4	21	6
5	23	27
6	7	2
7	28	45
8	105	97
9	36	31
10	16	11
11	13	20
12	42	34
13	6	0
14	13	4
Total	**23,303**	**3,770**

To ensure a robust comparison, the full data is split into training and test data based on complete simulations from start to finish so that each individual simulation leading to a fault are either only included in the test or training data set (Klein and Bergmann, 2019). Therefore, a fault that continued for multiple time steps was not found in both training and test data. A summary of the classification data is shown in Table 1, where the clear split between training data and test data can be seen for each of the 15 classes.

3.3. Time estimation model

The second stage of the proposed workflow involves creating a time estimation model. Thus, here the study provides a comparison between Gaussian Processes, Neural Networks, Gradient Boosting Regression, Support Vector Regression and Random Forests to evaluate the most promising regression technique to estimate the required time for the maintenance policy.

Irrespective of the technique being used, the models are trained and tested in the same way. Previously, Section 2 described the novel maintenance policy and the importance of using historical data to train the machine learning models. For the maintenance time estimation model, the objective is to accurately predict the time required for maintenance given the sensor readings and the machine number as input variables. To do this, the machine learning regression models listed above are trained using historical data. The machine learning models optimise their parameters to minimise the errors between the observed maintenance time and that predicted b the models. In a sense, the machine learning model is a blackbox function that is optimised to map the input variables to the output. Here, the models use the simulation data that consists of the 61 sensor readings and the machine number as input variables, and the output is the time required to fix the fault in hours. The distribution of the maintenance time in hours is shown in Fig. 2.

Here, cross-validation was chosen due to only 1,077 data points classed as faults being used from the original FT model simulation. Therefore, this study split the data into 15 data sets so that each complete run to failure simulation is kept together in the same set, creating 14 historical maintenance data sets for training, and one remaining data set as new work orders to be used as validation. This procedure is repeated to ensure all the data points are used for testing.

Table 2
Summary of FT model data.

Fault no.	Machine no.	Machine	Plant area	Fault	Fault fix time (hours)	Failure fix time (hours)	Cost of part (£)
1	1	Conveyor	txt15	Driveshaft Slippage	1	5	400
2	2	Lightbarrier	txt15	Lightbarrier Mode 1	0.5	2	100
3	2	Lightbarrier	txt15	Lightbarrier Mode 2	0.5	2	200
4	2	Lightbarrier	txt15	Lightbarrier Mode 3	0.5	2	300
5	3	M1	txt15	High Wear	1	7	200
6	3	M1	txt15	Low Wear	1	7	100
7	3	M1	txt15	Type 2 Wear	1	7	200
8	4	Pneumatic lift	txt15	Leakage Mode 1	0.5	3	100
9	4	Pneumatic lift	txt15	Leakage Mode 2	0.5	3	300
10	4	Pneumatic lift	txt15	Leakage Mode 3	0.5	3	500
11	5	Conveyor	txt16	Driveshaft Slippage	0.5	5	400
12	5	Conveyor	txt16	Big Gear Tooth Broken	2	12	500
13	5	Conveyor	txt16	Small Gear Tooth Broken	2	12	200
14	6	Switch	txt16	Switch Mode 2	0.5	3	200
15	7	Lightbarrier	txt16	Lightbarrier Mode 1	0.5	2	100
16	8	M3	txt16	High Wear	0.5	7	500
17	8	M3	txt16	Low Wear	0.5	7	200
18	8	M3	txt16	Type 2 Wear	0.5	7	300
19	9	Switch	txt17	Switch Mode 1	0.5	2	100
20	9	Switch	txt17	Switch Mode 2	0.5	2	200
21	10	Pneumatic Lift	txt17	Leakage Mode 1	0.5	3	100
22	11	Transport Workpiece	txt17	Transport Workpiece Missing	0.5	9	300
23	12	Pneumatic Lift	txt18	Leakage Mode 1	0.5	3	100
24	12	Pneumatic Lift	txt18	Leakage Mode 2	0.5	3	300
25	12	Pneumatic Lift	txt18	Leakage Mode 3	0.5	3	200
26	13	Transport Workpiece	txt18	Transport Workpiece Missing	0.5	9	200
27	14	Lightbarrier	txt19	Lightbarrier Mode 1	0.5	2	100
28	14	Lightbarrier	txt19	Lightbarrier Mode 2	0.5	2	200

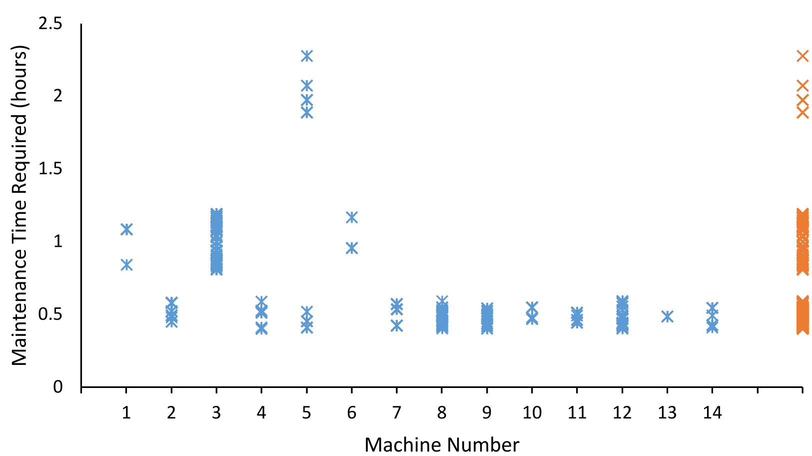

Fig. 2. The maintenance time required to fix a fault for each machine in the FT model plant. At then end of the x-axis, the total spread for all of the machines is shown.

3.4. Maintenance schedule optimisation

The maintenance schedule optimisation utilises the predictions from the previous two methodologies to produce an optimal maintenance schedule. This work focuses on providing a maintenance schedule with the minimum cost for the case study. Therefore, in this section, the mathematical formulation for the maintenance schedule optimisation is described.

3.4.1. Mathematical formulation

The objective of the maintenance schedule optimisation is to minimise the costs to the plant given system constraints resulting from plant procedures, plant layout data, and other operational considerations. Here, we present the mathematical optimisation model with a full nomenclature.

The problem is posed as follows:

Given:

- a set of machines (devices) in a plant;
- a set of possible faults per machine, the (predicted) time of occurrence and whether or not it causes a plant to be shutdown;
- estimated maintenance times required by each fault per machine before and after failure occurs;
- cost of parts and engineering personnel for each fault that occurs within a machine;
- downtime cost of the plant and machine;
- maximum number of available engineers for maintenance activities;

Determine:

- the maintenance schedule for each fault that occurs on a machine within the plant;

So as to:

- minimise the total cost over the time period of consideration. The cost comprises of the plant downtime cost, engineering personnel cost, and the cost of replacing machine parts during maintenance.

In addition to the assumptions previously stated, the following also apply:

- An engineer may only carry out a single maintenance activity for a specific fault at any given time;
- Only three machine states are considered: ‘Running’, ‘Failed’ or ’Under Maintenance’;
- A maintenance activity may start on a machine for a predicted fault before the time of occurrence, further referred to as preventative maintenance (PM);
- Different times are allocated to preventative maintenance activities and maintenance activities carried out after a fault has occurred i.e. a machine already in a failed state requires longer maintenance time (corrective maintenance, CM);

For any given time period, all machines must be in only one state (Eq. (3)), and the total number of machines being simultaneously maintained must not surpass the available number of engineers (Eq. (4)).

$$\sum_{s} S_{ist} = 1 \qquad \forall\, i,t \tag{3}$$

$$\sum_{i} S_{ist} \leq N^{p} \qquad \forall\, s \in \{M\} \tag{4}$$

In order to represent the states of each machine, and their transitions, over the time of consideration, Eqs. (5)–(10) are introduced. The state of machines at the first time period ($t = 0$) are defined using Eqs. (5) and (7), with a machine in a failed state if a fault is predicted to occur, else ‘Running’. The machine remains in a ‘Running’ state at time t except a failure occurs or maintenance activity begins (Eq. (6)).

$$S_{i,'R',0} \geq 1 - F_{i,0} \qquad \forall\, i \tag{5}$$

$$S_{i,'R',t} \geq S_{i,'R',t-1} - S_{i,'M',t} - S_{i,'F',t} \qquad \forall\, i, t > 0 \tag{6}$$

$$S_{i,'F',0} \geq F_{i,0} \qquad \forall\, i \tag{7}$$

A machine may only be in a ‘Failed’ state at any time period if a fault occurs (Eq. (8)), and it may only transition to the state ’Under Maintenance’ (Eq. (9)).

$$S_{i,'F',t} \leq S_{i,'F',t-1} + F_{it} \qquad \forall\, i, t > 0 \tag{8}$$

$$S_{i,'F',t} \geq S_{i,'F',t-1} - S_{i,'M',t} + F_{it} \qquad \forall\, i, t > 0 \tag{9}$$

Finally, the end of maintenance activities on a machine is tracked using the binary variable W^{e}_{it} in Eq. (10).

$$S_{i,'M',t} \geq S_{i,'M',t-1} - W^{e}_{it} \qquad \forall\, i, t > 0 \tag{10}$$

The start and end times of maintenance activities are tracked using the binary variables W^{s}_{it} and W^{e}_{it} respectively, which are evaluated using Eq. (11). Eqs. (12) and (13) ensure these variables may only take a value of 1 when the machine is under maintenance. Maintenance activities may not also start and end at the same time period (Eq. (14)).

$$W^{s}_{it} - W^{e}_{it} \geq S_{i,'M',t} - S_{i,'M',t-1} \qquad \forall\, i,t \tag{11}$$

$$W^{s}_{it} \leq S_{i,'M',t} \qquad \forall\, i,t \tag{12}$$

$$W^{e}_{it} \leq S_{i,'M',t-1} \qquad \forall\, i,t \tag{13}$$

$$W^{s}_{it} + W^{e}_{it} \leq 1 \qquad \forall\, i,t \tag{14}$$

Given that faults on each machines are being predicted, it becomes important to allow for preventative maintenance actions in the schedule, as opposed to the traditional corrective maintenance actions after a fault occurs. This feature is incorporated into the model using Eq. (15).

$$W^{s}_{it} \leq \sum_{t'=0}^{t+\upsilon\rho^{p}} F_{it'} - \sum_{t'=0}^{t-1} W^{s}_{it'}\bigg|_{t>0} \qquad \forall\, i,t \tag{15}$$

where υ denotes the number of time periods before fault occurrence maintenance activities are allowed to start for each fault.

As each machine can have a number of possible faults that can cause failure with different cost implications for maintenance, it is also important to predict each machine’s state per fault that occurs and maintenance activity required. Eq. (16) evaluates the start and end times of a maintenance action (PM or CM) for each fault that can occur on a machine. The binary variable S^{m}_{ift} takes a value of unity if a machine is currently under maintenance at time t for fault f.

$$W^{s'}_{ift} - W^{e'}_{ift} \geq S^{m}_{ift} - S^{m}_{if,t-1} \qquad \forall\, (i,f) \in I^{f}, t \tag{16}$$

$$W^{s}_{it} = \sum_{f:(i,f)\in I^{f}} W^{s'}_{ift} \qquad \forall\, i,t \tag{17}$$

$$W^{e}_{it} = \sum_{f:(i,f)\in I^{f}} W^{e'}_{ift} \qquad \forall\, i,t \tag{18}$$

It is assumed that only one fault is corrected during a maintenance activity, hence the start time of a maintenance activity for a machine (evaluated using W^{s}_{it}) can only be mapped to one fault (Eq. (17)). The same applies to the end times of maintenance activities (Eq. (18)).

The binary variable S^{m}_{ift} may only take a value of unity if a corresponding fault is predicted to occur on a machine (Eq. (19)), and only one fault is corrected (Eq. (20))

$$S^{m}_{ift} \leq \sum_{t'=0}^{t+\upsilon\rho^{p}} F^{f}_{ift'} \cdot S_{i,'M',t} \qquad \forall\, (i,f) \in I^{f}, t \tag{19}$$

$$S_{i,'M',t} = \sum_{f:(i,f)\in I^{f}} S^{m}_{ift} \qquad \forall\, i,t \tag{20}$$

A similar set of constraints to Eqs. (19) and (20) are used to determine when a machine is in a failed state for a specific fault (Eqs. (21)–(22)). In order to accurately calculate the downtime costs, Eqs. (23) and (24) are introduced. Eq. (23) determines if any machine under consideration in the plant is in a failed state for each time period using the binary variable $\overline{S}^{f}_{t}$. In Eq. (24) on the other hand, the binary variable S^{f}_{t} only takes a value of unity when a machine with a fault f which causes plant shutdown is in a failed state.

$$S_{i,'F',t} = \sum_{f:(i,f)\in I^{f}} S^{f}_{ift} \qquad \forall\, i,t \tag{21}$$

$$S^{f}_{ift} \leq \sum_{t'=0}^{t+\upsilon\rho^{p}} F^{f}_{ift'} \cdot S_{i,'F',t} \qquad \forall\, (i,f) \in I^{f}, t \tag{22}$$

$$\overline{S}^{f}_{t} \geq S_{i,'F',t} \qquad \forall\, i,t \tag{23}$$

$$S^{f}_{t} \geq \mu_{if} S^{f}_{ift} \qquad \forall\, (i,f) \in I^{f}, t \tag{24}$$

In order to properly attribute maintenance duration’s for corrective and preventative maintenance actions, Eqs. (25)–(32) are introduced. The time difference between fault occurrence and the start of a maintenance activity is determined using Eq. (25) when corrective ($\kappa^{f}_{if} > 0$) or preventative maintenance occurs ($\kappa^{p}_{if} > 0$) for each fault f on a machine i.

$$\kappa^{p}_{if} - \kappa^{f}_{if} = \sum_{t} t \cdot (F^{f}_{ift} - W^{s'}_{ift}) \qquad \forall\, (i,f) \in I^{f} \tag{25}$$

A binary variable, γ_{if}, which takes a value of 1 when PM actions are performed is then evaluated using Eq. (26), and big ’M’ constraints introduced to ensure only one of κ^{f}_{if} and κ^{p}_{if} take non-zero values for each machine-fault pair.

$$\gamma_{if} \leq \kappa^{p}_{if} \qquad \forall\, (i,f) \in I^{f} \tag{26}$$

$$\kappa^{p}_{if} \leq \hat{M} \cdot \gamma_{if} \qquad \forall\, (i,f) \in I^{f} \tag{27}$$

$$\kappa^{f}_{if} \leq \hat{M} \cdot (1 - \gamma_{if}) \qquad \forall\, (i,f) \in I^{f} \tag{28}$$

The number of contiguous times in which a machine with a particular fault is under maintenance is then enforced using Eq. (29) depending on whether CM (M_{if}^{f}) or PM (M_{if}^{p}) actions are deemed optimal by the model.

$$S_{ift}^{m} = \gamma_{if} \sum_{t} \delta_{if\theta}^{p} W_{if,t-\theta+1}^{s'} + (1-\gamma_{if}) \sum_{t} \delta_{if\theta}^{f} W_{if,t-\theta+1}^{s'} \qquad \forall\,(i,f) \in I^{f}, t \tag{29}$$

where

$$\delta_{ift}^{f} = \begin{cases} 1, & t < M_{if}^{f} \\ 0, & t \geq M_{if}^{f} \end{cases} \qquad \forall\,(i,f) \in I^{f}, t \tag{30}$$

$$\delta_{ift}^{p} = \begin{cases} 1, & t < M_{if}^{p} \\ 0, & t \geq M_{if}^{p} \end{cases} \qquad \forall\,(i,f) \in I^{f}, t \tag{31}$$

Finally, Eq. (32) enforces the maintenance times (corrective or preventative) for each machine-fault pair only if the fault is predicted to occur.

$$\sum_{t} S_{ift}^{m} = F_{ift}^{f}\left(M_{if}^{p}\gamma_{if} + M_{if}^{f}(1-\gamma_{if})\right) \qquad \forall\,(i,f) \in I^{f} \tag{32}$$

The objective function is defined by Eq. (33) and minimises the total cost accrued by the plant over the time of consideration. It comprises the sum of the plant downtime cost, machine downtime cost, personnel and parts cost for every maintenance activity per machine-fault pair.

$$min \sum_{it}\left(\hat{C}^{d} \cdot S_{t}^{f} + \hat{C}^{f} \cdot \overline{S}_{t}^{f} + \sum_{f:(i,f)\in I^{f}} \hat{C}_{if}^{e} \cdot W_{ift}^{s'}(1-\gamma_{if}) + \hat{C}^{p} \cdot S_{i,'M',t}\right) + N^{p} \tag{33}$$

subject to Eqs. (3)–(32). This results in a mixed integer non-linear programming (MINLP) model owing to the bilinear terms in Eq. (29) which can be solved using popular MINLP or mixed integer quadratically constrained programming (MIQCP) solvers.

4. Results

4.1. Predictive maintenance

As described in Section 2, this research analysed five machine learning classification techniques and evaluated each using three popular classification error diagnostics. The standard measures considered are Precision, Recall and the F1 Score, each calculated as weighted averages based on the number of examples per class. An accuracy value was not chosen due to the imbalance between classes as shown in Table 1, where 3,233 test points are in the class “No Fault” out of a total 3,770. Therefore, if the classification algorithm predicted “No Fault” constantly would achieve an accuracy score of 85.76%. Each error metric counts the number of true positive predictions, but the precision score represents this as a ratio to all of the predicted positives, whereas, the recall is the ratio to all actual positives. To combine them both into one metric, the F1 scores weight both recall and precision equally.

Table 3 show the weighted averages obtained for each of the machine learning methods. As can be seen, the best results were obtained by the Quadratic Discriminant Analysis model due to it having the largest value for the precision, the recall and the F1 score. The results for all five machine learning models are satisfactory, but to get the best from the maintenance policy, trust in the predicted faults is of highest priority. Therefore, for the given case study, the Quadratic Discriminant Analysis was the chosen classification model to predict the faults passed on to the time estimation model.

Table 3
Resulting diagnostic values from the predictive maintenance.

	Precision	Recall	F1 score
Decision Tree	0.839	0.878	0.838
Random Forest	0.735	0.858	0.792
Neural Network	0.788	0.861	0.799
AdaBoost	0.736	0.858	0.793
Quadratic Discriminant Analysis	0.880	0.883	0.877

4.2. Time estimation model

Once the predictive maintenance model has been trained, the next stage would be to ensure the time scheduled for maintenance is accurately fed into the optimisation model. Hence, validation of a time estimation model is of critical importance.

Once again, the case study was used to test five machine learning algorithms for regression, these being; Gaussian Processes, Neural Network, Gradient Boosting, Support Vector Regression, and Random Forest. Using these algorithms provides a variety of approaches that are widely available for use (Pedregosa et al., 2011; Milton and Brown, 2019) and so increases the chances of finding the optimum model.

As previously stated in Section 3.3, the machine learning models are tested using 15-fold cross-validation, ensuring every fault available as data were used for testing. Once again, three popular error diagnostics were chosen to evaluate each regression technique. For regression, these diagnostics were the coefficient of determination (R^2), the standardised root mean squared error (RMSE (-)), and the root mean squared error of time (RMSE (hours)).

Table 4 presents the values calculated for the validation of each method. Clearly, all five methods have an average R^2, above 50% but below 70%, providing an indication of the amount of variation in the observed maintenance time ascribable to the estimated maintenance time. The two RMSE diagnostics reveal the distance each prediction is away from the true value on average using Eq. (34).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(x_i - y_i)^2}{N}} \tag{34}$$

However, the RMSE (-) compares standardised values calculated using the mean and standard deviation of the data. Whereas, the RMSE (hours) puts the error in the predictions into a more visual context for time estimation using units of hours. The average actual work is 0.696 h, meaning an average offset of less than 0.35 h gives a 50% time overlap to consider when setting work tasks. In comparison, previous work from Khalid et al. (2020) has shown using traditional methods the offset is 87%, but the research used machine learning algorithms to reduce this to 73%. Therefore, although predictions are initially seen to be satisfactory, the results from this time estimation investigation show the techniques applied provide an improved accuracy in predicting the maintenance time.

Importantly, Table 4 shows the Gaussian Process has the best performance as it has the largest R^2 value and lowest values for error. As such, the Gaussian Process is chosen to be used in this case study for a time-estimation model. Fig. 3 clearly shows the performance of the Gaussian Process by plotting residuals of the predicted maintenance hours vs the actual maintenance hours. It can be clearly seen that the Gaussian Process performs exceptionally well, often predicting values close to the observed time. However, a significant anomaly did occur when the GP predicted a maintenance time of just 1 hour when the true maintenance time was actually 2.2 h. This anomaly highlights the importance of batch learning and cross-validation before implementing an online policy.

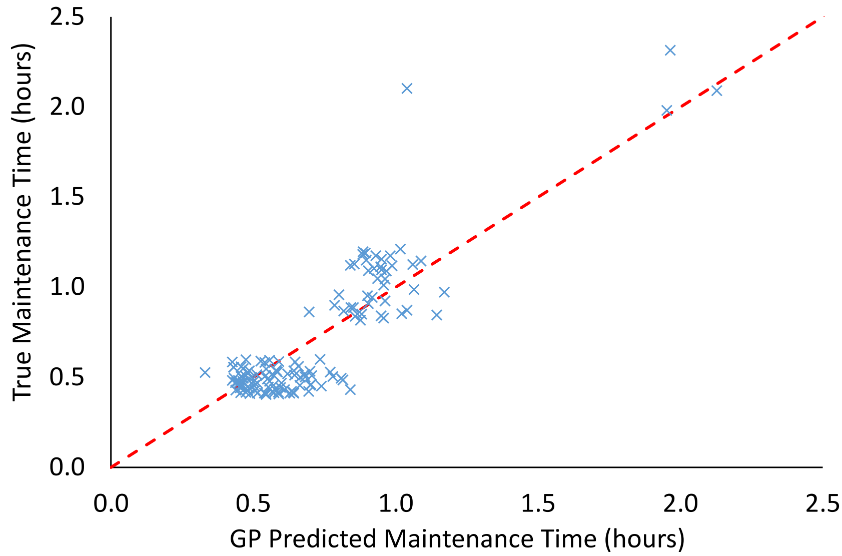

Fig. 3. The residuals predicted by the Gaussian Process regression model.

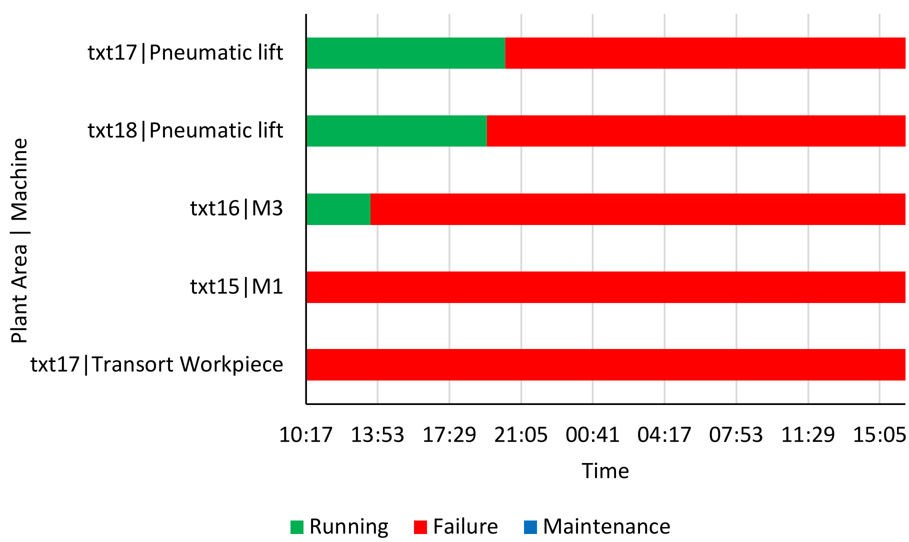

Fig. 4. Case 1 input data.

Table 4
Resulting diagnostic values from the time prediction.

	R squared	Standardised RMSE	RMSE (hours)
Gaussian Process	66.3%	0.600	0.202
Neural Network	57.9%	0.670	0.226
Gradient Boosting	56.9%	0.681	0.230
Support Vector Regression	51.8%	0.723	0.244
Random Forest	51.7%	0.720	0.243

4.3. Maintenance schedule optimisation

Following the preventative maintenance and time estimation model implementation, we use these models' outputs to obtain a cost optimal maintenance schedule for the case study presented in Section 3.1.

The data implemented was split into three cases based on calendar dates of fault occurrence. Figs. 4–6 shows the Gantt chart produced for each case. Each case corresponded to a day's worth of data showing the machines and their relative time periods of fault occurrence with each time unit corresponding to a five minute period. The maintenance scheduling model was solved using Gurobi 9.0.3 on 2 threads of an Intel i7-3615QM processor with 16 GB RAM.

In order to demonstrate the benefits of our proposed maintenance policy two strategies are considered:

- Condition-based scheduling — using predictive maintenance model outputs and performs maintenance activities as soon as a fault occurs based on available resources;
- Our proposed workflow which extends the condition-based scheduling allowing for preventative maintenance actions. Hence, maintenance actions can be performed on machines before faults occur, which lead to failures in a preventative manner. This allows for a more flexible approach to maintenance and allows for downtime cost savings and a reduced maintenance time.

The Gantt chart in Fig. 7 shows the results for Case 1 for the condition-based scheduling strategy. Using the preventative maintenance model predictions, and performing maintenance only at the point of, or after, a fault occurs, a total cost of £8,196 was obtained requiring three engineers during the period of consideration. A reduced cost of £7,322 however, is obtained using our proposed strategy with the same number of personnel (Fig. 8). This is as preventative maintenance actions for faults detected take shorter times translating to reduced

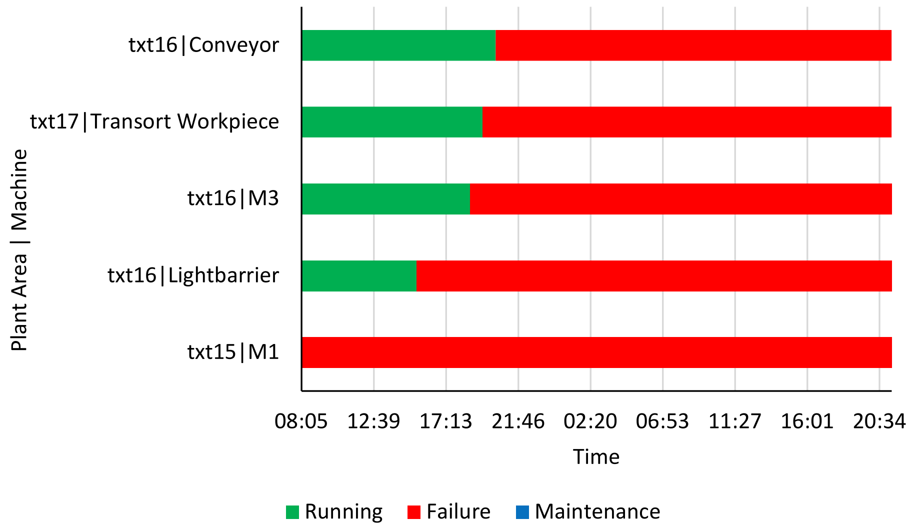

Fig. 5. Case 2 input data.

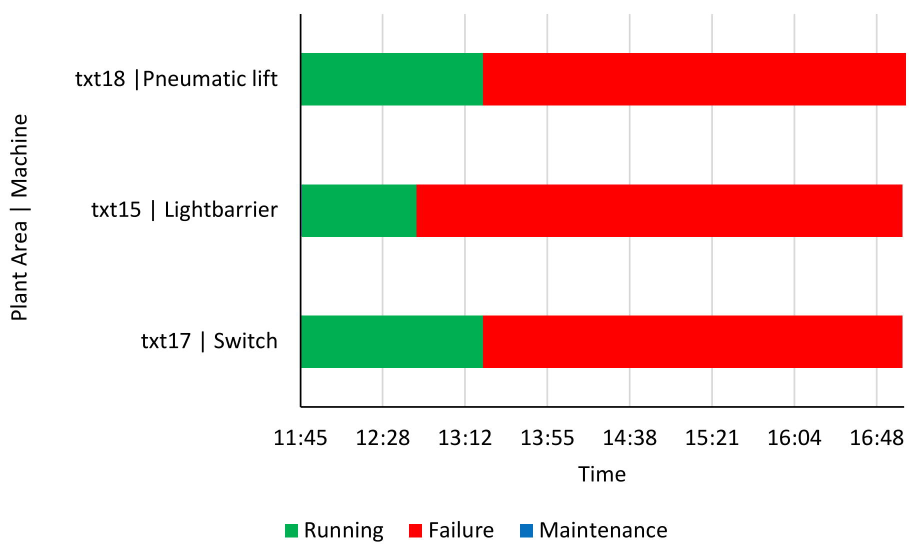

Fig. 6. Case 3 input data.

man-power costs, as well as a cost saving on parts replacement, overall reducing costs by 10.7%.

Similar sets of results are observed for Case 2 for the condition-based (Fig. 9) and preventative maintenance scheduling (Fig. 10) strategies. A cost reduction of up to 44% was obtained, which can be attributed not only to the difference in maintenance times for preventative and corrective maintenance actions, but also to the reduced number of engineering personnel required for the time period in question. As in Case 1, but also more obvious in the current case, our proposed maintenance strategy also leads to a shorter completion time despite the reduced number of personnel. This allows for additional production hours for the plant which can lead to higher revenues and/or flexibility of plant operation.

In Case 3 with a smaller number of machines subject to predicted faults, Fig. 11 shows the cost optimal maintenance schedule when failures are corrected at the point of or after occurrence. A 25% reduction in cost is also observed with a reduced number of personnel when our preventative maintenance strategy is adopted instead (Fig. 12). In each of these three cases, it becomes evident that obtaining optimal schedules for preventative maintenance tasks (in comparison to condition-based maintenance) leads not only to a general reduction in costs via reduced total maintenance times, but also to possible reduction in the number of personnel required.

5. Conclusion

A poor maintenance system can be catastrophic to an industrial plant's performance and safety. In this paper, we propose a novel methodology to be used in industry to create an optimum maintenance schedule. The methodology utilises the abundance of data made available due to Industry 4.0 by combining various machine learning methods to create an optimum maintenance schedule.

The proposed framework consists of three stages, predictive maintenance, maintenance time estimation and optimisation. In this research, the main objective of this work is to build such a methodology and investigate each stage as a proof of concept. Due to the lack of availability of real-data, we applied the proof of concept investigation to a Fischertechnik (FT) simulation model (Klein and Bergmann, 2019). Thus, each stage of the algorithm was investigated through batch training, whereas, in a real application, we would recommend this batch training approach before implementing an online version to ensure validation of the best machine learning model is chosen to fit the plant's data.

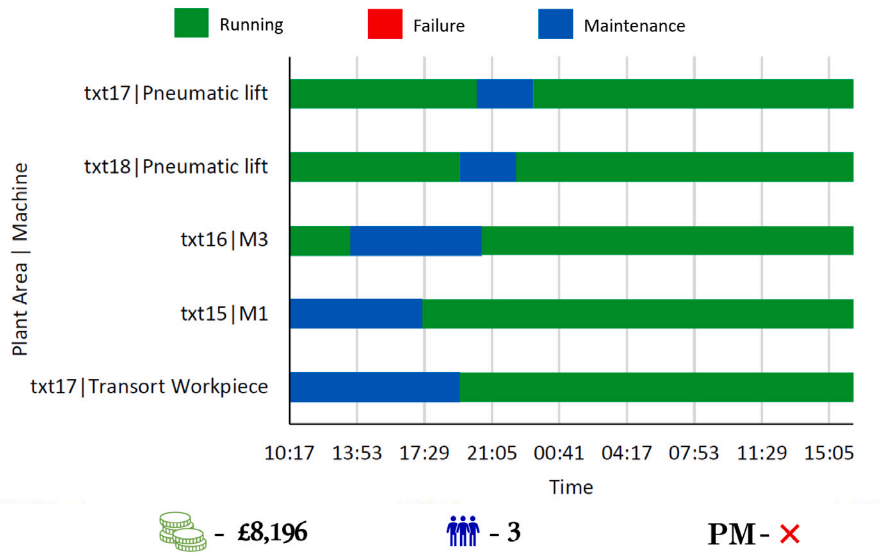

Fig. 7. Case 1 results — No preventative maintenance.

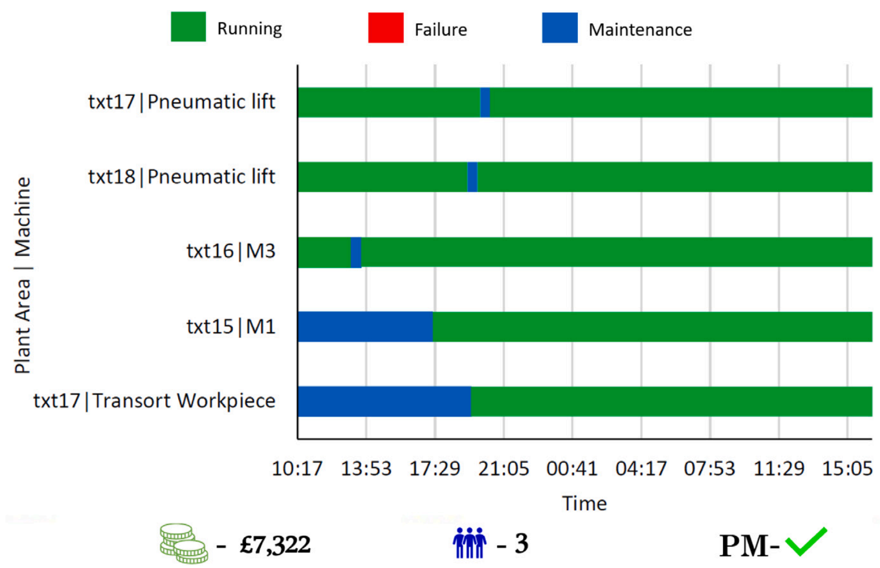

Fig. 8. Case 1 results — With preventative maintenance.

The algorithm describing the maintenance policy was robustly tested by investigating the three steps of the framework. First, the popular method of predictive maintenance was analysed by comparing five readily available machine learning methods to ensure faults occurring in the FT plant can be identified. In this work, the demonstration of methodology, training and validation of machine learning methods, is vital to understanding how the maintenance policy begins its successful implementation. The predictive maintenance results showed the Quadratic Discriminant Analysis model to be superior of the five methods as it has the largest value for the precision, the recall and the F1 score. Therefore, the Quadratic Discriminant Analysis was chosen to identify faulty machines and share predictions to get the best from the maintenance policy.

The second stage of the algorithm requires accurate predictions to ensure the final proposed schedule can be followed without delays. Here, we addressed a gap in literature by investigation maintenance time estimation models. Once again, the method was demonstrated using five regression techniques that use historical data from the FT plant to map the sensor readings to the time it takes to fix a fault before failure occurs. Results found the Gaussian Process (Yeardley et al., 2020, 2021) to be the best performing machine learning method, often predicting values close to the observed time.

At the final stage, the FT model data was split into three cases based on calendar dates of fault occurrence. To demonstrate the benefits of the proposed maintenance policy, the maintenance schedules of the three cases were optimised using condition-based scheduling and our proposed strategy. The results of the optimisation provided high quality maintenance schedules and evidence that preventative maintenance obtains schedules with a lower cost, personnel requirement and/or overall maintenance times when compared to condition-based maintenance.

The workflow presented in this work could readily be applied to reduce maintenance costs; however, first, to assess its efficacy at a wider scale it would be beneficial to apply this data to other case studies based on real industrial plant data. Further, the workflow could be implemented in an online environment that allows data to flow automatically to update maintenance schedules.

Nomenclature

The abbreviations and symbols used are defined as follows:

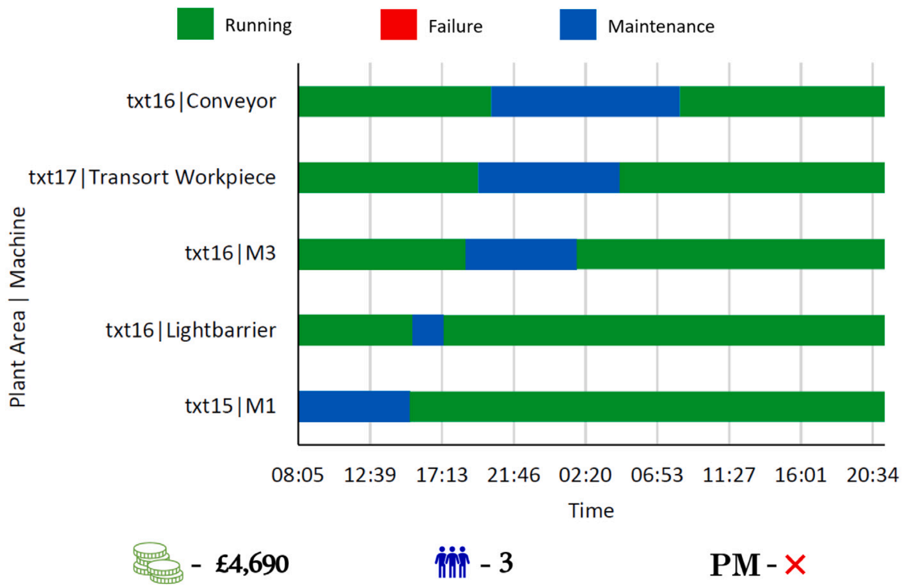

Fig. 9. Case 2 results — No preventative maintenance.

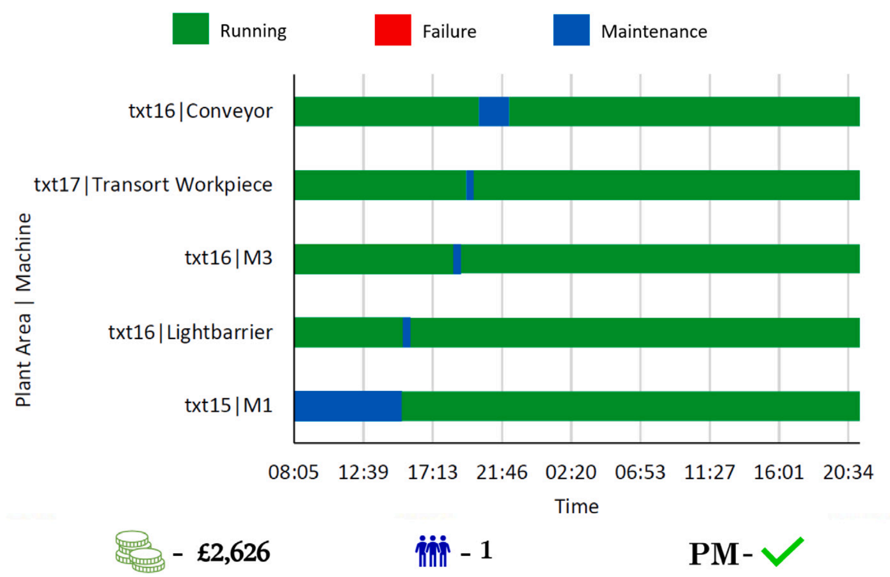

Fig. 10. Case 2 results — With preventative maintenance.

Abbreviations/Acronyms	
CM	Corrective Maintenance
PM	Preventative Maintenance
MINLP	Mixed Integer Non-linear Programming
Indices	
f	fault description
i	devices/machines
s	device/machine state
t	time/period
Set	
I	devices/machines
I^f	ordered pairs of device and possible faults
S	set of possible device/machine states
T	set of time/periods

Parameters	
μ_{if}	0,1 parameter denoting if fault f on device i causes the plant to shutdown
ρ^p	0,1 parameter denoting if preventative maintenance is allowed
υ	number of times period before fault occurrence maintenance is allowed to start
$\hat{C}^p$	personnel/engineer cost per hour
$\hat{C}^e_{if}$	cost of device parts for replacement for fault f in device i
$\hat{C}^d$	plant downtime cost per hour
$\hat{C}^f$	device downtime cost per hour
F_{it}	0,1 values denoting time device i fails

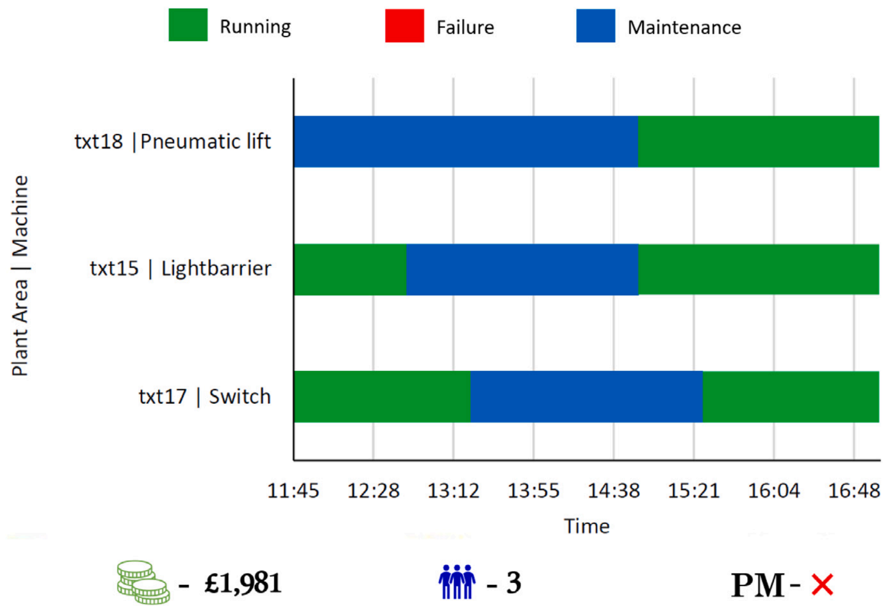

Fig. 11. Case 3 results — No preventative maintenance.

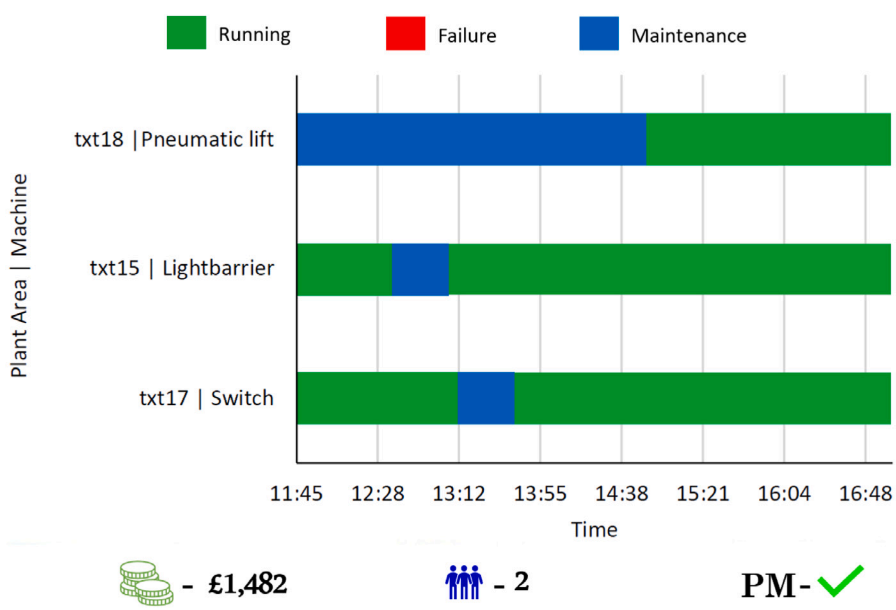

Fig. 12. Case 3 results — With preventative maintenance.

F^f_{ift}	0,1 values denoting time fault f occurs on device i
$\hat{M}$	a 'big' number
M_i	time taken for maintenance on item i
M^f_{if}	time taken for maintenance on item i for fault f after failure occurs
M^p_{if}	time taken for preventative maintenance on item i for fault f on fault detection

Binary variables

γ_{if}	1 if preventative maintenance occurs on device i for fault f; 0 otherwise
S_{ist}	1 if device i is in state s at time t; 0 otherwise
S^f_t	1 if any device is in a failed state at time t and causes the plant to shutdown; 0 otherwise
$\overline{S}^f_t$	1 if any device is in a failed state at time t; 0 otherwise
S^f_{ift}	1 if device i is currently in a failed state for fault f at time t; 0 otherwise
S^m_{ift}	1 if device i is currently under maintenance for fault f at time t; 0 otherwise
W^e_{it}	1 if maintenance is completed on device i at time t; 0 otherwise
$W^{e'}_{ift}$	1 if maintenance is completed on device i for fault f at time t; 0 otherwise
W^s_{it}	1 if maintenance starts on device i at time t; 0 otherwise
$W^{s'}_{ift}$	1 if maintenance starts on device i for fault f at time t; 0 otherwise

Integer variables

N^p	number of technicians available for maintenance

Continuous variables

κ_{if}^{p} difference in time periods between the time of fault occurrence and the start of maintenance for preventative maintenance actions

κ_{if}^{f} difference in time periods between the time of fault occurrence and the start of maintenance for corrective maintenance actions

CRediT authorship contribution statement

Aaron S. Yeardley: Conceptualization, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing – original draft, Project administration. **Jude O. Ejeh:** Conceptualization, Methodology, Software, Data curation, Writing – review & editing. **Louis Allen:** Writing – review & editing, Resources. **Solomon F. Brown:** Conceptualization, Writing – review & editing, Supervision, Project administration. **Joan Cordiner:** Conceptualization, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alkan, B., Vera, D., Ahmad, M., Ahmad, B., Harrison, R., 2016. A model for complexity assessment in manual assembly operations through predetermined motion time systems. Procedia CIRP 44 (Vm), 429–434. http://dx.doi.org/10.1016/j.procir.2016.02.111.

Amaran, S., Sahinidis, N., Sharda, B., Morrison, M., Bury, S., Miller, S., Wassick, J., 2015. Long-term turnaround planning for integrated chemical sites. Comput. Chem. Eng. 72, 145–158. http://dx.doi.org/10.1016/j.compchemeng.2014.08.003.

Bogojeski, M., Sauer, S., Horn, F., Müller, K.R., 2021. Forecasting industrial aging processes with machine learning methods. Comput. Chem. Eng. 144, 107123. http://dx.doi.org/10.1016/j.compchemeng.2020.107123, arXiv:2002.01768.

Carvalho, T.P., Soares, F.A., Vita, R., Francisco, R.d.P., Basto, J.P., Alcalá, S.G., 2019. A systematic literature review of machine learning methods applied to predictive maintenance. Comput. Ind. Eng. 137 (April), 106024. http://dx.doi.org/10.1016/j.cie.2019.106024.

Dedopoulos, I., Shah, N., 1995. Optimal short-term scheduling of maintenance and production for multipurpose plants. Ind. Eng. Chem. Res. 34, 192–201. http://dx.doi.org/10.1021/ie00040a019.

Dos Santos, T., Ferreira, F., Pires, J., Damasio, C., 2017. Stator winding short-circuit fault diagnosis in induction motors using random forest. In: 2017 IEEE International Electric Machines and Drives Conference. IEMDC 2017, pp. 1–8. http://dx.doi.org/10.1109/IEMDC.2017.8002350.

Duffuaa, S., Raouf, A., 2015. Planning and Control of Maintenance Systems: Modelling and Analysis. Springer International Publishing, pp. 1–348. http://dx.doi.org/10.1007/978-3-319-19803-3.

Feng, W., Feng, Y., Zhang, Q., 2021. Multistage distributionally robust optimization for integrated production and maintenance scheduling. AIChE J. 67 (9), 1–19. http://dx.doi.org/10.1002/aic.17329.

Gilchrist, A., 2016. Industry 4.0. the Industrial Internet of Things. A Press, p. 250. http://dx.doi.org/10.1007/978-1-4842-2047-4, arXiv:arXiv:1011.1669v3.

Glassdoor Inc., 2021. Chemical plant operator salaries. https://www.glassdoor.co.uk/Salaries/chemical-plant-operator-salary-SRCH_KO0,23.htm.

Görür, O.C., Yu, X., Sivrikaya, F., 2021. Integrating predictive maintenance in adaptive process scheduling for a safe and efficient industrial process. Appl. Sci. (Switzerland) 11 (11), http://dx.doi.org/10.3390/app11115042.

Gunes, V., Peter, S., Givargis, T., Vahid, F., 2014. A survey on concepts, applications, and challenges in cyber-physical systems. KSII Trans. Internet Inf. Syst. 8 (12), 4242–4268. http://dx.doi.org/10.3837/tiis.2014.12.001.

Jain, V., Grossmann, I., 1998. Cyclic scheduling of continuous parallel-process units with decaying performance. AIChE J. 44, 1623–1636. http://dx.doi.org/10.1002/aic.690440714.

Jardine, A.K., Lin, D., Banjevic, D., 2006. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mech. Syst. Signal Process. 20 (7), 1483–1510. http://dx.doi.org/10.1016/j.ymssp.2005.09.012.

Khalid, W., Albrechtsen, S.H., Sigsgaard, K.V., Mortensen, N.H., Hansen, K.B., Soleymani, I., 2020. Predicting maintenance work hours in maintenance planning. J. Qual. Maint. Eng. 27 (2), 366–384. http://dx.doi.org/10.1108/JQME-06-2019-0058.

Klein, P., Bergmann, R., 2019. Generation of complex data for AI-based predictive maintenance research with a physical factory model. In: ICINCO 2019 - Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics, Vol. 1, no. September. pp. 40–50. http://dx.doi.org/10.5220/0007830700400050.

Klein, P., Malburg, L., Bergmann, R., 2019. Ftonto: A domain ontology for a fischertechnik simulation production factory by reusing existing ontologies. In: CEUR Workshop Proceedings, Vol. 2454, no. October.

Kobbacy, K.A.H., Murthy, D.N.P., 2008. Complex System Maintenance Handbook. Springer Science and Business Media, Berlin Heidelberg, Germany.

Kumar, A., Shankar, R., Thakur, L., 2018. A big data driven sustainable manufacturing framework for condition-based maintenance prediction. J. Comput. Sci. 27, 428–439. http://dx.doi.org/10.1016/j.jocs.2017.06.006.

Lingitz, L., Gallina, V., Ansari, F., Gyulai, D., Pfeiffer, A., Sihn, W., 2018. Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer. Procedia CIRP 72, 1051–1056. http://dx.doi.org/10.1016/j.procir.2018.03.148.

Liu, B., Jiang, Z.-h., 2005. The man-hour estimation models and its comparison of interim products assembly for shipbuilding. Int. J. Oper. Res. 2 (1), 14–19.

Macek, K., Endel, P., Cauchi, N., Abate, A., 2017. Long-term predictive maintenance: A study of optimal cleaning of biomass boilers. Energy Build. 150, 111–117. http://dx.doi.org/10.1016/j.enbuild.2017.05.055.

Milton, R.A., Brown, S.F., 2019. ROMCOMMA. URL https://github.com/C-O-M-M-A/rom-comma.

Mobley, R.K., 2002. 18 - World-class maintenance. In: Mobley, R.K. (Ed.), An Introduction to Predictive Maintenance, (Second Edition), second ed. In: Plant Engineering, Butterworth-Heinemann, Burlington, pp. 394–433. http://dx.doi.org/10.1016/B978-075067531-4/50018-X, URL https://www.sciencedirect.com/science/article/pii/B978075067531450018X.

Nyman, D., Levitt, J., 2006. Maintenance Planning, Scheduling and Coordination, second ed. Industrial Press Inc, pp. 1–84.

Palmer, R.D., 2013. Maintenance Planning and Scheduling Handbook. McGraw-Hill Education.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Prytz, R., Nowaczyk, S., Rögnvaldsson, T., Byttner, S., 2015. Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. Eng. Appl. Artif. Intell. 41, 139–150. http://dx.doi.org/10.1016/j.engappai.2015.02.009.

Susto, G.A., Schirru, A., Pampuri, S., McLoone, S., Beghi, A., 2015. Machine learning for predictive maintenance: A multiple classifier approach. IEEE Trans. Ind. Inf. 11 (3), 812–820. http://dx.doi.org/10.1109/TII.2014.2349359.

Thomas, D.S., 2018. The costs and benefits of advanced maintenance in manufacturing. http://dx.doi.org/10.6028/NIST.AMS.100-18, URL http://nvlpubs.nist.gov/nistpubs/ams/NIST.AMS.100-18.pdf.

Thomas, D.S., Weiss, B.A., 2020. Economics of Manufacturing Machinery Maintenance. Technical Report, National Institute of Standards and Technology, Gaithersburg, MD, http://dx.doi.org/10.6028/NIST.AMS.100-34, URL https://nvlpubs.nist.gov/nistpubs/ams/NIST.AMS.100-34.pdf.

Van Horenbeek, A., Pintelon, L., 2013. A dynamic predictive maintenance policy for complex multi-component systems. Reliab. Eng. Syst. Saf. 120, 39–50. http://dx.doi.org/10.1016/j.ress.2013.02.029.

Vassiliadis, C.G., Pistikopoulos, E.N., 2001. Maintenance scheduling and process optimization under uncertainty. Comput. Chem. Eng. 25 (2–3), 217–236. http://dx.doi.org/10.1016/S0098-1354(00)00647-5.

Wu, Y., Maravelias, C.T., Wenzel, M.J., ElBsat, M.N., Turney, R.T., 2021. Predictive maintenance scheduling optimization of building heating, ventilation, and air conditioning systems. Energy Build. 231, 110487. http://dx.doi.org/10.1016/j.enbuild.2020.110487.

Yan, J., Meng, Y., Lu, L., Li, L., 2017. Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance. IEEE Access 5, 23484–23491. http://dx.doi.org/10.1109/ACCESS.2017.2765544.

Yeardley, A.S., Bellinghausen, S., Milton, R.A., Litster, J.D., Brown, S.F., 2021. Efficient global sensitivity-based model calibration of a high-shear wet granulation process. Chem. Eng. Sci. 238, 116569. http://dx.doi.org/10.1016/j.ces.2021.116569.

Yeardley, A.S., Bugryniec, P.J., Milton, R.A., Brown, S.F., 2020. A study of the thermal runaway of lithium-ion batteries: A Gaussian process based global sensitivity analysis. J. Power Sources 456 (February), 228001. http://dx.doi.org/10.1016/j.jpowsour.2020.228001.

Zonta, T., da Costa, C.A., da Rosa Righi, R., de Lima, M.J., da Trindade, E.S., Li, G.P., 2020. Predictive maintenance in the industry 4.0: A systematic literature review. Comput. Ind. Eng. 150 (August), http://dx.doi.org/10.1016/j.cie.2020.106889.