Adoption of machine learning technology for failure prediction in industrial maintenance: A systematic review

Joerg Leukel[a*], Julian González[a], and Martin Riekert[a]

[a] *Faculty of Business, Economics & Social Sciences*
University of Hohenheim, Schwerzstr. 35, 70599 Stuttgart, Germany
**corresponding author: joerg.leukel@uni-hohenheim.de*

Abstract. Failure prediction is the task of forecasting whether a material system of interest will fail at a specific point of time in the future. This task attains significance for strategies of industrial maintenance, such as predictive maintenance. For solving the prediction task, machine learning (ML) technology is increasingly being used, and the literature provides evidence for the effectiveness of ML-based prediction models. However, the state of recent research and the lessons learned are not well documented. Therefore, the objective of this review is to assess the adoption of ML technology for failure prediction in industrial maintenance and synthesize the reported results. We conducted a systematic search for experimental studies in peer-reviewed outlets published from 2012 to 2020. We screened a total of 1,024 articles, of which 34 met the inclusion criteria. We focused on understanding the datasets analyzed, the preprocessing to generate features, and the training and evaluation of prediction models. The results reveal (1) a broad range of systems and domains addressed, (2) the adoption of up-to-date approaches to preprocessing and training, (3) some lack of performance evaluation mitigating the overfitting problem, and (4) considerable heterogeneity in the reporting of experimental designs and results. We identify opportunities for future research and suggest ways to facilitate the comparison and integration of evidence obtained from single studies.

Keywords: Failure prediction; Fault prediction; Machine learning; Predictive maintenance; Systematic review.

This is an Accepted Manuscript of the following article:

Leukel, J., González, J., & Riekert, J. (2021). Adoption of machine learning technology for failure prediction in industrial maintenance: A systematic review. *Journal of Manufacturing Systems*, 61, 87-96. https://doi.org/10.1016/j.jmsy.2021.08.012

1. Introduction

Failure prediction is an essential component of industrial maintenance strategies aimed at preventing the occurrence of system failures and minimizing unplanned downtimes of equipment, machines, and processes. Predictive maintenance relies upon accurate predictions of future failures to devise the timely scheduling of maintenance activities [1,2]. Approaches to failure prediction analyze current and past data representing system states, events, and operations. An increasingly used technology for failure prediction is machine learning (ML), which enables the training of a prediction model from time-series data, evaluation of the model's performance, and deployment in a productive environment [3]. The increased adoption of ML technology for failure prediction has been facilitated by improvements of ML algorithms, implementation of these algorithms in freely available software packages, and enhanced provision and richness of industrial data in the context of big data analytics and stream processing platforms [4,5].

There is evidence indicating that ML-based failure prediction models are effective for a variety of systems, including agricultural machines [6], wind turbines [7], aircraft components [8,9], ICT devices [9], and even production plants [10]. Moreover, similar evidence exists both for predictions very near to the time of failure, e.g., a few minutes [11], and very far from the time of failure, e.g., several weeks [12]. This variety in the prediction task coincides with a wide array of ML techniques from which researchers and practitioners can choose when developing a specific prediction model. The techniques are concerned with the underlying ML algorithms, the transformation of operational data into features, the training of prediction models from historic data, and the assessment of how well the models perform.

To inform the development of effective prediction models, insights into the ML techniques and their effects on prediction performance are necessary. However, the accumulating evidence from previous research is not well documented. Although the adoption of ML technology for predictive maintenance has been assessed several times, there has not been a review that focused on failure prediction. One group of reviews examined approaches for many different tasks related to industrial maintenance [13–18]. The tasks included failure detection (did a failure occur?), failure diagnosis (why did the failure occur?), condition monitoring (what is the current condition?), failure prediction, and prediction of other variables, such as remaining useful life and degradation. For instance, a review by Zhang et al. [15] included thirty-three studies of which only three examined failure prediction A review by Stetco et al. [19] considered failure detection and failure prediction. Yet, the evidence from studies for one task cannot be compared and integrated with evidence for a different task. For that reason, another group of reviews focused on the task of failure diagnosis but excluded the failure prediction task [20,21].

Collectively, the insights gained from extant reviews insufficiently inform us regarding the development of ML-based failure prediction models and their performance evaluation. Our research addresses this important gap in the literature by focusing on the failure prediction task and conducting a task-specific systematic review. Thus, the objectives of our research are to (1) assess

the adoption of ML technology in previous research examining failure prediction in industrial maintenance and (2) synthesize the reported results to suggest avenues for future research.

2. Method

We conducted the systematic review in accordance with the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines [22].

2.1. Information sources and search strategy

The literature search covered the years 2012 through 2020 and was carried out using the electronic database Scopus and previous reviews. We chose Scopus because it has greater coverage of peer-reviewed literature than the Web of Science [23]. The search in Scopus was performed on February 12, 2021. In addition, we considered articles from three recent systematic reviews on applications of ML for predictive maintenance [14,15,24].

We performed the bibliographic search on the article's title, abstract, and keywords using search terms for three concepts: (1) ML technology was represented as ("machine learning" OR "deep learning" OR classification OR "support vector machine" OR "random forest" OR regression OR "neural network*), (2) failure prediction was coded as ("predictive maintenance" OR "machine failure" OR "machine prognostics" OR "machine health" OR "condition based maintenance" OR "machine degradation"), and (3) performance evaluation was defined as (experiment* OR metric OR evaluat* OR performance OR accuracy OR precision OR recall OR auc).

2.2. Eligibility criteria and study selection

We selected articles that reported about the adoption of ML technology for predicting failures of a material system using real-world data. We excluded literature reviews, conceptual research (e.g., [25,26]), case studies if they lacked performance evaluation [27,28], simulation studies using artificial data [29,30], and approaches for failure detection [31], failure diagnosis [32], and condition monitoring [33]. We also excluded approaches for predicting other variables than failures, such as remaining useful life [34,35], degradation [36,37], and system performance [38,39], which are metric variables. Further inclusion criteria were defined as follows: article published in a journal or conference proceedings, written in English, original contribution, and full-text available.

The screening of articles was independently carried out by three reviewers (JL, JG and MR), who used a codebook describing the eligibility and exclusion criteria. Conflicting codes were resolved by discussing the title, abstract, and keywords of each article in detail. For the articles that went through the screening, the full-texts were downloaded and then independently assessed by the same reviewers. This step was followed by a discussion of all codes to resolve any inconsistency.

2.3. Data collection process

For the articles that met all eligibility criteria, two reviewers (JG and MR) independently extracted data using a codebook for the data items defined in Section 2.4. Data points were recorded in a spreadsheet format and the results were discussed with the main investigator (JL) to agree upon the final data points.

2.4. Data items

Fig. 1 presents the conceptual model of our review by structuring the process for ML-based failure prediction and indicating the relevant data items. We derived the process from fundamental steps concerning the application of ML algorithms [40].

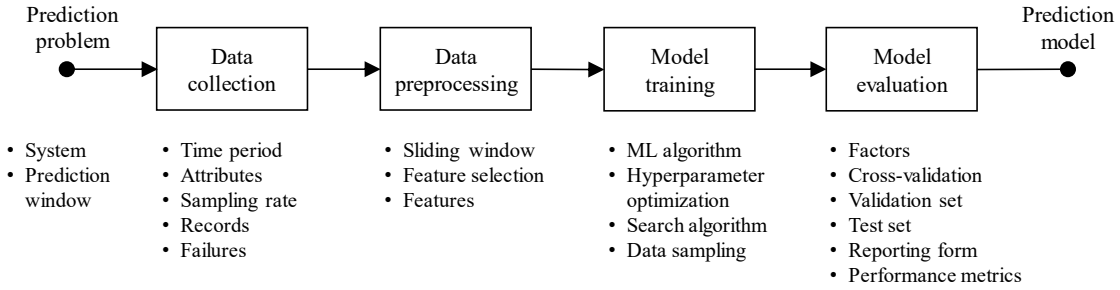

Fig. 1. ML process and data items.

The prediction problem is forecasting whether a system will fail at a specific point of time in the future. *System* can be any material artifact, such as machine or component, being operated by an organization to fulfill some meaningful purpose (e.g., manufacturing goods, providing energy, and offering transportation). *Prediction window* represents the time in advance the prediction will be made. The process for ML-based failure prediction is organized into subprocesses for data collection, data preprocessing, model training, and model evaluation, which we discuss in the following paragraphs.

Data collection is concerned with the acquisition of operational data and the creation of a dataset. *Time period* describes the duration for which historic data was collected. *Attributes* define the number of variables that were recorded for each time step. *Sampling rate* represents the interval between each time step, e.g., ten minutes. *Records* is the total number of entries in the dataset, which often corresponds to a flattened table-based data structure. *Failures* indicate the absolute and relative frequencies of failure states.

Data preprocessing deals with how the dataset is transformed into a representation from which a prediction model can be learned. Specifically, preprocessing works on the dataset's attributes to produce so called features. *Sliding window* defines the time span how long historic data will be analyzed for the prediction (e.g., one week). *Feature selection* is concerned with selecting a subset of relevant attributes and can adopt different statistical techniques, such as correlation and

principal component analysis (PCA) [41]. *Features* are the total number of features, which will be forwarded to the next step.

Model training is the task of learning a function that maps a set of input variables onto output variables based on example pairs of input-output. The input variables are all the features describing the system and the output is the failure state. *ML algorithm* denotes the supervised learning algorithm for determining the mapping function, such as support vector machines (SVM) [42], artificial neural networks (ANN) [43], and random forests (RF) [44]. To control the learning process, *hyperparameter optimization* can be adopted. This optimization can be performed using different *search algorithms*, such as grid search and random search [45]. *Data sampling* addresses the difficulties resulting from imbalanced data [46]. A key characteristic of operational data is that the number of non-failures is several orders of magnitude greater than the number of failures. The data imbalance makes the prediction of failures much more difficult than the prediction of non-failures. Whereas undersampling reduces the number of non-failures in the training dataset, oversampling adds examples derived from the failures in the dataset.

Model evaluation assesses the performance of the prediction model. It is usually handled by exploring how varying one or more *factors*, such as ML algorithm, features, and prediction window, can enhance performance. *Cross-validation* is a procedure that partitions the dataset into complementary subsets (denoted as k), conducts the training on k-1 subsets, and uses the remaining subset (*validation set*) to validate the prediction model on a small set of unknown data [47]. *Test set* indicates whether the prediction model was also evaluated using a separate dataset of unknown data. *Reporting form* indicates the means for the presentation of results (e.g., figures, tables). *Performance metrics* represent the adoption of standard metrics, such as precision and recall [48].

3. Results

3.1. Study selection

Implementing the search strategy described above, we retrieved a total of 1,024 articles. Of these, 147 were selected for the full-text assessment, and 34 met the criteria for inclusion. The initial agreement between reviewers was 84.5% during the screening phase and 75.5% for the full-text assessment (all conflicts were resolved through discussions between the three reviewers involved in each stage). Fig. 2 summarizes the selection process and states the reasons for the exclusion of articles.

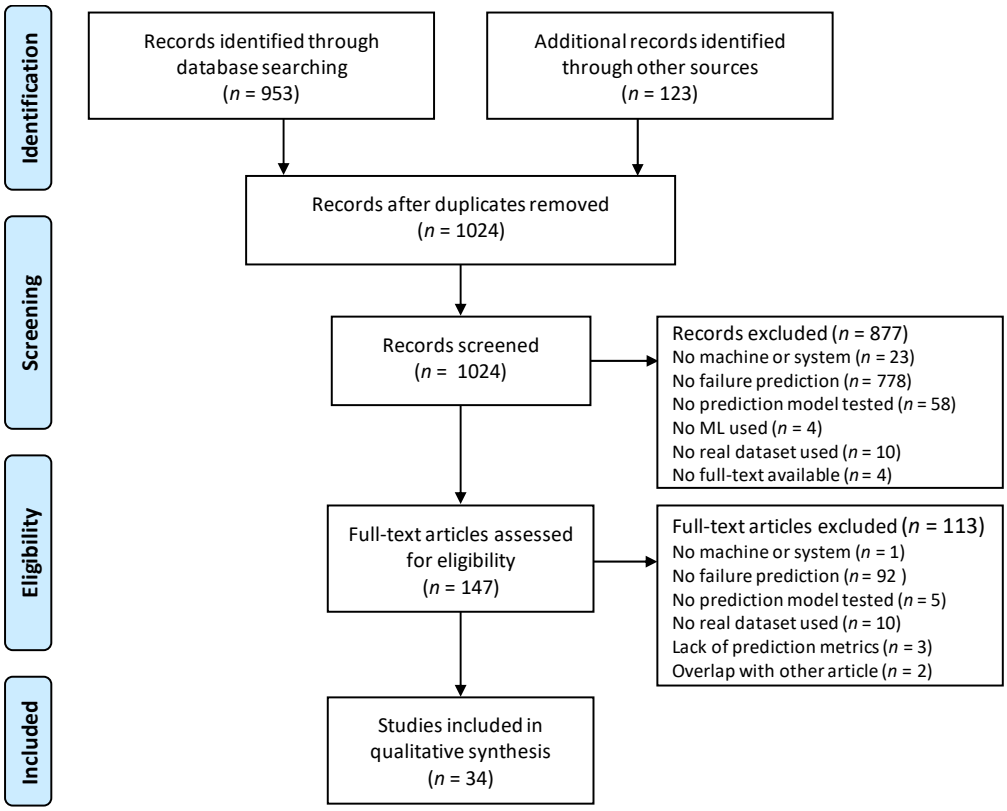

Fig. 2. PRISMA flow diagram.

Table 1 provides an overview over the selected studies. The majority of studies have been published in the last three years (2020: 12; 2019: 6; 2018: 6). The range of systems spanned from specific components, such as fans, pumps, and compressors, to manufacturing plants [49]. Wind turbines, compressors of vehicles, automatic teller machines, and hard disk drives were the only systems examined in more than one study.

Table 1. List of studies included in qualitative synthesis ($N = 34$).

Study	System	Prediction window
Abu-Samah et al. [50]	Thermal treatment equipment (semiconductor)	NR
Alves et al. [11]	Metal stamping machine	5 min
Bonnevay et al. [51]	Electrical device	up to 15 d
Canizo et al. [52]	Wind turbine	1 h
Chen et al. [12]	Air compressor of truck	up to 90 d
Colone et al. [53]	Wind turbine drive-train	1 and 4 h
Dangut et al. [54]	Aircraft component	NR
Figueroa Barraza et al. [55]	a) Offshore natural gas treatment plant b) Sea water injection centrifugal pump	a) 20 to 480 min b) 48 h
Hamaide and Glineur [56]	Rotating condensor inside a synchrocyclotron	5 d
Jansen et al. [57]	Metal machining	1 row
Kaparthi and Bumblauskas [9]	Hard drive disk	NR
Korhoseed and Beyca [58]	Sewerage treatment plant	1 to 9 h
Kolokas et al. [10]	a) Anode production plant b) Injection molding machine	a) 15, 20, 30, 45 min b) 45, 450 min
Kulkarni et al. [59]	Refrigeration and cold storage system	7 d
Kusiak and Verma [7]	Wind turbine	10 to 300 s
Leahy et al. [60]	Induction generator of wind turbine	0.5, 1, 2, 6, 12, 24 h
Lee et al. [61]	Data center	NR
Li et al. [62]	Railway network	a) 7, 14 d, b) 3 m
Lüttenberg et al. [63]	Agricultural machine	NR
Mishra and Manjhi [64]	Component of automatic teller machine	1 m
Nowaczyk et al. [65]	Air compressor of truck	up to 50 w
Orrù et al. [66]	Centrifugal pump	1 w
Pertselakis et al. [67]	Carbon steel cylinder (white goods)	1 d
Proto et al. [49]	Manufacturing plant (white goods)	NR
Prytz et al. [68]	Vehicle compressor	3 to 50 w
Renga et al. [69]	Medium-voltage distribution network	1, 7, 30 d
Rombach and Keuper [70]	Hard drive disk	7 d
Savitha et al. [8]	Aircraft component	NR
Silva and Capretz [71]	Supply fan of building	NR
Susto et al. [72]	Ion implantation tool (semiconductor)	1 to 85 (no unit)
Wang et al. [73]	Automatic teller machine	<1 d
Wijs et al. [74]	Subsurface asset (construction)	NR
Xiang et al. [75]	Component of vending machine	10 d
Yu et al. [76]	Turbine syngas compressor	3, 6, 7 d

Note. d = days. h = hours. m = months. min = minutes. s = seconds. w = weeks. NR = not reported.

Prediction window also exhibited high variability, ranging from a few seconds [7] to several days and even up to 50 weeks [65]. Yet, only a few studies give a justification for the length of the prediction window. For instance, Prytz et al. [68] set the prediction window to enable warnings prior to the next planned inspection. Li et al. [62] chose the length according to the time necessary for inspection by the operator.

3.2. Data collection

Table 2 shows key characteristics of the datasets used. On average, data was recorded over a period of 23.5 months, ranging between 29 days of data center traces for 12,500 machines [61] and eight years of aircraft flight operations [54].

Table 2. Characteristics of the datasets used in studies ($N = 34$).

Study	Time period	Attributes	Sampling rate	Records	Failures
Abu-Samah et al. [50]	10 m	23	Various (m to h)	6,300	82
Alves et al. [11]	NR	NR	0.003 s to 5 min	NR	NR
Bonnevay et al. [51]	1.5 y	4	Event-based	1,250,000	NR
Canizo et al. [52]	2 y	104	10 min	1,787,040	NR
Chen et al. [12]	2 y	NR	Event-based	160,000	200
Colone et al. [53]	5 y	48	10 min	NR	2042
Dangut et al. [54]	8 y	NR	Event-based	NR	NR
Figueroa Barraza et al. [55]	a) NR b) 3 y	a) 10 b) 16	a) 20 min b) NR	a) 4,885 b) 90,965	a) 1,332 b) 26,074
Hamaide and Glineur [56]	1 y	8	NR	30,000,000	8
Jansen et al. [57]	NR	21	NR	260,000	900
Kaparthi and Bumblauskas [9]	1 y	>100	1 d	232,662	1,381
Korhoseed and Beyca [58]	3 m	6	1 min	130,956	NR
Kolokas et al. [10]	a) 2.5 y b) 13 m	a) 15 b) 42	a) 2 to 5 s b) 6 to 10 s	NR	a) 604 b) NR
Kulkarni et al. [59]	2 m	NR	5 min	NR	150
Kusiak and Verma [7]	4 m	>100	10 s	NR	17,609
Leahy et al. [60]	4.5 y	90	10 min	>300,000	NR
Lee et al. [61]	29 d	6	Event-based	10,400,0000	8,957
Li et al. [62]	a) NR b) 25 m	a) 55 b) NR	NR	NR	NR
Lüttenberg et al. [63]	NR	NR	Event-based	3,407	86
Mishra and Manjhi [64]	NR	380	Event-based	NR	NR
Nowaczyk et al. [65]	NR	NR	Event-based	50 to 1,500	180
Orrù et al. [66]	3.5 y	NR	1 h	NR	4
Pertselakis et al. [67]	NR	NR	1 d	NR	NR
Proto et al. [49]	183 d	NR	NR	NR	NR
Prytz et al. [68]	3 y	1,250	Event-based	NR	NR
Renga et al. [69]	6 y	NR	Event-based	153,094	3,901
Rombach and Keuper [70]	1 y	129	1 d	NR	1,155
Savitha et al. [8]	2 m	NR	NR	17,104	5
Silva and Capretz [71]	1 y	9	NR	NR	NR
Susto et al. [72]	NR	31	NR	NR	33
Wang et al. [73]	7 m	NR	Event-based	24,579	3,785
Wijs et al. [74]	1 y	27	NR	107,500	181
Xiang et al. [75]	NR	NR	Event-based	NR	NR
Yu et al. [76]	9 m	>100	1 s	NR	2
Count	26	21	26	17	21

Note. d = days. m = months. min = minutes. s = seconds. y = years. NR = not reported.

All of the studies analyzed numerical data, and categorical data was present in two-thirds of the datasets ($n = 23$). The number of attributes ranged from 4 [51] to 1,250 [68] in two-thirds of the studies reporting that information. The sampling rate varied between fractions of a second [11] and one day ($n = 3$). Eleven studies examined event-based data instead of data recorded with a fixed sampling rate. Because of the broad range of time periods and sampling rates, the total number of records per dataset varied greatly. The smallest dataset included 1,500 records [65], whereas four datasets had more than one million records [51,52,56,61]. One half of the studies give no information about the number of records. Considering that system failures are rare events, it is not surprising that the number of failures was small in one-third of the studies reporting that characteristic ($n = 21$), with seven datasets including fewer than a hundred failures. On the other hand, eight studies examined datasets including thousands of failures. Four studies indicated the

data imbalance by reporting the relative frequency of failures [11,55,63,66].

3.3. Data preprocessing

Table 3 presents an overview of how the datasets were transformed into features. Most studies stated that a sliding window was defined ($n = 20$). The specific approach to feature selection was available from 17 studies, including PCA ($n = 4$), correlation analysis ($n = 4$), variance analysis ($n = 3$), and wrapper-based approaches ($n = 3$). The final number of features ranged between 3 and 144 (for the 20 studies reporting that information).

Table 3. Data preprocessing in studies ($N = 34$).

Study	Sliding window	Feature selection	Features
Abu-Samah et al. [50]	Yes	NR	NR
Alves et al. [11]	NR	NR	NR
Bonnevay et al. [51]	NR	NR	39
Canizo et al. [52]	NR	PCA, correlation	14
Chen et al. [12]	NR	NR	NR
Colone et al. [53]	NR	None (explicitly stated)	NR
Dangut et al. [54]	Yes	NR	39
Figueroa Barraza et al. [55]	Yes	Statistical analysis (no details reported)	a) 4, b) 15
Hamaide and Glineur [56]	Yes	Wrapper-based, backward selection	80, 144
Jansen et al. [57]	Yes	NR	3
Kaparthi and Bumblauskas [9]	Yes	NR	NR
Korhoseed and Beyca [58]	Yes	PCA	7
Kolokas et al. [10]	Yes	a) Hidden Markov model, forward selection b) Forward selection	NR
Kulkarni et al. [59]	Yes	NR	40
Kusiak and Verma [7]	NR	Wrapper genetic & best fit search, boosting tree	11
Leahy et al. [60]	Yes	Variance, missing values, PCA	NR
Lee et al. [61]	Yes	NR	72
Li et al. [62]	Yes	a) PCA, b) Variance	a) 12, b) 19
Lüttenberg et al. [63]	NR	NR	NR
Mishra and Manjhi [64]	Yes	NR	NR
Nowaczyk et al. [65]	NR	NR	NR
Orrù et al. [66]	Yes	NR	8
Pertselakis et al. [67]	NR	NR	NR
Proto et al. [49]	Yes	Correlation, multicollinearity test	NR
Prytz et al. [68]	NR	Wrapper-based, Kolmogorov-Smirnov test	4 to 20
Renga et al. [69]	NR	RF feature selection	13, 17
Rombach and Keuper [70]	Yes	Correlation	6
Savitha et al. [8]	NR	NR	NR
Silva and Capretz [71]	Yes	NR	50
Susto et al. [72]	Yes	Discard constant variables	125
Wang et al. [73]	Yes	Feature ranking, feature evaluation	NR
Wijs et al. [74]	NR	Backward selection	10
Xiang et al. [75]	Yes	Variance, correlation, RF feature selection	100
Yu et al. [76]	NR	Statistical analysis (no details reported)	17
Count	20	17	20

Note. NR = not reported. PCA = principal component analysis. RF = random forests.

3.4. Model training

Table 4 shows that the most frequently adopted algorithms were RF (n = 18), SVM (14), and ANN (12). Fewer studies adopted decision tree (8), gradient boosting (6), k-nearest neighbor (3), and Naïve Bayes (3). A total of 15 different algorithms were tested, but six algorithms were examined in only one study each.

Table 4. Machine learning algorithms in studies (N = 34).

Algorithm	Number of studies	Studies
Random forests	18	[7,9,12,49,51,52,58–61,63–65,67,68,71,73,75]
Support vector machines	14	[7,8,56,58,60–62,66,67,70–73,75]
Artificial neural network	12	[7,8,11,12,53–55,57,61,66,67,71]
Decision tree	8	[7,9,58,60–62,65,67]
Gradient boosting	6	[7,49,58,63,73,75]
Logistic regression	4	[9,60,61,74]
k-nearest neighbor	3	[65,67,72]
Naïve Bayes	3	[50,53,67]
Isolation forest	2	[10,70]
AdaBoost	1	[73]
Associative classification	1	[69]
Elliptic envelope	1	[10]
Local outliner factor	1	[70]
MapReduce-based DPCA approach	1	[76]
T-Squared	1	[76]

Table 5 gives the results for model selection. With respect to hyperparameter optimization, half of the studies followed that approach to control the learning process, of which seven reported about the specific search algorithm used. Three studies stated that they did not adopt hyperparameter optimization; no information was available from the remaining studies (n = 14). To mitigate the problems arising from imbalanced data, four studies adopted oversampling, three studies used undersampling, and two further studies applied under- and oversampling in combination. Two studies indicated the specific sampling ratio (1:4); hence, the failure and non-failure classes accounted for 20% and 80% of the training set, respectively [53,57]

Table 5. Model selection in studies (N = 34).

Study	Hyperparameter optimization	Search algorithm	Data sampling
Abu-Samah et al. [50]	NR	NR	NR
Alves et al. [11]	Yes	NR	NR
Bonnevay et al. [51]	No	No	NR
Canizo et al. [52]	Yes	Grid search	NR
Chen et al. [12]	No	NR	Undersampling, Oversampling
Colone et al. [53]	NR	NR	Undersampling (1:4)
Dangut et al. [54]	NR	NR	Oversampling
Figueroa Barraza et al. [55]	Yes	Grid search	NR
Hamaide and Glineur [56]	Yes	Grid search	NR
Jansen et al. [57]	Yes	Grid search	Undersampling (1:4)
Kaparthi and Bumblauskas [9]	NR	NR	NR
Korhoseed and Beyca [58]	Yes	NR	NR
Kolokas et al. [10]	Yes	NR	NR
Kulkarni et al. [59]	No	NR	NR
Kusiak and Verma [7]	Yes	NR	NR
Leahy et al. [60]	Yes	NR	Undersampling
Lee et al. [61]	Yes	NR	NR
Li et al. [62]	NR	NR	NR
Lüttenberg et al. [63]	NR	NR	Oversampling
Mishra and Manjhi [64]	NR	NR	NR
Nowaczyk et al. [65]	NR	NR	NR
Orrù et al. [66]	Yes	Grid search	Oversampling
Pertselakis et al. [67]	NR	NR	NR
Proto et al. [49]	Yes	NR	NR
Prytz et al. [68]	NR	NR	Oversampling
Renga et al. [69]	Yes	NR	NR
Rombach and Keuper [70]	Yes	Grid search	NR
Savitha et al. [8]	NR	NR	NR
Silva and Capretz [71]	Yes	Random search	NR
Susto et al. [72]	Yes	NR	NR
Wang et al. [73]	NR	NR	NR
Wijs et al. [74]	Yes	NR	Undersampling, Oversampling
Xiang et al. [75]	NR	NR	NR
Yu et al. [76]	NR	NR	NR
Count	17	7	9

Note. NR = not reported.

3.5. Model evaluation

The evaluation phase usually includes (1) the manipulation of factors, (2) assessment of performance using operational data, and (3) reporting of results using standard metrics.

3.5.1. Factors

Table 6 shows that the majority of studies manipulated the ML algorithm used (n = 21), one-third examined different lengths of the prediction window (n = 11), and five studies each varied the number of features and the size of the sliding window, respectively. Although performance depends on many factors throughout the ML process, each additional factor enhances the complexity of the analysis. Thus, it was not surprising that the sample predominantly included single-factor (n = 15)

and two-factor studies (n = 11) but only five three-factor studies; three studies did not manipulate any factor.

Table 6. Factors in studies (N = 34).

Factor	Number of studies	Studies
ML algorithm	21	[7–9,12,49,53–55,57,58,61,65–67,70–76]
Prediction window	11	[7,10,51,53,55,58,60,62,65,68,72]
Number of features	5	[49,56,68,71,73]
Sliding window	5	[9,49,60,62,73]
Oversampling	4	[54,63,68,74]
Hyperparameter	3	[11,54,57]
Dataset size	1	[65]
Rules	1	[50]
Training set	1	[52]

With respect to different ML algorithms, RF and Gradient Boosting were found to be superior in five and three studies, respectively. However, most studies found no evidence for differences in performance between algorithms. Again, it should be noted that the studies examined overlapping but different sets of algorithms.

The findings for different prediction windows were largely consistent, with nine of eleven studies showing that smaller windows enhanced performance. Regarding the number of features, two studies reported positive effects [56,68], two studies found negative effects [49,73], and one study reported a negative effect for RF (but not for SVM and ANN) [71]. Of the five studies that examined different sliding windows, only the study by Proto et al. [49] found a positive effect. Oversampling of the failure class in the training set enhanced performance in three of four studies [63,68,74]. Different oversampling ratios were tested in two studies, of which one study showed a positive effect when oversampling attenuated the data imbalance [68]. Results for different hyperparameters of neural networks were provided in three studies, but no considerable effects were observed [11,54,57]. The dataset size was manipulated in the study by Nowaczyk et al. [65] (positive effect for a greater number of records). Moreover, no effects were found for using specific prediction rules [50] and learning from a larger training set [52], respectively.

3.5.2. Model assessment

Perusal of Table 7 shows that 18 studies stated the specific k-fold cross-validation method used, which partitioned the dataset into three (n = 2), four (n = 3), five (n = 9), nine (n = 1), and ten (n = 3) complementary subsets, respectively.

Table 7. Model assessment in studies ($N = 34$).

Study	Cross-validation	Validation set	Test set	Reporting form	
				Chart	Table
Abu-Samah et al. [50]	4-fold	Yes	Yes	Yes	Yes
Alves et al. [11]	No	Yes	No	Yes	No
Bonnevay et al. [51]	No	NR	No	Yes	No
Canizo et al. [52]	NR	Yes	No	Yes	Yes
Chen et al. [12]	4-fold	Yes	No	Yes	Yes
Colone et al. [53]	5-fold	Yes	Yes	Yes	Yes
Dangut et al. [54]	NR	Yes	No	No	Yes
Figueroa Barraza et al. [55]	5-fold	Yes	Yes	Yes	Yes
Hamaide and Glineur [56]	9-fold	Yes	No	Yes	Yes
Jansen et al. [57]	NR	Yes	NR	Yes	No
Kaparthi and Bumblauskas [9]	No	Yes	No	Yes	Yes
Korhoseed and Beyca [58]	5-fold	Yes	No	Yes	Yes
Kolokas et al. [10]	No	Yes	No	Yes	Yes
Kulkarni et al. [59]	No	No	Yes	Yes	Yes
Kusiak and Verma [7]	10-fold	Yes	No	Yes	Yes
Leahy et al. [60]	NR	Yes	Yes	Yes	Yes
Lee et al. [61]	5-fold	Yes	Yes	No	Yes
Li et al. [62]	5-fold	Yes	Yes	Yes	Yes
Lüttenberg et al. [63]	4-fold	Yes	No	Yes	Yes
Mishra and Manjhi [64]	NR	NR	No	Yes	No
Nowaczyk et al. [65]	No	Yes	No	Yes	No
Orrù et al. [66]	3-fold	Yes	No	Yes	Yes
Pertselakis et al. [67]	5-fold	Yes	No	No	Yes
Proto et al. [49]	3-fold	Yes	No	No	Yes
Prytz et al. [68]	10-fold	Yes	No	Yes	Yes
Renga et al. [69]	No	Yes	No	No	No
Rombach and Keuper [70]	No	NR	No	Yes	Yes
Savitha et al. [8]	5-fold	Yes	No	No	Yes
Silva and Capretz [71]	5-fold	Yes	No	No	Yes
Susto et al. [72]	NR	Yes	No	Yes	Yes
Wang et al. [73]	No	Yes	No	Yes	Yes
Wijs et al. [74]	5-fold	Yes	No	No	Yes
Xiang et al. [75]	10-fold	Yes	No	Yes	Yes
Yu et al. [76]	NR	Yes	No	Yes	Yes
Count	18	30	7	26	28

Note. NR = Not reported.

Every fifth study evaluated the prediction model separately on unknown data (test set, $n = 7$). Overall, the results were reported using various types of charts ($n = 26$) and tables ($n = 28$).

3.5.3. Performance metrics

Table 8 summarizes the adoption of performance metrics and gives the quantitative results. For each study and metric, we coded the best performance across all experimental conditions but we did not include the specific condition.

The accuracy (ACC) metric, which is defined as the share of correct predictions among all predictions, ranged from 73.0 to 99.0% ($n = 16$). Nevertheless, it should be noted that ACC is an inappropriate measure for imbalanced classification problems: An extremely high ACC can be achieved by making correct predictions of the majority class (non-failure), even if the model performs extremely worse in predicting the minority class (failure). Therefore, accuracy needs to

be differentiated for the minority and majority classes, for which precision, recall, and specificity, among other metrics, are available [48].

Precision (PRE) represents the share of correct predictions among all failure predictions, and it ranged from 34.0 to 96.0% (n = 20). In general, precision should be high, because false failure predictions may cause unnecessary time and effort for verifying the current system condition. This expectation was not supported, with only 11 studies reporting a PRE greater than 80%.

Table 8. Performance metrics and percentages reported in studies (N = 34).

Study	ACC	PRE	REC	SPE	FSC	AUC	Other
Abu-Samah et al. [50]							x
Alves et al. [11]	99.0						
Bonnevay et al. [51]	89.0						x
Canizo et al. [52]	82.0		92.3	60.6			
Chen et al. [12]	86.0	83.0	73.0		78.0	91.0	
Colone et al. [53]						94.7	
Dangut et al. [54]		88.0	66.0				x
Figueroa Barraza et al. [55]		90.8	90.8	89.0	99.5		
Hamaide and Glineur [56]		96.0	78.0				x
Jansen et al. [57]	82.0	55.0	83.0		66.0		
Kaparthi and Bumblauskas [9]	79.4						
Korhoseed and Beyca [58]	92.3	92.6	89.5			92.8	
Kolokas et al. [10]	98.3	73.4	74.4	99.1	73.9		x
Kulkarni et al. [59]		89.0	46.0				
Kusiak and Verma [7]	99.5						x
Leahy et al. [60]		46.0	51.0				
Lee et al. [61]		72.9	79.5	99.1	87.8		x
Li et al. [62]			99.8				
Lüttenberg et al. [63]	89.4	88.4	92.1	88.1		97.1	
Mishra and Manjhi [64]		65.0	80.0				
Nowaczyk et al. [65]					46.0		
Orrù et al. [66]	98.2	71.1	27.7	99.7	39.9		x
Pertselakis et al. [67]	95.0						
Proto et al. [49]		84.3	83.0		82.2		
Prytz et al. [68]	73.0				23.0		x
Renga et al. [69]		34.0	54.0				
Rombach and Keuper [70]		77.9	96.2				x
Savitha et al. [8]			98.7	100.0			x
Silva and Capretz [71]	98.1	96.0	96.0		95.9		
Susto et al. [72]	98.5	69.3	100.0				
Wang et al. [73]		82.2	49.3			81.8	
Wijs et al. [74]			58.0	100.0		74.0	x
Xiang et al. [75]	88.1	86.5	76.2		81.0		
Yu et al. [76]							x
Count	16	20	24	8	11	6	13

Note. ACC = accuracy. AUC = area under the curve. FSC = F-score. PRE = precision. REC = recall. SPE = specificity.

Recall (REC) is defined as the proportion of failures that were correctly predicted. REC should be high to avoid running into unexpected failures and downtimes. It was larger than 80% in 11 of 24 relevant studies; hence, these studies overlooked less than every fifth failure. One-third of the relevant studies missed less than every tenth failure (REC larger than 90%; n = 8). In eight studies,

recall was complemented with specificity (SPE), defined as the proportion of non-failures that were correctly predicted. Due to the extreme data imbalance, predicting the non-failures was highly effective in five studies (> 99%).

The performance of a prediction model can collectively be measured using the F-score, which is calculated from PRE and REC. For imbalanced datasets, achieving higher precision often comes at the cost of lower recall, and vice versa. In this sense, the F-score metric balances the different information conveyed by PRE and REC as class-specific metrics. Surprisingly, only one-third of the studies reported the F-score (n = 11). In five studies, F-score was greater than 80%. Nine studies provided F-score, PRE, and REC to paint a comprehensive picture of performance. Six studies reported results for Area Under the Curve (AUC), which measures the ability of a classifier to distinguish between the minority and majority classes. Using standard cut-offs [77], one classifier can be considered as acceptable (AUC larger than 0.7) [74], one classifier as excellent (> 0.8) [73], and four classifiers as outstanding (> 0.9) [12,53,58,63].

4. Discussion

The systematic review of ML technology for failure prediction assessed the adoption of specific ML techniques in previous research and their role in developing effective prediction models. In this section, we discuss the principal findings and implications for future research as well as the limitations of our review.

4.1. Principal findings and implications

Collectively, the studies included in this review address failure prediction for a highly diverse set of material systems in various domains, with very few studies examining similar systems. This diversity also mirrors in the prediction windows used (i.e., from a few seconds to several months), because different systems require different minimum times to perform maintenance actions. On one hand, these results demonstrate the applicability of ML technology for failure prediction in manifold contexts. On the other hand, the variation of application contexts has resulted in a large but fragmented body of knowledge.

4.1.1. Data collection

With respect to the datasets analyzed, our review indicates the practice of collecting data over a long period of time, usually many months or several years. Because learning is only possible if many example pairs of input-output for the same failure type are present, the time period increases for less frequent failures and greater data imbalance. Surprisingly, 38 percent of the studies did not report the number of failures, although this information is essential for determining whether learning a prediction model is feasible at all. To make matters worse, four studies used datasets that included less than ten failures [8,56,66,76]; such small sizes of the minority class make estimates of the precision for the minority class impossible, and thus evidence for the performance of the

trained prediction model cannot be derived from these studies.

As in any empirical research, a detailed specification of the dataset is necessary to assess the generalizability of the results. Our review highlights that a relevant subset of studies do not give information about attributes, sampling rates, and records, while this information exhibits considerable variation in the remaining studies. Examples of detailed reporting are the study by Figueroa Barraza [55], which provides definitions of all attributes and their units of measurement, and the study by Wang et al. [73], which reports the number of records differentiated for the training and evaluation phases. Only one study manipulated the dataset size, but its prediction performance was very low, with the F-Score being smaller than 0.35 [65]. Opportunities exist to further examine the role of the dataset size, which can then help ascertain the amount of data that must be collected for a specific context.

Although our review assessed the reported datasets for abstract data characteristics (e.g., number of attributes, data types), the range of data available to failure prediction has broadened by also including data from service processes and human interactions with systems [78]. In this social manufacturing context, ML techniques need to be adapted that allow recognizing the knowledge present in very different data attributes (granular computing perspective) [79].

4.1.2. Data preprocessing

Regarding the preprocessing of data, feature selection is an important part of ML technology and the specific techniques used can have substantial effects by focusing on relevant attributes and reducing noise in the data [80]. Our results reveal that the adoption of feature selection is rather low, with no information about feature selection being available from one half of the studies. The other half of the studies adopted a wide array of techniques, demonstrating awareness of the design alternatives available from the literature. Indeed, four of the five studies that tested different techniques found positive effects on prediction performance. Given this evidence, the low level of adoption points to a missed opportunity in ML-based failure prediction. Future research should examine the role of feature selection by contrasting different techniques and varying the number of features. Moreover, researchers are advised to clearly state whether feature selection has been applied and eventually specify the techniques used. For instance, the study by Prytz et al. [68] both provides a detailed description of seven different techniques, including the rationale for their adoption, and an evaluation using a three-factor experiment. An alternative to feature selection is feature learning, which automatically learns features using unsupervised ML algorithms. Recent research has shown that feature learning is effective for fault detection [81], fault diagnosis [82] and degradation stage classification [83]. It would be interesting to empirically test whether feature learning can improve failure prediction.

4.1.3. Model training

With respect to the training of prediction models, we found the highest adoption rates for RF, SVM, ANN, and decision tree. This observation corroborates results of previous reviews examining

related tasks in industrial maintenance [14,15]. The prevalence of these algorithms is coincident with two-thirds of studies comparing the performance of different algorithms. However, few studies observed performance differences (except for RF and gradient boosting). Collectively, there is no evidence for one group of algorithms being superior to another group. This finding also holds true for hyperparameter optimization, which has been adopted in one half of the studies but evaluated in only three studies.

The training of prediction models is made difficult by imbalanced data, with the failure class representing a small fraction of all system states. Learning from imbalanced data is an active field of ML research, and differentiated sampling techniques have been developed and experimentally evaluated [84,85], including task-specific oversampling techniques for fault detection in industrial maintenance [86]. Despite the design knowledge from this field, only one-fourth of studies adopted any sampling technique. It is noteworthy that four studies tested the impact of oversampling, and three of these studies found positive effects on performance. We believe that this evidence provides further support to the importance of data sampling. For adopting data sampling, we recommend to report the specific sampling rate along with the resulting number of failures and non-failures in the training set (a good example is the reporting by Wijs et al. [74]). As data sampling originates from classification tasks, a potentially useful alternative is data augmentation for for time-series data and forecasting tasks [87].

4.1.4. Model evaluation

Insights into the design of effective prediction models can be obtained from rigorous evaluations of the most relevant design alternatives. However, the sheer number of factors that potentially affect performance make the evaluation challenging. Our review substantiates this challenge through identifying nine groups of factors (Table 6). Nevertheless, the variety of data collection, preprocessing, and training undermines the ability to compare results for specific factors *between* studies.

A standard evaluation technique is cross-validation, which also allows detecting the overfitting problem in learned prediction models. Overfitting characterizes a model that fits well to the training data but actually fails to learn the dataset features well, and thus performs worse on unknown data. In other words, when deployed in a productive system, the performance in predicting failures will be much lower, and the model might become useless. Our review uncovers that the adoption rate of cross-validation in the failure prediction literature is rather low at 53%. Therefore, it is likely that some of the proposed models suffer from overfitting.

An important assumption of cross-validation is that the prediction performance for each fold should be very similar. This assumption can easily be verified by not only reporting the mean value calculated for all folds but presenting further distribution parameters (e.g., standard deviation, minimum, maximum), and reporting the specific results for each fold, as demonstrated in the study by Xiang et al. [75].

Model evaluation using a *separate* dataset of unknown data could be regarded as the "gold standard". Although seven studies chose that approach, its implementation will often be overly

difficult or not possible at all. The additional effort required for collecting operational data could exceed the resources and time available. Specifically, the longer time needed might delay the intended deployment of the model. Against this backdrop, cross-validation as a technique that systematically derives unknown data from the training set and integrates it in the evaluation takes on even greater importance.

With respect to the adoption of standard performance metrics, our review reveals that many studies relied upon metrics that are either inadequate for imbalanced data (accuracy, specificity) or only represent one facet of prediction performance (e.g., recall). On the other end of the spectrum, nine of the studies comprehensively assessed performance by reporting precision, recall, and F-score. Considering that seven of the nine studies adopted cross-validation of which one study lacked a sufficient number of failures, six studies ranked highest in the quality of the evaluation [12,49,55,61,71,75]. For this set of studies, prediction performance was rather high as signified by the lower bounds and upper bounds for precision [0.729; 0.960], recall [0.730; 0.960] and F-score [0.810; 0.995]. Although a quantitative synthesis of results is beyond the scope of our review, these indicative results demonstrate the usefulness of adopting comprehensive metrics specific to the context of imbalanced data. Therefore, we suggest to: (1) perform cross-validation with the number of folds adjusted to the dataset size; (2) abandon the inadequate and misleading metrics accuracy and specificity; and (3) report precision, recall and F-score for all experimental conditions. Moreover, complementing the composite metric F-score with AUC enables a context-indepenent interpretation using standard cut-offs.

4.2. Limitations

The results of our review should be viewed in light of the following limitations. The first limitation is the heterogeneity of the included studies. Although we defined strict eligibility criteria, the studies were very heterogeneous with regard to data collection, preprocessing, training, and evaluation; hence, a direct comparison of outcomes using performance metrics was not possible. Second, the included studies exhibited rather few common reporting practices, which made the data extraction and synthesis intricate. Data extraction was further undermined by some studies that reported charts but no exact quantitative results. Given these challenges, we ensured the validity of the extracted data through using two independent reviewers and resolving inconsistencies by discussion with a third reviewer. Third, the data items of our review did not include quality criteria, such as justification of conducting the study, generalizability of the dataset used, and validity of conclusions. We applied a single proxy measure for quality (published in a refereed journal or conference proceedings), whereas all data items were specific to the adoption of ML technology.

5. Conclusion

This systematic review provides insights into the richness and depth of the published research adopting ML technology for failure prediction in industrial maintenance. The review's conceptual

model – organized into data collection, data preprocessing, model training, and model evaluation – allowed us to assess the level of adoption and identify important contributions as well as opportunities for future research. The review results reveal that barriers towards the accumulation of knowledge about the development of ML-based failure prediction models and their performance evaluation continue to exist. As our review of thirty-four studies shows, diversity of ML designs is coincident with considerable heterogeneity in the reporting. We suggest specific recommendations for future research including reporting items that can enhance uniformity and comparability of studies. The new practices will assist in interpreting and comparing results obtained from single studies, and pave the road for improved synthesis of quantitative evidence. Overall, we believe that these efforts will help in enhancing knowledge about how failure prediction can benefit from advances in ML technology.

Declaration of Competing Interests

None.

Acknowledgment

This work was supported by the Federal Ministry for Economic Affairs and Energy [grant: 01MT19005D] and the Federal Ministry of Food and Agriculture [grant: 28DE106A18], Germany.

References

[1] Jonge B de, Teunter R, Tinga T. The influence of practical factors on the benefits of condition-based maintenance over time-based maintenance. Reliab Eng Sys Safe 2017;158(2):21–30. https://doi.org/10.1016/j.ress.2016.10.002.

[2] McKone KE, Weiss EN. Guidelines for implementing predictive maintenance. Prod Oper Manage 2002;11(2):109–24. https://doi.org/10.1111/j.1937-5956.2002.tb00486.x.

[3] Jordan MI, Mitchell TM. Machine learning: Trends, perspectives, and prospects. Science 2015;349(6245):255–60. https://doi.org/10.1126/science.aaa8415.

[4] Wang J, Xu C, Zhang J, Zhong R. Big data analytics for intelligent manufacturing systems: A review. J Manuf Syst 2021. https://doi.org/10.1016/j.jmsy.2021.03.005.

[5] Sahal R, Breslin JG, Ali MI. Big data and stream processing platforms for industry 4.0 requirements mapping for a predictive maintenance use case. J Manuf Syst 2020;54:138–51. https://doi.org/10.1016/j.jmsy.2019.11.004.

[6] Rajakumar MP, Ramya J, Maheswari BU. Health monitoring and fault prediction using a lightweight deep convolutional neural network optimized by Levy flight optimization algorithm. Neural Comput & Applic 2021. https://doi.org/10.1007/s00521-021-05892-0.

[7] Kusiak A, Verma A. A data-mining approach to monitoring wind turbines. IEEE Trans Sustain Energy 2012;3(1):150–7. https://doi.org/10.1109/TSTE.2011.2163177.

[8] Savitha R, Ambikapathi A, Rajaraman K. Online RBM: Growing restricted boltzmann machine on the fly for unsupervised representation. Appl Soft Comput 2020;92:106278. https://doi.org/10.1016/j.asoc.2020.106278.

[9] Kaparthi S, Bumblauskas D. Designing predictive maintenance systems using decision tree-based machine learning techniques. Int J Qual Reliab Manage 2020;37(4):659–86. https://doi.org/10.1108/IJQRM-04-2019-0131.

[10] Kolokas N, Vafeiadis T, Ioannidis D, Tzovaras D. A generic fault prognostics algorithm for manufacturing industries using unsupervised machine learning classifiers. Simul Model Pract Theor 2020;103:102109. https://doi.org/10.1016/j.simpat.2020.102109.

[11] Alves F, Badikyan H, Antonio Moreira HJ, Azevedo J, Moreira PM, Romero L et al. Deployment of a smart and predictive maintenance system in an industrial case study. In: Proc. 2020 IEEE 29th Int. Symp. Ind. Electron. ISIE; 2020, p. 493–498.

[12] Chen K, Pashami S, Fan Y, Nowaczyk S. Predicting air compressor failures using long short term memory networks. In: Moura Oliveira P, Novais P, Reis LP, editors. Progress in artificial intelligence. EPIA 2019. Cham: Springer; 2019, p. 596–609.

[13] Çınar ZM, Abdussalam Nuhu A, Zeeshan Q, Korhan O, Asmael M, Safaei B. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. Sustain 2020;12(19):8211. https://doi.org/10.3390/su12198211.

[14] Carvalho TP, Soares FAAMN, Vita R, Francisco RdP, Basto JP, Alcalá SGS. A systematic literature review of machine learning methods applied to predictive maintenance. Comput Ind Eng 2019;137:106024. https://doi.org/10.1016/j.cie.2019.106024.

[15] Zhang W, Yang D, Wang H. Data-driven methods for predictive maintenance of industrial equipment: A survey. IEEE Sys J 2019;13(3):2213–27. https://doi.org/10.1109/JSYST.2019.2905565.

[16] Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX. Deep learning and its applications to machine health monitoring. Mech Syst Sig Process 2019;115:213–37. https://doi.org/10.1016/j.ymssp.2018.05.050.

[17] Montero Jimenez JJ, Schwartz S, Vingerhoeds R, Grabot B, Salaün M. Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. J Manuf Syst 2020;56:539–57. https://doi.org/10.1016/j.jmsy.2020.07.008.

[18] Dalzochio J, Kunst R, Pignaton E, Binotto A, Sanyal S, Favilla J et al. Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges. Comput Ind 2020;123(9–12):103298. https://doi.org/10.1016/j.compind.2020.103298.

[19] Stetco A, Dinmohammadi F, Zhao X, Robu V, Flynn D, Barnes M et al. Machine learning methods for wind turbine condition monitoring: A review. Renew Energ 2019;133:620–35. https://doi.org/10.1016/j.renene.2018.10.047.

[20] Lei Y, Yang B, Jiang X, Jia F, Li N, Nandi AK. Applications of machine learning to machine fault diagnosis: A review and roadmap. Mech Syst Sig Process 2020;138:106587. https://doi.org/10.1016/j.ymssp.2019.106587.

[21] Liu R, Yang B, Zio E, Chen X. Artificial intelligence for fault diagnosis of rotating machinery: A review. Mech Syst Sig Process 2018;108:33–47. https://doi.org/10.1016/j.ymssp.2018.02.016.
[22] Moher D, Liberati A, Tetzlaff J, Altman DG. Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. PLoS Med 2009;6(7):e1000097. https://doi.org/10.1371/journal.pmed.1000097.
[23] Mongeon P, Paul-Hus A. The journal coverage of Web of Science and Scopus: a comparative analysis. Scientometrics 2016;106(1):213–28. https://doi.org/10.1007/s11192-015-1765-5.
[24] Zonta T, da Costa CA, da Rosa Righi R, Lima MJ de, da Trindade ES, Li GP. Predictive maintenance in the industry 4.0: A systematic literature review. Comput Ind Eng 2020;150(6):106889. https://doi.org/10.1016/j.cie.2020.106889.
[25] Kiangala KS, Wang Z. Initiating predictive maintenance for a conveyor motor in a bottling plant using industry 4.0 concepts. Int J Adv Manuf Technol 2018;97(9-12):3251–71. https://doi.org/10.1007/s00170-018-2093-8.
[26] Lee J, Jin C, Bagheri B. Cyber physical systems for predictive production systems. Prod. Eng. Res. Devel. 2017;11(2):155–65. https://doi.org/10.1007/s11740-017-0729-4.
[27] Matyas K, Nemeth T, Kovacs K, Glawar R. A procedural approach for realizing prescriptive maintenance planning in manufacturing industries. CIRP Ann 2017;66(1):461–4. https://doi.org/10.1016/j.cirp.2017.04.007.
[28] Welte R, Estler M, Lucke D. A method for implementation of machine learning solutions for predictive maintenance in small and medium sized enterprises. Procedia CIRP 2020;93:909–14. https://doi.org/10.1016/j.procir.2020.04.052.
[29] Huang Y-C, Kao C-H, Chen S-J. Diagnosis of the hollow ball screw preload classification using machine learning. Appl Sci 2018;8(7):1072. https://doi.org/10.3390/app8071072.
[30] Yu J. Machine health prognostics using the Bayesian-inference-based probabilistic indication and high-order particle filtering framework. J Sound Vib 2015;358:97–110. https://doi.org/10.1016/j.jsv.2015.08.013.
[31] Luo B, Wang H, Liu H, Li B, Peng F. Early fault detection of machine tools based on deep learning and dynamic identification. IEEE Trans Ind Electron 2019;66(1):509–18. https://doi.org/10.1109/TIE.2018.2807414.
[32] Shao H, Jiang H, Lin Y, Li X. A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. Mech Syst Sig Process 2018;102:278–97. https://doi.org/10.1016/j.ymssp.2017.09.026.
[33] Martin-del-Campo S, Sandin F. Online feature learning for condition monitoring of rotating machinery. Eng Appl Artif Intell 2017;64:187–96. https://doi.org/10.1016/j.engappai.2017.06.012.
[34] Costello JJA, West GM, McArthur SDJ. Machine learning model for event-based prognostics in gas circulator condition monitoring. IEEE Trans Rel 2017;66(4):1048–57. https://doi.org/10.1109/TR.2017.2727489.

[35] Gutschi C, Furian N, Suschnigg J, Neubacher D, Voessner S. Log-based predictive maintenance in discrete parts manufacturing. Procedia CIRP 2019;79:528–33. https://doi.org/10.1016/j.procir.2019.02.098.
[36] Lu C, Wang S. Performance degradation prediction based on a Gaussian mixture model and optimized support vector regression for an aviation piston pump. Sensors (Basel) 2020;20(14). https://doi.org/10.3390/s20143854.
[37] Wu W, Liu M, Liu Q, Shen W. A quantum multi-agent based neural network model for failure prediction. J Syst Sci Syst Eng 2016;25(2):210–28. https://doi.org/10.1007/s11518-016-5308-2.
[38] Crespo Márquez A, La Fuente Carmona A de, Antomarioni S. A process to implement an artificial neural network and association rules techniques to improve asset performance and energy efficiency. Energies 2019;12(18):3454. https://doi.org/10.3390/en12183454.
[39] Marcelino P, Lurdes Antunes M de, Fortunato E, Gomes MC. Machine learning approach for pavement performance prediction. Int J Pavement Eng 2021;22(3):341–54. https://doi.org/10.1080/10298436.2019.1609673.
[40] Han J, Pei J, Kamber M. Data mining: Concepts and techniques. 3rd ed. Waltham, MA: Morgan Kaufmann; 2012.
[41] Chandrashekar G, Sahin F. A survey on feature selection methods. Comput Electr Eng 2014;40(1):16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024.
[42] Chang C-C, Lin C-J. LIBSVM. ACM Trans Intell Sys Tech 2011;2(3):1–27. https://doi.org/10.1145/1961189.1961199.
[43] Bishop CM. Pattern recognition and machine learning. New York: Springer; 2006.
[44] Ho TK. Random decision forests. In: Proc. 3rd Int. Conf. Doc. Anal. Recognit. ICDAR. IEEE; 1995, p. 278–282.
[45] Luo G. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. Netw Model Anal Health Inform Bioinforma 2016;5(1). https://doi.org/10.1007/s13721-016-0125-6.
[46] He H, Garcia EA. Learning from imbalanced data. IEEE Trans Knowl Data Eng 2009;21(9):1263–84. https://doi.org/10.1109/TKDE.2008.239.
[47] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proc. 14th Int. J. Conf. Artif. Intell; 1995, p. 1137–1145.
[48] Tharwat A. Classification assessment methods. Appl Comput Inform 2021;17(1):168–92. https://doi.org/10.1016/j.aci.2018.08.003.
[49] Proto S, Ventura F, Apiletti D, Cerquitelli T, Baralis E, Macii E et al. PREMISES, a scalable data-driven service to predict alarms in slowly-degrading multi-cycle industrial processes. In: Proc. 2019 IEEE Int. Congr. Big Data. IEEE; 2019, p. 139–143.
[50] Abu-Samah A, Shahzad MK, Zamai E, Said A. Failure prediction methodology for improved proactive maintenance using bayesian approach. IFAC-PapersOnLine 2015;48(21):844–51. https://doi.org/10.1016/j.ifacol.2015.09.632.

[51] Bonnevay S, Cugliari J, Granger V. Predictive maintenance from event logs using wavelet-based features: An industrial application. In: Martínez Álvarez F, Troncoso Lora A, Sáez Muñoz JA, Quintián H, Corchado E, editors. 14th Int. Conf. Soft Comput. Mod. Ind. Environ. Appl. SOCO 2019. Cham: Springer; 2020, p. 132–141.
[52] Canizo M, Onieva E, Conde A, Charramendieta S, Trujillo S. Real-time predictive maintenance for wind turbines using big data frameworks. In: Proc. 2017 IEEE Int. Conf. Progn. Health Manage. ICPHM. IEEE; 2017, p. 70–77.
[53] Colone L, Dimitrov N, Straub D. Predictive repair scheduling of wind turbine drive-train components based on machine learning. Wind Energy 2019;22:1230–42. https://doi.org/10.1002/we.2352.
[54] Dangut M, Skaf Z, Jennions I. Rescaled-LSTM for predicting aircraft component replacement under imbalanced dataset constraint. In: Proc. 2020 Adv. Science Eng. Technol. Int. Conf. ASET. IEEE; 2020, p. 1–9.
[55] Figueroa Barraza J, Guarda Bräuning L, Benites Perez R, Morais CB, Martins MR, Droguett EL. Deep learning health state prognostics of physical assets in the oil and gas industry. Proc. Inst Mech Eng O J Risk Reliab 2020:1748006X2097681. https://doi.org/10.1177/1748006X20976817.
[56] Hamaide V, Glineur F. Predictive maintenance of a rotating condenser inside a synchrocyclotron. In: Proc. 28th Belgian Dutch Conf. Mach. Learn; 2019, p. 1–12.
[57] Jansen F, Holenderski M, Ozcelebi T, Dam P, Tijsma B. Predicting machine failures from industrial time series data. In: Proc. 2018 5th Int. Conf. Control, Decis. Inf. Technol. CoDIT. IEEE; 2018, p. 1091–1096.
[58] Khorsheed RM, Beyca OF. An integrated machine learning: Utility theory framework for real-time predictive maintenance in pumping systems. Proc. Inst Mech Eng B J Eng Manuf 2021;235(5):887–901. https://doi.org/10.1177/0954405420970517.
[59] Kulkarni K, Devi U, Sirighee A, Hazra J, Rao P. Predictive maintenance for supermarket refrigeration systems using only case temperature data. In: Proc. 2018 Ann. American Control Conf. ACC. IEEE; 2018, p. 4640–4645.
[60] Leahy K, Gallagher C, O'Donovan P, Bruton K, O'Sullivan D. A robust prescriptive framework and performance metric for diagnosing and predicting wind turbine faults based on SCADA and alarms data with case study. Energies 2018;11(7):1738. https://doi.org/10.3390/en11071738.
[61] Lee Y-L, Juan D-C, Tseng X-A, Chen Y-T, Chang S-C. DC-Prophet: Predicting catastrophic machine failures in datacenters. In: Altun Y, Das K, Mielikäinen T, Malerba D, Stefanowski J, Read J et al., editors. Machine learning and knowledge discovery in databases. ECML PKDD 2017. Cham: Springer; 2017, p. 64–76.
[62] Li H, Parikh D, He Q, Qian B, Li Z, Fang D et al. Improving rail network velocity: A machine learning approach to predictive maintenance. Transp Res Part C Emerg Technol 2014;45:17–26. https://doi.org/10.1016/j.trc.2014.04.013.

[63] Lüttenberg H, Bartelheimer C, Beverungen D. Designing predictive maintenance for agricultural machines. In: Proc. 26th Europ. Conf. Inf. Sys. ECIS 2018; 2018.
[64] Mishra K, Manjhi SK. Failure prediction model for predictive maintenance. In: Proc. 2018 IEEE Int. Conf. Cloud Comput. Emerg. Mark. CCEM. IEEE; 2018, p. 72–75.
[65] Nowaczyk S, Prytz R, Rognvaldsson T, Byttner S. Towards a machine learning algorithm for predicting truck compressor failures using logged vehicle data. In: Proc. 12th Scandinavian Conf. Artif. Intell. IOS Press; 2013.
[66] Orrù PF, Zoccheddu A, Sassu L, Mattia C, Cozza R, Arena S. Machine learning approach using MLP and SVM algorithms for the fault prediction of a centrifugal pump in the oil and gas industry. Sustain 2020;12(11):4776. https://doi.org/10.3390/su12114776.
[67] Pertselakis M, Lampathaki F, Petrali P. Predictive Maintenance in a digital factory shop-floor: Data mining on historical and operational data coming from manufacturers' information systems. In: Proper HA, Stirna J, editors. Advanced information systems engineering workshops. CAiSE 2019. Cham: Springer; 2019, p. 120–131.
[68] Prytz R, Nowaczyk S., Rögnvaldsson T, Byttner S. Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. Eng Appl Artif Intell 2015;41:139–50. https://doi.org/10.1016/j.engappai.2015.02.009.
[69] Renga D, Apiletti D, Giordano D, Nisi M, Huang T, Zhang Y et al. Data-driven exploratory models of an electric distribution network for fault prediction and diagnosis. Comput 2020;102(5):1199–211. https://doi.org/10.1007/s00607-019-00781-w.
[70] Rombach P, Keuper J. SmartPred: Unsupervised hard disk failure detection. In: Jagode H, Anzt H, Juckeland G, Ltaief H, editors. High performance computing. ISC High Performance 2020. Cham: Springer; 2020, p. 235–246.
[71] Silva W, Capretz M. Assets predictive maintenance using convolutional neural networks. In: Proc. 2019 20th IEEE/ACIS Int. Conf. Softw. Eng., Artifi. Intell., Netw. Parallel/Distrib. Comput. SNPD. IEEE; 2019, p. 59–66.
[72] Susto GA, Schirru A, Pampuri S, McLoone S, Beghi A. Machine learning for predictive maintenance: A multiple classifier approach. IEEE Trans Ind Inf 2015;11(3):812–20. https://doi.org/10.1109/TII.2014.2349359.
[73] Wang J, Li C, Han S, Sarkar S, Zhou X. Predictive maintenance based on event-log analysis: A case study. IBM J Res Dev 2017;61(1):11:121-11:132. https://doi.org/10.1147/JRD.2017.2648298.
[74] Wijs RJA, Nane GF, Leontaris G, van Manen TRW, Wolfert ARM. Improving subsurface asset failure predictions for utility operators: A unique case study on cable and pipe failures resulting from excavation work. ASCE-ASME J Risk Uncertain Eng Sys A 2020;6(2):5020002. https://doi.org/10.1061/AJRUA6.0001063.
[75] Xiang S, Huang D, Li X. A generalized predictive framework for data driven prognostics and diagnostics using machine logs. In: Proc. 2018 IEEE Region 10 Conf. TENCON. IEEE; 2018, p. 695–700.

[76] Yu W, Dillon T, Mostafa F, Rahayu W, Liu Y. A global manufacturing big data ecosystem for fault detection in predictive maintenance. IEEE Trans Ind Inf 2020;16(1):183–92. https://doi.org/10.1109/TII.2019.2915846.
[77] Hosmer DW, Lemeshow S. Applied logistic regression. 2nd ed. New York: John Wiley & Sons; 2000.
[78] Leng J, Jiang P. Granular computing–based development of service process reference models in social manufacturing contexts. Concurr Eng 2017;25(2):95–107. https://doi.org/10.1177/1063293X16666312.
[79] Leng J, Chen Q, Mao N, Jiang P. Combining granular computing technique with deep learning for service planning under social manufacturing contexts. Knowl Based Syst 2018;143:295–306. https://doi.org/10.1016/j.knosys.2017.07.023.
[80] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res 2003:1157–82.
[81] Zhang T, Ding B, Zhao X, Liu G, Pang Z. LearningADD: Machine learning based acoustic defect detection in factory automation. J Manuf Syst 2021;60:48–58. https://doi.org/10.1016/j.jmsy.2021.04.005.
[82] Ye Z, Yu J. AKSNet: A novel convolutional neural network with adaptive kernel width and sparse regularization for machinery fault diagnosis. J Manuf Syst 2021;59:467–80. https://doi.org/10.1016/j.jmsy.2021.03.022.
[83] Alfeo AL, Cimino MG, Vaglini G. Degradation stage classification via interpretable feature learning. J Manuf Syst 2021. https://doi.org/10.1016/j.jmsy.2021.05.003.
[84] Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G. Learning from class-imbalanced data: Review of methods and applications. Expert Syst Appl 2017;73:220–39. https://doi.org/10.1016/j.eswa.2016.12.035.
[85] López V, Fernández A, García S, Palade V, Herrera F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Inf Sci 2013;250:113–41. https://doi.org/10.1016/j.ins.2013.07.007.
[86] Zhang Y, Li X, Gao L, Wang L, Wen L. Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning. J Manuf Syst 2018;48:34–50. https://doi.org/10.1016/j.jmsy.2018.04.005.
[87] Bandara K, Hewamalage H, Liu Y-H, Kang Y, Bergmeir C. Improving the accuracy of global forecasting models using time series data augmentation. Pattern Recognit 2021;120:108148. https://doi.org/10.1016/j.patcog.2021.108148.