Received: 8 March 2022 | Accepted: 31 May 2022

DOI: 10.1002/widm.1471

ADVANCED REVIEW

WIREs DATA MINING AND KNOWLEDGE DISCOVERY WILEY

Data mining in predictive maintenance systems: A taxonomy and systematic review

Aurora Esteban | **Amelia Zafra** | **Sebastián Ventura**

Department of Computer Science and Numerical Analysis, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Cordoba, Cordoba, Spain

Correspondence
Sebastián Ventura, Department of Computer Science and Numerical Analysis, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Cordoba, Cordoba 14071, Spain.
Email: sventura@uco.es

Funding information
Ministerio de Ciencia e Innovación, Grant/Award Numbers: PID2020-115832GB-I00, FPU19/03924

Edited by: Witold Pedrycz, Editor-in-Chief

Abstract

Predictive maintenance is a field of study whose main objective is to optimize the timing and type of maintenance to perform on various industrial systems. This aim involves maximizing the availability time of the monitored system and minimizing the number of resources used in maintenance. Predictive maintenance is currently undergoing a revolution thanks to advances in industrial systems monitoring within the Industry 4.0 paradigm. Likewise, advances in artificial intelligence and data mining allow the processing of a great amount of data to provide more accurate and advanced predictive models. In this context, many actors have become interested in predictive maintenance research, becoming one of the most active areas of research in computing, where academia and industry converge. The objective of this paper is to conduct a systematic literature review that provides an overview of the current state of research concerning predictive maintenance from a data mining perspective. The review presents a first taxonomy that implies different phases considered in any data mining process to solve a predictive maintenance problem, relating the predictive maintenance tasks with the main data mining tasks to solve them. Finally, the paper presents significant challenges and future research directions in terms of the potential of data mining applied to predictive maintenance.

This article is categorized under:

Application Areas > Industry Specific Applications
Technologies > Internet of Things

KEYWORDS

literature survey, machine learning, predictive maintenance, systematic review

1 | INTRODUCTION

In recent years, the industry has been attracted to Artificial Intelligence and data Mining (DM) because these fields of study have proven to be effective in processing the large amounts of data generated by current industrial systems. Maintenance of these systems is a crucial task due to the costs that undetected failures may entail both in economic terms and in the company's positioning. In this line, three major strategies are considered in system maintenance: Corrective maintenance (CM), Preventive Maintenance (PvM) and Predictive Maintenance (PdM). CM attends the fault once it has occurred. PvM schedules periodical maintenance actions based on operating hours. Finally, PdM seeks to optimize the maintenance actions when they are really necessary based on system monitoring and predictive techniques.

Currently, most industrial systems rely on PvM, but this is not the most efficient technique for industry. Sometimes, maintenance is done when it is not yet needed and other times it is carried out too late. Latest advances in real-time monitoring and computing systems are leading the industry toward its fourth revolution. Industry 4.0 has emerged as a new production paradigm that integrates physical and digital systems in manufacturing environments (Carvalho et al., 2019). Integrating these environments allows collecting a large amount of data about processes, events, and alarms along the industrial production line. In this context, PdM arises as an appealing paradigm able to detect early failures using predictive tools and the historical data of the monitored system. Thus, maintenance is carried out when it really is necessary.

The interest in PdM has result in numerous research works in last decade, and specifically from 2018, that apply DM predictive techniques to different applications: motors (Aremu, Hyland-Wood, & McAree, 2020; Pang et al., 2020; Wang, Zhang, et al., 2020), manufacturing plants (Axenie et al., 2020; Yu et al., 2020), medical equipment (Shamayleh et al., 2020), energy production plants (de Carvalho Chrysostomo et al., 2020; Gohel et al., 2020) or vehicle fleets (C. Chen et al., 2020; Oh & Lee, 2020) among others. However, there is not a direct application of DM to PdM. On the contrary, it depends on the nature of the data, the frequency of the monitoring, and the PdM task to perform (Angelopoulos et al., 2020). Thus, to obtain satisfactory results, PdM from DM perspective must take into account each specific industrial environment to which it is to be applied. Depending on the objective to solve, the nature and amount of data available, the desired response speed, the level of interpretability of the predictions, or the sensitivity in the detection of possible faults, not only a specific predictive model will be required, but also a specific data preprocessing. This paper presents a review that allows to find at a glance the works based on the type of problem and information available. For it, the most relevant works in the area of PdM from a DM perspective are analyzed and classified.

Due to the relevance of PdM in recent years, several reviews of the evolution of the field are published from different approaches. Most of them from the point of view of the industry, such as PdM procedures and standards (Zonta et al., 2020), data acquisition and sensors (Namuduri et al., 2020; Zhang, Yang, & Wang, 2019). While others are focus on Machine Learning (ML) and Deep Learning (DL) techniques (Angelopoulos et al., 2020; Carvalho et al., 2019). However, the present work provides a novel approach that addresses the entire PdM process from a DM perspective, including the following contributions:

1. It has carried out an exhaustive review of DM tasks applied to PdM between the years 2015 and 2021. This review proposes a taxonomy based on three main DM steps to solve a specific problem. It has covered 132 articles, which are analyzed exhaustively and included in the taxonomy:
   a. *Data acquisition*: the first step in PdM concerns specific industrial domains. This step depends on the monitored system and how to extract the data from the physical medium.
   b. *Data preprocessing*: the second step in PdM addresses the data preparation to improve predictions quality. Sometimes there is a high volume of data that is difficult to label. Thus, specific data transformations are needed to improve the performance of models used in the next step.
   c. *Model building*: the third step in PdM concerns to solve one of their specific PdM problems. Data are analyzed employing ML methods to discover new knowledge to detect possible anomalies, failures, or remaining useful life (RUL) of machinery.
2. The proposed taxonomy relates learning paradigms with DM tasks and PdM problems. Previous reviews do not explore these relationships, and it is essential to know the state of the art that solves different PdM problems from DM perspective:
   a. Supervised learning is related to regression and classification tasks. They are applied to PdM problems such as RUL estimation, prediction of gradual states of degradation, or prediction of failures. In this context, it is necessary that data is labeled and there is previous knowledge of correct operation and faults.
   b. Unsupervised learning is related to clustering tasks. PdM problems such as identification of different working conditions employ this paradigm. In this context, data are unlabeled, and there is no previous knowledge of correct operation and faults.

   c. Semi-supervised learning is related to one-class classification and novelty detection tasks. They are applied to PdM problems where failures are rare and infrequent. In this context, data is labeled, but there are not examples of faults in the system.
3. It is gathered a list of software tools and libraries with methods useful for applying DM to solve PdM problems.
4. It is compiled a list of public datasets used to test PdM solutions in a common framework.
5. An analysis of the latest trends and the current challenges in PdM from a data mining perspective is carried out.

The rest of this article is structured as follows: in Section 2, main terms and considerations in PdM are presented, as well as the analysis of previous reviews of the field. Section 3 explains the objective, the procedure, and the articles included in the systematic literature search performed for this work. Section 4 presents the results of the revision, including the taxonomy for data acquisition, data preparation, and model building in PdM from a DM perspective. In Section 5, available datasets and programming tools for PdM are commented. Section 6 indicates the trends of the latest works and the most significant challenges facing the DM in PdM. Finally, Section 7 analyzes the general conclusions obtained.

2 | RELATED WORK

This section addresses the definition of PdM in the context of the different maintenance management methods and their impact on the industry. Moreover, due to the tremendous attention attracted by this area, this section analyzes previous reviews and shows the main motivations and novelties to carry out this work.

2.1 | Predictive maintenance

Industrial systems of all kinds need maintenance that guarantees their correct operation: from industrial equipment in a manufacturing plant to a vehicle fleet or a railway grid. Thus, the impact of the maintenance costs can represent between 15% and 60% of the cost of goods produced (Mobley, 2002). The solution of companies to avoid, as far as possible, unexpected failures that paralyze their industrial activity is the performance of periodic maintenance in search of incipient failures. Thus, one-third of all maintenance costs are estimated to waste on unnecessary or improperly carried out maintenance (Mobley, 2002). In this context, PdM appears as a solution based on analyzing the available data, both historical and in real time, to anticipate maintenance interventions for machinery, equipment, vehicles, and systems of all kinds only when they are necessary.

To understand the relevance of PdM, it is important to contextualize the three main maintenance management strategies that have been defined (Mobley, 2002):

- *Corrective Maintenance*, *Reactive Maintenance*, or *Run to failure*: This is the simplest maintenance strategy and does not require any previous planning. This reactive method waits for equipment failure to take the maintenance. CM is also the most expensive strategy because of the high cost of spare parts inventory, overtime labor, and system downtime.
- *Preventive Maintenance*, *Time-based Maintenance*, or *Scheduled Maintenance*: This is the most widespread strategy because it is generally effective and does not require real-time monitoring of the system. PvM performs the maintenance periodically based on hours or cycles of operations. Thus, the mean-time-between-failure (MTBF) statistic indicates that, for a given machine under a normal operating life, its probability of failure increases after a fixed time. Therefore, the maintenance action should schedule when this probability exceeds a certain threshold. However, the mode of operation of a system directly affects its normal operating life, so commonly unnecessary corrective actions are taken, leading to increased operating costs.
- *Predictive Maintenance or Condition-based Maintenance*: This is the strategy that is currently attracting the most attention. Its purpose is to optimize the maintenance actions by applying them at the right moment. The premise of PdM is that regular monitoring of the current mechanical conditions, operating efficiency, and other indicators of the operating condition of machine will provide the data required to ensure the optimization of the resources destined for maintenance. Predictive tools process these historical data to perform early detection of the failures. Based on these predictions, maintenance actions are scheduled to minimize associated costs. Different predictive tools have

been studied to generate predictions since the 1970s, starting with statistical models and expert systems. However, today the area is being boosted thanks to advances in monitoring through sensor technologies and models based on ML that can learn from data extracted (Carvalho et al., 2019).

Figure 1 shows an outline of the different strategies based on the evolution of the current working condition of the system and when to perform the maintenance. The first segment shows the PvM strategy: the maintenance action is scheduled based on the MTBF, that is, using the mean degradation estimated for the monitored system. Thus, the maintenance action performed came too early (the real condition is not close to the failure threshold). However, during the second cycle, there is a high degradation. Thus the failure comes before the scheduled maintenance, and a CM action must be applied. The third segment illustrates a PdM strategy: it is shown both the actual evolution of the working condition together with a predicted signal that estimates the future state of the system in advance. In this strategy, the maintenance action is hold when the estimated signal reaches the failure threshold. Thus, thanks to the real-time monitoring of the system evolution and the use of predictive tools, the maintenance is optimized based on the real needs of the system, saving time and resources.

This review studies the works classified on the third maintenance management strategy commented (PdM). Due to the excellent results that are shown in the industry and its high applicability, in the last years, this area is growing with stunning speed from proposals driven by DM. A fast overview of the historical data about publications related to PdM according to the academic database Scopus supports the increasing interest in this area. Figure 2 shows annual publications referred to PdM in title, abstract, or keywords together to the annual publications referred to PdM and some of

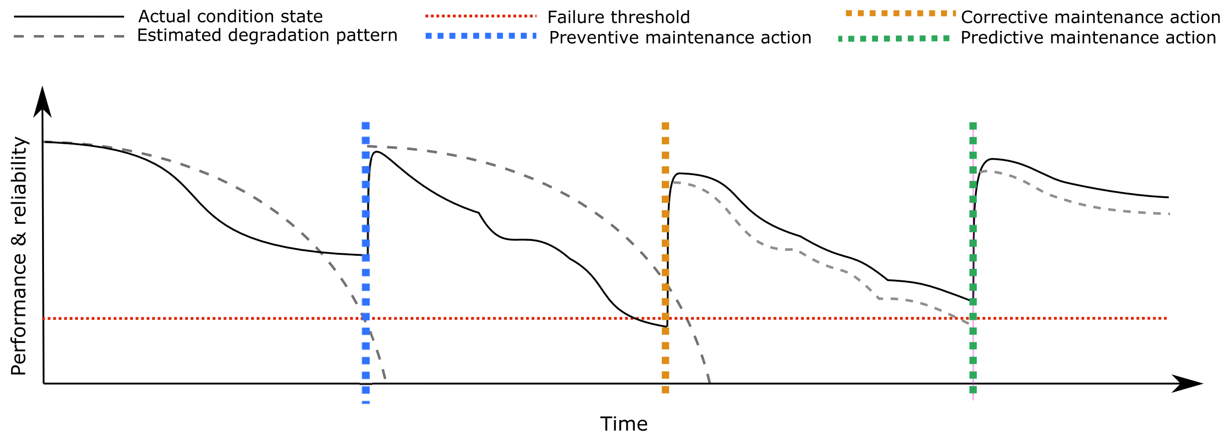

FIGURE 1 The different maintenance strategies

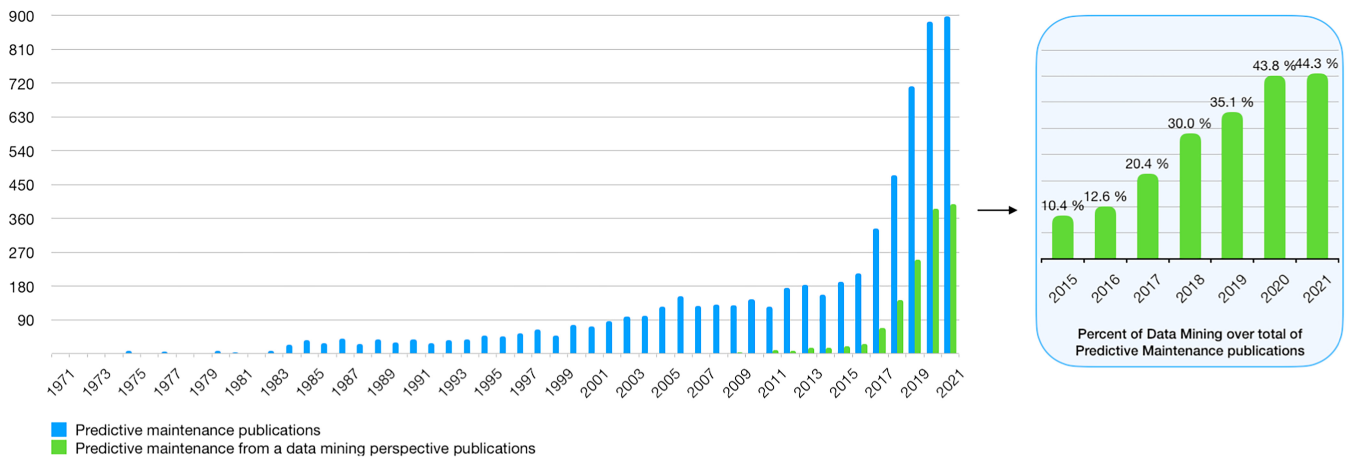

FIGURE 2 Number of academic publications per year in Scopus on February 7, 2022: predictive maintenance (PdM) against PdM related to machine learning

the main terms related to DM field, like *data mining*, *deep learning*, and *machine learning*. The numbering is performed on February 2, 2022, and the search includes the years 1970–2021. This search shows that the first publications in the field date from 1970. During the 2000s the field of study is consolidated with around 100 publications per year. Finally, since 2016 the area grows exponentially, reaching 712 publications in 2019, 884 in 2020 and 900 in 2021. In parallel, first DM approaches to PdM appear in 2002. During the next decade, the number of works per year is less than 10, but since 2016 its growth is also exponential. During last year, DM-based works in PdM represent around a third of the total of publications related to PdM, and around 44% of the total during 2020 and during 2021. As we analyze in the next section, previous PdM reviews cover the period until 2018. However, with this exponential growth and important changes in the application of DM approaches (such DL), it is necessary to carry out a new review that makes an exhaustive revision and taxonomy considering all advances in the period of greatest growth of the field, that is, from 2015 to 2021.

2.2 | Previous reviews in PdM

PdM has emerged as a research area in multidisciplinary research groups integrating lines of research related to data acquisition, infrastructure, storage, distribution, security, and intelligence (Zonta et al., 2020). This has led several authors to perform reviews about the field evolution from different perspectives. This section discusses the different approaches, distinguishing between nonsystematic reviews and systematic ones.

Nonsystematic reviews are useful to understand general tendencies and important contents related to PdM. Thus, in Hashemian and Bean (2011) the different ways of data acquisition through sensors are discussed, as well as its advantages and limitations in the context of PdM. Zhang, Yang, and Wang (2019) make an introduction to data-driven PdM analyzing main acquisition techniques, as well as the processing steps and main ML techniques used to perform the prediction. Other works (Angelopoulos et al., 2020; Wang, 2016) take a more industrial perspective analyzing key aspects of failure detection in the context of Industry 4.0 and the integration of the Internet of Things paradigm in industrial plants. Several works are specialized in the implantation of PdM in specific industrial environment, specifically in manufacturing (Wuest et al., 2016), in oil and gas industry (Hanga & Kovalchuk, 2019), in thermal power plants and pump systems (Olesen & Shaker, 2020), in medium voltage switchgear (Hoffmann et al., 2020) and in highway assets maintenance (Karimzadeh & Shoghli, 2020).

In Table 1, previous existing systematic reviews related to PdM and DM are outlined and compared with the present work. Carvalho et al. (2019) perform a brief review of the main ML techniques applied to PdM between 2009 and 2018, being the most frequent random forest, artificial neural networks, support vector machines, and k-means. There is also collected a list of four public datasets about failure prediction. This work can help to get a high-level view of the first tendencies of data-driven-based PdM. However, the research area has greatly grown, with about 65% of papers published from 2019 to the present day. Thus, new tendencies, such DL, are not included in the proposed taxonomy. Moreover, this study does not propose a specific taxonomy for DM in PdM, it analyzes the proposals directly by the type of algorithm used, often distinguishing between traditional ML, also called shallow learning, and DL. Merkt (2020) reviews a selection of 19 works between 2016 and 2019 analyzing the most frequent ML techniques in CM, PvM, and PdM. It studied the pros and cons of implantation from an industrial perspective. Zonta et al. (2020) review several aspects of PdM focus on the paradigm of Industry 4.0 from a selection of 47 works between 2008 and 2018. Thus, main goals of PdM in an industrial context are identified, as well as models of PdM (RUL, condition-based maintenance, prognostic health management), and finally, it presents a taxonomy of main ML techniques applied in the field. This work covers many aspects of the implantation of PdM, but it is delimited to a specific industrial context. Thus, it is neither specialized in DM techniques nor it presents a taxonomy from a DM perspective. Concretely, the taxonomy used in these works only divide the methods into three categories: (1) *physical model-based methods* focus on the mathematical modeling of the monitored system, (2) *knowledge-based methods* focus on rule systems and expert systems based on prior expert knowledge, and (3) *data-driven methods* cover all proposals based on applying DM on the past behavior of the data to find patterns able to predict future failures of the monitored systems.

In another recent proposal, Çinar et al. (2020) analyze around 120 articles, mainly focused on conference papers, from the last 10 years extracting the ML technique used, the equipment of the prediction, and different aspects related to the extracted data in the context of Industry 4.0. However, the proposal does not draw high-level conclusions from the review, such as an analysis of the main tasks in data-driven PdM, a taxonomy of the phases of the PdM process from a DM perspective, or emerging trends that may guide the coming years.

TABLE 1 Existing systematic reviews related to predictive maintenance (PdM)

Approach	Years	Works	ML-related features	ML-related weaknesses
Carvalho, 2019: ML applied to PdM	2009–October 18, 2018	36	• It shows the most frequent ML methods. • It shows four public datasets.	• It considers to 2018. • It does not distinguish between different data-driven PdM tasks. • It does not propose a taxonomy from different DM phases. • It does not propose a relation between DM tasks, ML paradigms and PdM problems.
Merkt, 2020: ML applied to industrial maintenance	2016–2019	19	• It shows the most frequent ML techniques for CM, PvM and PdM. • It shows a taxonomy relating ML techniques with PdM problems.	• It considers to 2019. • It does not propose a taxonomy from different DM phases. • It does not propose a relation between DM tasks, ML paradigms and PdM problems. • It is very limited the number of ML techniques considered.
Cinar, 2020: ML applied to PdM in manufacturing plants	2010–July 30, 2020	126	• It shows for every analyzed work the algorithm used and a note about the ML task employed, equipment, data acquisition method and data type. • It presents 20 tools for ML programming.	• It considers to 2020. • It does not group the results of the work to extract general tendencies. • It does not propose a taxonomy from different DM phases. • It does not propose a relation between DM tasks, ML paradigms and PdM problems. • The proposed classification of ML methods does not follow the general established taxonomy that relates learning paradigms with algorithms.
This review: DM applied to PdM	2015–2021	129	• It considers to 2021. • It shows a complete taxonomy considering different DM phases: data acquisition and preparation steps together to building model. • It shows a complete taxonomy considering the relation between DM tasks, ML paradigms and PdM problems. • It analyzes latest trends considering most frequent algorithms applied to different DM tasks. • It includes the main future lines to work. • It specifies 15 public datasets. • It specifies 22 frameworks for PdM solutions.	

The review presented in this work is a recent analysis of the current trends in PdM considering the years of more activity in the field, that is, between 2015 and 2021, which involves an analysis of 132 related works. It is specifically focused on the DM perspective, analyzing not only the *model building* phase, but also the most important operations of *data preprocessing* depending on the *data acquisition* method. Thus, it is given a detailed categorization of models based on different *DM* tasks that are related to specific ML paradigms and specific PdM problems rather than on different type of algorithms. In this way, starting from the data nature and the PdM problem to solve, one can review easily all the

DM techniques previously applied in that context. In addition, complementary information to develop PdM techniques from a DM perspective will be provided, like an updated list of public datasets of failure prediction and programming libraries and software tools to deploy the PdM environment. Finally, the latest trends carried out and the significant challenges facing DM are also analyzed.

3 | SYSTEMATIC LITERATURE REVIEW

Given the interest generated in the field of PdM and the amount of data extracted from production processes due to the proliferation of sensing technologies there has been an exponential growth of peer-reviewed articles since the year 2015 approximately. It is important to identify the current tendencies and to be able to distinguish the relevant, reproducible, and high-quality results of the application of the DM to different industry problems for faults detection, health prognosis, and RUL estimation. In this context, the main scientific challenges to address in the current review are:

- To propose a taxonomy for the most updated tasks in PdM, together with the ML paradigms and DM tasks applied to them. DM has emerged as a promising area in PdM applications to prevent failures in equipment that make up the production lines in the industry.
- To review, the main DM steps implied for solving PdM problems: the acquisition of the data from the physical environment of the monitored system, the preprocessing stage for the preparation of the data according to the needs of the model and the building model for solving PdM problems.
- To identify existing tools and libraries to implement PdM tasks.
- To gather a list of public datasets and benchmarks related to PdM in order to facilitate the comparison among proposals.
- To identify the current challenges in PdM from a DM perspective.

Given these scientific challenges and the theme of the review based on PdM from a DM perspective, this systematic literature review follows a procedure based on those proposed in (Carvalho et al., 2019; Zonta et al., 2020): first, the research questions are formulated, then the search strategy is defined, and finally, the article selection is performed.

3.1 | Research questions

With scientific challenges and contributions identified and taking into account the related work, the following research questions are formulated to guide the review:

Q1 What are the main tasks in PdM from a DM perspective?
Q2 What are the ML paradigms and methods applied in the last years in the different PdM tasks?
Q3 What are the most common scenarios to apply PdM?
Q4 What kind of data use to try the validity of the different PdM proposals?
Q5 How are the PdM solutions implemented concerning technical aspects?
Q6 What are the open challenges in PdM?

Q1 and Q2 are formulated to build an updated taxonomy of the main types of PdM techniques and how to address them. Q3 is proposed to define the relevance of PdM in different fields of Industry 4.0, such as smart mobility or the smart maintenance of large-scale supplies like transportation networks or energy grids. Through Q4, it is intended to identify the general tendency of application of PdM proposals between real-world problems and existing datasets, as well as to gather a list of public datasets of the field. Q5 is formulated to identify the most common frameworks, programming languages, and software libraries used to implement the PdM tasks in different scenarios. Q6 brings challenges and future directions.

3.2 | Search strategy

The main purpose of the search is to find the most recent and relevant works that apply a DM approach to solve a PdM task. Thus, the search strategy starts with specific scientific databases to gather an initial collection of publications using a specific search string and applying specific selection criteria. Then, the compilation can increase by adding specifically related works derived from the initial search. A fixed period and a list of quality assessment criteria demarcate the complete search process.

The study is conducted on three well-known scientific databases: the search starts with the Web of Science database, providing an initial set from indexed journals. Then, an equivalent search in Scopus adds more articles from peer-reviewed journals. Finally, Google Scholar is used as a complementary database to consult specifically related works.

The string of search used to do the initial searches has been defined based on the main terminology to refer to PdM applications, as well as the main DM terms identified in the field. Thus, the string must include at least one of the terms *predictive maintenance*, *condition-based monitoring* and *prognostic health management*, and at least one of the DM related terms *data mining*, *machine learning*, and *deep learning*. In this fashion, the final string is the following:

> ("Predictive maintenance" OR "Condition base monitoring" OR "Prognostic Health and Management") AND ("Data mining" OR "Machine learning" OR "Deep learning")

The quality assessment criteria that should guide the search are designed to make accessible a field which gathers hundreds of proposals in recent years. These are:

- To remove all publications that do not present any type of specific proposal related to PdM solved from a DM approach with associated experimentation.
- To limit to international publications written in the English language.
- To exclude books, technical reports, dissertation, and thesis.
- To exclude conference proceedings from the initial search, but the complementary search can add specific works if they present a sufficiently developed and novel work.

Due to the high volume of works related to PdM published in the last years, the purpose of this review is to focus on the most recent ones. Thus, the search has been restricted to years between 2015 and 2021, covering a more significant period than previous reviews. As has been commented in related work, Carvalho et al. (2019) present a general systematic survey of DM in PdM between years 2009 and 2018. Likewise, a systematic survey in monitoring in Industry 4.0 between the years 2008 and 2018 can be consulted in Zonta et al. (2020), although it is more focused on defining activities, goals, or methods of data acquisition than methods based on ML for building models.

3.3 | Article selection

As discussed above, there is a high volume of works in the field. Thus the general approach to the search is to start with a relatively limited number of quality works and then expanding the selection with articles related to the initial set. Thus, the flow of the search has gone from the most restricted database to the most open, as Figure 3 shows. The search was performed on February, 2, 2022, starting with the scientific database Web of Science. Here, the search string, defined in the previous section, is applied with a "Theme search," that is, in the title, abstract, and keywords of the works. The initial query returns 643 articles between the years 2015 and 2021. Then, the quality assessment criteria C3 and C4 are applied to restrict the search to journal articles. Next, 16 works in noninternational journals are removed (C2 criterion). Finally, a comprehensive lecture of the remaining 358 works is performed following the C1 criterion to keep 93 works from this database.

After the cataloging performed in Web of Science, the search string has been applied in the Scopus database to complement the initial set of articles. In this case, the search is performed in the fields *Article title*, *Abstract*, and *Keywords*. The initial query returns 1020 works between 2015 and 2021. Application of the C3 and C4 criteria reduced the selection to 324 works. The next step is removing the duplicated works already processed through the Web of Science database. The C1 criterion is applied to the remaining 62 works to finally select 32 articles.

Attending to Google Scholar database, with an initial set of articles of 80 related articles about PdM addressed from a DM perspective, relevant proposals have been selected attending to the C1 criterion. This search has resulted in seven additional works published between 2015 and 2021. Thus, the final corpus of this study is composed of 132 articles.

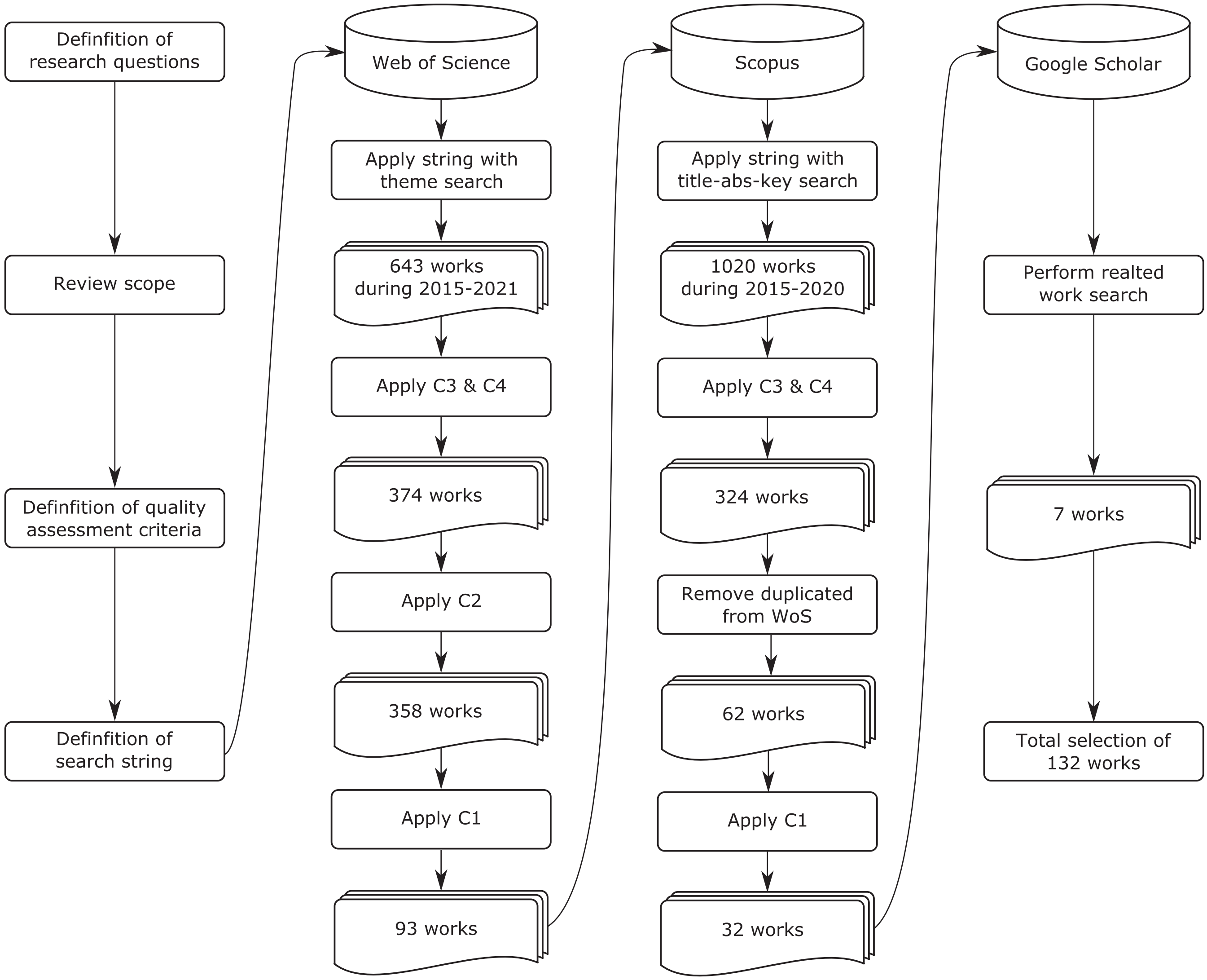

FIGURE 3 Flow of the search performed on February 7, 2022

4 | DATA MINING IN PREDICTIVE MAINTENANCE

This section gathers the main results of the review conducted for DM-based PdM solutions in recent years. Thus, it is provided an up-to-date systematic review on PdM from DM perspective. To effectively carry out DM in PdM processes, ML has played an important role in the step of model building. Although ML is a broad and constantly evolving field, one can consider that there are four main learning paradigms (Burkov, 2019): supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Reinforcement learning is widely employed in other fields in the process industry, while the other three ML paradigms have been extensively used for DM in PdM.

4.1 | Taxonomy

The proposed taxonomy is based on the DM methodology applied to PdM, including descriptions and discussions in each of the main procedures of the knowledge discovery process that are implied: data acquisition, data preprocessing, and learning model building. The proposed taxonomy includes a review of the different learning frameworks considering supervised learning, unsupervised learning, and semi-supervised learning methods. Under each type of learning, DM tasks are discussed, such as classification, regression, or clustering. For each task, various ML algorithms are specified with detailed descriptions and their main applications.

Figure 4 presents the proposed taxonomy of this work, which covers the general steps considered in the process of application of DM to PdM. Thus, this study focuses on three parts:

1. *Data acquisition*: The first step in PdM concerns the industrial domain, and more specifically, how to monitor the system and how to extract data from it. This step conditions the whole PdM process because the nature of the data and the possibilities of monitoring determine what PdM problem to address and what ML paradigm to use for it.
2. *Data preprocessing*: The second step in PdM addresses the data preparation to improve the quality of the predictions, that is, some appropriate data transformations may be needed to make the data modeling more efficient in the next step. Several approaches can be followed: addressing possible missing values, summarizing data amount by extracting statistics or applying projections, or selecting the most relevant features. However, the study of the current state of the art has shown that this is an optional step, especially if DL is employed in the next phase. This is due to DL paradigm can be applied over the raw data.
3. *Model building*: This is the core step in DM applied to PdM, where data are analyzed to detect possible anomalies, failures, or RUL in advance. Depending on the task, the type of the data, and how they are labeled, different types of learning can be applied (supervised, semi-supervised, or unsupervised learning). Moreover, specific problem characteristics, such as the response time, the computational capability, or the explainability of the prediction influence the learning model to choose.

In following sections, the most recent state of the art in PdM concerning to each category is analyzed in terms of data acquisition methods, the preprocessing operations used and model building.

4.2 | Data acquisition

This section discusses the main methods for data obtaining from the monitored systems, which will also depend on the type of data being worked with, as listed in Table 2. Four main methods for data acquisition have been identified and they are explained in the following sections: sensor connection, external capture, inspection logs, and simulation of the data. Besides, it has been detected several data categories: physical types categories like vibration, temperature, and pressure; electrical signals related to the inner workings of the machinery; image analysis; oil analysis; geolocation

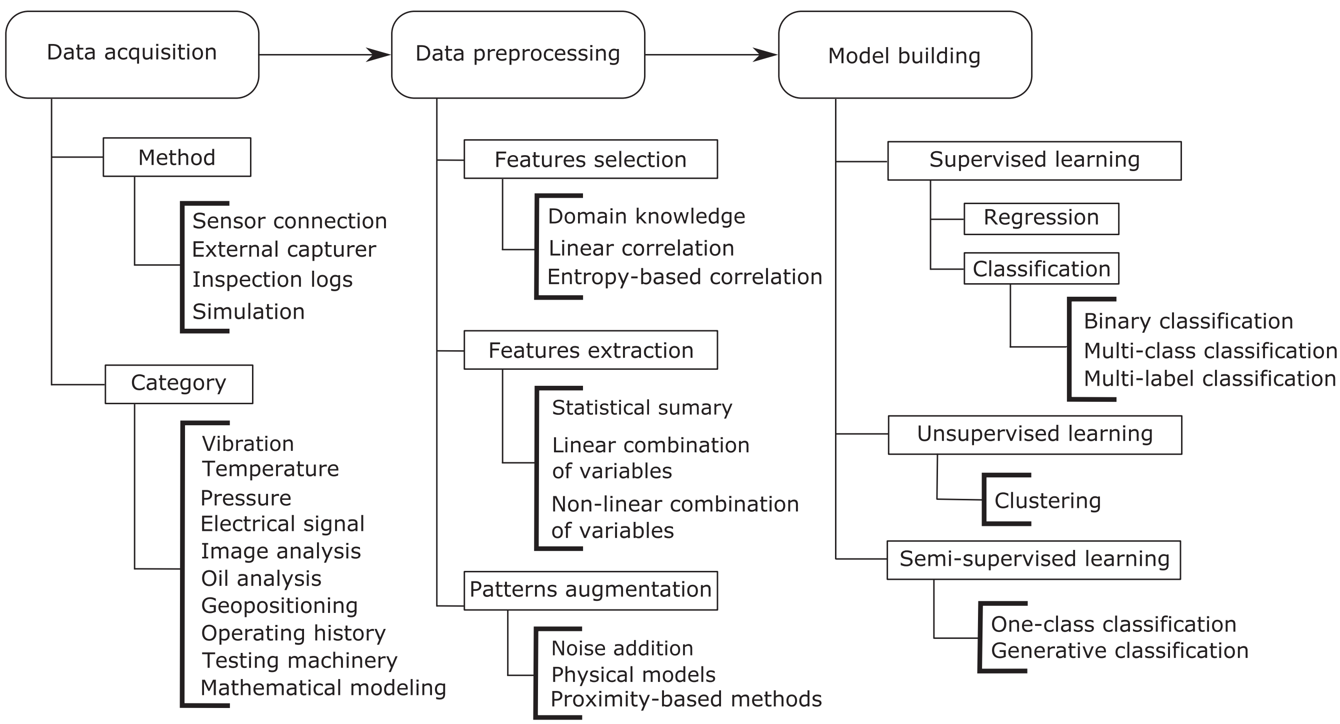

FIGURE 4 Taxonomy of main predictive maintenance steps from a data mining perspective

TABLE 2 Data acquisition in PdM

Method	Category	References
Sensor connection	Vibration frequency	Chen et al. (2019), Cho et al. (2020), Liang et al. (2020), Oluwasegun and Jung (2020), Wang, Liu, et al. (2020), Zhou et al. (2019), and Bouabdallaoui et al. (2021)
	Acceleration and relative position	Shamayleh et al. (2020), Casoli et al. (2019), Mishra and Huhtala (2019), Malawade et al. (2021), Yang et al. (2021), and Aqueveque et al. (2021)
	Temperature, pressure, and vibration	Axenie et al. (2020), Orrù et al. (2020), Quatrini et al. (2020), Hsu et al. (2020), Yu et al. (2020), Cheng et al. (2020), Acernese et al. (2020), Khorsheed and Beyca (2021), and Steurtewagen and Van den Poel (2021)
	Electrical signal—Inner operation in general	Uhlmann et al. (2018), Srivastava et al. (2018), Rafique et al. (2018), Leahy et al. (2018), Fernandes et al. (2020), Crespo Márquez et al. (2020), Pałasz and Przysowa (2019), Zschech et al. (2019), Kang et al. (2021), Bampoula et al. (2021), Kim et al. (2021), Fathi et al. (2021), and Santolamazza et al. (2021)
	Electrical signal—Manufacturing	Susto et al. (2015), Li et al. (2017), Morariu et al. (2020), Kolokas et al. (2020), Ruiz-Sarmiento et al. (2020), Bekar et al. (2020), Lepenioti et al. (2020), Ayvaz and Alpay (2021), Cerquitelli et al. (2021), Azab et al. (2021), Serradilla et al. (2021), Liu et al. (2021), Chang et al. (2021), Giordano et al. (2021)
	Electrical signal—Power plants	Khodabakhsh et al. (2018), Zhang, Liu, et al. (2018), de Carvalho Chrysostomo et al. (2020), Sun et al. (2021)
	Electrical signal—Transportation	Prytz et al. (2015), Ribeiro et al. (2016), Yang et al. (2017), Shafi et al. (2018), Chen et al. (2020), Savitha et al. (2020), Fernández-Barrero et al. (2021), Basora et al. (2021), Gribbestad et al. (2021), Ning et al. (2021), Berghout et al. (2021), Patil et al. (2021), Baptista et al. (2021)
External capture	Image analysis	Ullah et al. (2017), Lasisi and Attoh-Okine (2018), Oliveira et al. (2020), Consilvio et al. (2020), Schlagenhauf and Burghardt (2021)
	Oil concentrations	Keartland and Van Zyl (2020)
	Geopositioning	Proto et al. (2020)
Inspection logs	Maintenance log	Cheng et al. (2020), Consilvio et al. (2020), Zschech et al. (2019), Bampoula et al. (2021), Chen, Liu, et al. (2021), Usuga-Cadavid et al. (2021), Steurtewagen and Van den Poel (2021)
	Position registration	Proto et al. (2020), Chen, Liu, et al. (2021)
	Log standard output	Koca et al. (2020), Calabrese et al. (2020), Pezze et al. (2021), Dangut et al. (2021)
Simulation	Testing machinery	Liao et al. (2016), Satishkumar and Sugumaran (2017), Una et al. (2017), Cupek et al. (2018), D. Wang et al. (2018), Wang, Zhang, et al. (2020), Liang et al. (2020), Zhang, Li, Wang, et al. (2019), Yang, Lei, et al. (2019), Qian et al. (2019), K. Li et al. (2019), Pang et al. (2020), Zenisek, Holzinger, and Affenzeller (2019), Hu et al. (2020), Panicucci et al. (2020), Sampaio et al. (2019), Song et al. (2021), Cakir et al. (2021), Xu et al. (2021)
	Mathematical models	Zenisek, Holzinger, and Affenzeller (2019), Zenisek, Kronberger, et al. (2019), Venkataswamy et al. (2020)

values; different types of operating history; and simulation-based on physical machinery or mathematical modeling. The works using public datasets deserve a separate mention, which is discussed in Section 5.1.

4.2.1 | Sensor connection

Most PdM works acquire their data directly from a real environment by means of sensors that capture different working properties in the monitored system. Sensors are very useful devices because of their ease of installation, their versatility in capturing different types of data, and their small size and connectivity capabilities. All these aspects make sensors widely used devices that allow connectivity by minimally altering an existing industrial environment, thus obtaining the monitoring required to apply PdM. Different kind of sensors are reported in last years in PdM for measuring vibration frequency (Bouabdallaoui et al., 2021; Chen et al., 2019; Cho et al., 2020; Liang et al., 2020; Oluwasegun & Jung, 2020; Orrù et al., 2020; Wang, Liu, et al., 2020; Zhou et al., 2019) or vibration acceleration alone (Aqueveque et al., 2021; Casoli et al., 2019; Malawade et al., 2021; Shamayleh et al., 2020; Yang et al., 2021) or in conjunction with relative position (Mishra & Huhtala, 2019). Other works use sensors that capture atmosphere-based features like temperature (Axenie et al., 2020). But most of the works that use this type of data acquired from sensors combine several information like temperature, pressure and vibration, among others (Acernese et al., 2020; Cheng et al., 2020; Hsu et al., 2020; Khorsheed & Beyca, 2021; Orrù et al., 2020; Quatrini et al., 2020; Steurtewagen & Van den Poel, 2021; Yu et al., 2020). Another approach is the use of internal operating logs through sensors already present in complex systems, which monitor various aspects of the processes taking place in the monitored systems. These records are extracted later or in real time to apply PdM. Examples of this case are found on the one hand in industrial processes (Bampoula et al., 2021; Bekar et al., 2020; Chang et al., 2021; Kim et al., 2021; Kolokas et al., 2020; Lepenioti et al., 2020; Li et al., 2017; Morariu et al., 2020; Rafique et al., 2018; Ruiz-Sarmiento et al., 2020; Susto et al., 2015; Uhlmann et al., 2018; Zschech et al., 2019), in production lines (Ayvaz & Alpay, 2021; Azab et al., 2021; Cerquitelli et al., 2021; Fathi et al., 2021; Giordano et al., 2021; Liu et al., 2021), in power plants (de Carvalho Chrysostomo et al., 2020; Khodabakhsh et al., 2018; Sun et al., 2021; Zhang, Liu, et al., 2018), in wind turbines (Chen, Hsu, et al., 2021; Leahy et al., 2018; Santolamazza et al., 2021), in ventilation systems (Fernandes et al., 2020), cryogenic pumps (Crespo Márquez et al., 2020), heat meters (Pałasz & Przysowa, 2019), press machines (Serradilla et al., 2021), or water treatment plants (Srivastava et al., 2018). Another common scenario for PdM is related to transportation: different types of land vehicles (Chen et al., 2020; Patil et al., 2021; Prytz et al., 2015; Shafi et al., 2018), aircrafts (Baptista et al., 2021; Basora et al., 2021; Ning et al., 2021; Savitha et al., 2020; Yang et al., 2017), and naval ships (Berghout et al., 2021; Fernández-Barrero et al., 2021; Gribbestad et al., 2021) have been monitored through their electronic control units. Also, Ribeiro et al. (2016) found a case that monitors train doors status.

4.2.2 | External capture

In PdM applied to real industrial cases, sensors are not always used to capture operating data, either because of the cost of installation or because they are too invasive for the monitored system. Thus, in Oliveira et al. (2020) a pulse-echo ultrasonic technique based on local immersion was used to acquire data from wind turbine blade test specimens. Schlagenhauf and Burghardt (2021) proposed an image-based analysis from the photographs of the degradation of the analyzed surface of a ball screw drive. In Lasisi and Attoh-Okine (2018) and Consilvio et al. (2020), the track geometry measures of railway assets are used, specifically in Consilvio et al. (2020), as a complementary source of data together with inspection logs. Other example of images analysis is found in Ullah et al. (2017) that takes infrared thermal images of power substations to detect temperature anomalies. In Keartland and Van Zyl (2020), the condition of gearbox compartments is monitored through the analysis of their oil and the different concentration levels in it. Proto et al. (2020) used a NFC approach to perform geopositioning of shipped packages in a parcel delivery service.

4.2.3 | Inspection logs

This section groups the works that use for predictive maintenance data initially generated in order to serve as a failure log for human monitoring. Thus, some works use these records, either in a complementary way or as the only source of data to make predictions. One approach is shown in Cheng et al. (2020), Consilvio et al. (2020), Zschech et al. (2019), Bampoula et al. (2021), Steurtewagen and Van den Poel (2021), where inspection and historical maintenance records and manual failure registers are used together with other kind of data depending on the domain and generally obtained through sensors. In the context of parcel delivery services, Proto et al. (2020) follow a similar approach that combines manual registers of packages position with another data automatically collected. In the case of Usuga-Cadavid et al. (2021), maintenance logs provided by operators are used as the unique source for the PdM. As they are written by humans, the unstructured, nosily text is treated with natural language processing in successive stages to perform the PdM task. A similar approach, but applied to vehicles, is followed in Chen, Liu, et al. (2021): instead of using sensor connectivity, it combines both maintenance records and weather, traffic and terrain factors, captured by the geographical information system of the vehicles fleet. In some works (Calabrese et al., 2020; Dangut et al., 2021; Koca et al., 2020; Pezze et al., 2021), the use of alarms or warnings designed for human interaction is proposed as a cheaper alternative to the implantation of sensors. These human-readable logs must be properly parsed to obtain numerical data that can be used in the subsequent phases of DM-based PdM.

4.2.4 | Simulation

In many PdM cases, the data are obtained under simulated or controlled conditions in a laboratory. In this way, the aim is to simulate behaviors that could occur in a real system to which there is not access or which has not yet failures that can be used to train the models. In this line, there are many works that collect vibration signals from different experimental rotors (Cakir et al., 2021; Li et al., 2019; Liang et al., 2020; Liao et al., 2016; Pang et al., 2020; Qian et al., 2019; Satishkumar & Sugumaran, 2017; Song et al., 2021; Una et al., 2017; Wang et al., 2018; Wang, Zhang, et al., 2020; Yang, Lei, et al., 2019; Zhang, Li, Wang, et al., 2019), fans (Sampaio et al., 2019; Xu et al., 2021; Zenisek, Holzinger, & Affenzeller, 2019), centrifugal pumps (Hu et al., 2020), air compressors (Cupek et al., 2018) or industrial robots (Panicucci et al., 2020). Some works (Venkataswamy et al., 2020; Zenisek, Holzinger, & Affenzeller, 2019; Zenisek, Kronberger, et al., 2019) generate synthetic data from a mathematical approach that models the behavior of their problem, using for that specialized software in simulation.

4.3 | Data preprocessing

This section groups different data preparation strategies observed in PdM in recent years. These range from manual or automatic feature selection, to the processing of sparse data, or the generation of patterns to improve the detection of infrequent events. This section is approached as a preliminary step to the predictive methods that will be addressed in Section 4.4, although the strategies described here are neither mandatory nor mutually exclusive. Thus, for example, with the rise of DL, the direct use of raw data without prior processing is becoming more and more frequent. Likewise, it is common to use both feature selection and transformation in many works.

The different categories found in the reviewed work were presented in the taxonomy commented in Figure 4, and they are addressed in the following sections. Table 3 presents a summary of the analyzed works that carry out the preprocessing step following the division proposed in our taxonomy: feature selection, feature extraction, and pattern augmentation, as well as the type of data that makes it possible to apply the different techniques listed below. The last column contains the references of the papers that employ each technique. The sections below explain in detail the review corresponding to data preprocessing.

4.3.1 | Feature selection

Often the systems monitored in PdM have a large number of sensors that are not necessarily useful for a particular PdM problem. Their presence in the prediction model adds noise and slows down the learning process. Feature selection is a data preprocessing technique that reduces the number of inputs that are used by algorithm, that is, the number of attributes or characteristics of the patterns, by selecting those most relevant to the desired target in the prediction. There are two main approaches: manual search, which in PdM means expert knowledge of the monitored systems, or automatic search based on relationships between data. In the latter case, the methods reviewed may belong to the linear correlation analysis or the entropy-based method.

Domain knowledge

Feature selection based on domain knowledge refers to use the previous experience of system users to select the most relevant variables according to their criteria. Thus, this method is applied when the studied case belongs to a real environment (Bampoula et al., 2021; Chen et al., 2020; Chen, Hsu, et al., 2021; Cheng et al., 2020; Consilvio et al., 2020; Lepenioti et al., 2020; Patil et al., 2021; Prytz et al., 2015; Ruiz-Sarmiento et al., 2020; Yang et al., 2017; Zenisek, Holzinger, & Affenzeller, 2019; Zschech et al., 2019) where there is a wide experience in terms of human knowledge of how the system behaves. This technique is also used by resorting to previous literature as shown in Leahy et al. (2018), where previous works on the same subject, wind turbines, are used to select the most relevant variables. This type of

TABLE 3 Data preprocessing in PdM

Problem	Type of data	Technique based on	References
Feature selection	Multiple variables not related with the PdM problem to solve.	Domain knowledge	Prytz et al. (2015), C. Yang et al. (2017), Leahy et al. (2018), Zenisek, Holzinger, and Affenzeller (2019), Zhang, Zhang, and Li (2019), Zschech et al. (2019), C. Chen et al. (2020), Lepenioti et al. (2020), Consilvio et al. (2020), Ruiz-Sarmiento et al. (2020), Cheng et al. (2020), Chen, Hsu, et al. (2021), Bampoula et al. (2021), Patil et al. (2021)
		Linear correlation	Quatrini et al. (2020), Hsu et al. (2020), Shamayleh et al. (2020), Panicucci et al. (2020), Kolokas et al. (2020), Pałasz and Przysowa (2019)
		Entropy-based correlation	Aremu, Cody, et al. (2020), K. B. Zhou et al. (2019), Prytz et al. (2015)
Feature extraction	High number of variables related with the PdM problem to solve.	Statistical summary	Susto et al. (2015), Ullah et al. (2017), Uhlmann et al. (2018), Ruiz-Sarmiento et al. (2020), Orrù et al. (2020), Quatrini et al. (2020), Panicucci et al. (2020), Kolokas et al. (2020), Zenisek, Holzinger, and Affenzeller (2019), Morariu et al. (2020), Yu et al. (2020), Proto et al. (2020), Calabrese et al. (2020), Zschech et al. (2019), Cerquitelli et al. (2021), Kamat et al. (2021), Baptista et al. (2021))
		Linear combinations	Zhang, Liu, et al. (2018), Lasisi and Attoh-Okine (2018), Shafi et al. (2018), Yu et al. (2020), Bekar et al. (2020), Shamayleh et al. (2020), Casoli et al. (2019), Oliveira et al. (2020), Oluwasegun and Jung (2020), Axenie et al. (2020), Quatrini et al. (2020), Ruiz-Sarmiento et al. (2020), Kang et al. (2021), Chen, Hsu, et al. (2021), Ayvaz and Alpay (2021), Khorsheed and Beyca (2021)
		Nonlinear combinations	Ullah et al. (2017), Una et al. (2017), Aremu, Hyland-Wood, and McAree (2020), Casoli et al. (2019), Shamayleh et al. (2020), Oliveira et al. (2020), Z. Chen et al. (2019), K. B. Zhou et al. (2019), Hu et al. (2020), Sampaio et al. (2019), Mishra and Huhtala (2019), Zschech et al. (2019), Chen, Zhu, et al. (2021), Pillai and Vadakkepat (2021), Song et al. (2021), Aqueveque et al. (2021), Xu et al. (2021), Chui et al. (2021)
Pattern augmentation	Few records of data failures.	Noise addition	Qian et al. (2019)
		Physical models	Wang, Bu, and He (2020)
		Proximity-based methods	Oh and Lee (2020), Chen, Hsu, et al. (2021), Aqueveque et al. (2021), Dangut et al. (2021)

feature selection can be performed in a more theoretical environment (Zhang, Zhang, & Li, 2019), working with a public dataset with multiple characteristics and it is desired to use a technique only applicable to one of the data types.

Linear correlation

The correlation analysis is a statistical method that measures the association on two numerical variables. The most common method to find correlations is the Pearson correlation coefficient, which focuses only on linear relationships. This is a common method for feature selection in PdM, selecting only those variables that are most strongly related to the target variable (Hsu et al., 2020; Pałasz & Przysowa, 2019; Quatrini et al., 2020), or reducing the number of input variables used for prediction by avoiding those that are redundant because they are strongly correlated with others (Panicucci et al., 2020; Shamayleh et al., 2020). Another feature selection method called Mathew Correlation Coefficient is used in Kolokas et al. (2020). This correlation method takes into account the correlation between the observed and the predicted class and it is used in Kolokas et al. (2020) as a feature selection method, as well as for the horizon prediction adjustment.

Entropy-based correlation

Similar to linear correlation, some studies explore other approaches different to linear one to find correlations between variables and to eliminate those that do not provide new information. Thus, in Aremu, Cody, et al. (2020), features are grouped by hierarchical clustering and then, within each group a relative entropy operation based on logistic distribution is performed, in such a way that only features with a percentage change exceeding a given threshold are selected. The study of Zhou et al. (2019) used Gram–Schmidt orthogonal process which sorts the correlation between input features and output target in descending order by a criterion. Then, the N more relevant are selected to perform the prediction. In Prytz et al. (2015), two methods based on statistical differences are studied to select the most relevant features.

4.3.2 | Feature extraction

PdM systems require a fast response in the processing of incoming data to enable real-time monitoring of the operating status. As commented in the previous section, sensor-based system monitoring has several input variables. In this approach, unlike in feature selection, all features are assumed to be relevant to the PdM problem. Thus, it is sought to summarize them for faster and more efficient processing. Feature extraction methods seek to combine several real-world features by mapping them to a lower dimensionality space to speed up the response of ML methods fed with these data. This section describes the most current methods used for feature extraction in PdM.

Statistical summary

This category groups strategies that include increasing the available information by calculating classical statistics such as maximum and minimum values, mean or standard deviation on the input data used for the predictive models. In Ullah et al. (2017), Uhlmann et al. (2018), and Ruiz-Sarmiento et al. (2020), the statistics are combined with the original features in order to provide additional information. In many PdM works, due to the temporal relationship of the data, it is common to perform statistics operations by working cycles (Baptista et al., 2021; Cerquitelli et al., 2021; Kolokas et al., 2020; Orrù et al., 2020; Panicucci et al., 2020; Quatrini et al., 2020; Susto et al., 2015), in such a way that new features are computed at each time interval. Another approach is found in Zenisek, Holzinger, and Affenzeller (2019), where time series are preprocessed by generating time lagged values which are referred to extract new features that historically saw the last n values in a series to a new input vector. Some approaches (Calabrese et al., 2020; Morariu et al., 2020; Proto et al., 2020; Yu et al., 2020) work with event-related data that are summarized to obtain higher level features by aggregation specific events to the problem domain. Most of the works (Calabrese et al., 2020; Kamat et al., 2021;Morariu et al., 2020; Yu et al., 2020) summarize the characteristics from a time series perspective, summing or obtaining other statistics such as the mean at a higher sampling frequency, but preserving the temporal relationship of the data. Others (Proto et al., 2020) transform the problem from time-series domain to a discrete attributes one by obtaining statistics of all the records of a given event. In some cases (Morariu et al., 2020; Yu et al., 2020), the data streams are processed in a distributed way using Map Reduce.

Linear combination of variables

The linear combination of two or more variables allows to preserve the operations of vector addition and scalar multiplication reducing the dimensions needed to represent these variables. Principal Component Analysis (PCA) is the most popular process for finding the optimal combinations. This process computes the sequence of direction vectors that best fits with the data set, and projects each data point on the first few principal components to obtain lower-dimensional data. Thus, the data variation is preserved. PCA has been used in several PdM works (Ayvaz & Alpay, 2021; Bekar et al., 2020; Chen, Hsu, et al., 2021; Kang et al., 2021; Lasisi & Attoh-Okine, 2018; Shafi et al., 2018; Yu et al., 2020) to reduce the number of variables produced by sensors measurements. In the context of vibration signal analysis, PCA has been applied to reduce the feature vector space in frequency domain (Casoli et al., 2019; Khorsheed & Beyca, 2021; Oliveira et al., 2020; Shamayleh et al., 2020), or in combination with partitional clustering to label the data according to clusters and then applying supervised learning (Oluwasegun & Jung, 2020). In Axenie et al. (2020), several accumulate-retract learning methods are used to accelerate PCA process achieving low-latency high-throughput processing of data streams in PdM. In Zhang, Liu, et al. (2018), PCA is used in combination with a curve-registration method of correlation maximization algorithm to address the problem of time-lagged correlation for multiple sensors. Linear Discriminant Analysis (LDA) is another related method of linear combination that takes into account the class label of the data that is also used as a preprocessing step in PdM (Quatrini et al., 2020). Another approach based on the coefficient of determination is used in Ruiz-Sarmiento et al. (2020) to study how well the target degradation state is predicted by each one of the candidate features.

Nonlinear combination of variables

Some PdM works explore more sophisticated method of dimensionality reduction using nonlinear methods for combining features. The study of Aremu, Hyland-Wood, and McAree (2020) proposed a framework for high dimensional data with discontinuity across the time. Thus, the framework learns a clustering of observations based on the modality of the dataset, defined by a kernel density estimation. Next, low-dimensional representations of each cluster are learned using Laplacian eigenmaps. Finally, the original temporal sequence of observations in the low-dimensional clusters is used to reindex the observations into a low-dimensional continuous feature set. When signals come from some kind of vibration activities like wind turbines or motors, different mathematical transformations have been applied to extract frequency domain features from the signal in time domain. The most commonly used methods belong to the Fourier transform family which is applied to decompose the signal into components of different frequencies. Specifically, several works use the Fast Fourier Transformation (FFT) in order to reduce the feature space (Aqueveque et al., 2021; Casoli et al., 2019; Sampaio et al., 2019; Shamayleh et al., 2020; Zschech et al., 2019). The study of Song et al. (2021) employed the Short-time Fourier Transform, which focuses on determining the sinusoidal frequency and phase content of local sections of the signal as it changes over time. Another analysis uses the Wavelet Transform which is based on applying operations that affect the time, but not the shape, so that the transformed signal contains both frequency and time information. In Oliveira et al. (2020), Discrete Wavelet Transform is applied for noise reduction and then, Discrete Fourier Transform is used to access frequency-domain information. In Chen et al. (2019), the Continuous Wavelet Transform is applied over time signals in order to convert them into time-frequency images. In Zhou et al. (2019), Empirical Wavelet Transform is used to extract several modes from vibration signal. The study of Chui et al. (2021) proposed a combination of the Complete Ensemble Empirical Mode Decomposition and Wavelet Packet Transform in a two-step decomposition able to capture both time and frequency information in signal analysis. Another proposal (Xu et al., 2021) that works with vibration data compares the performance of the FFT, Higher Order Statistics and Structural Co-occurrence Matrix. In Una et al. (2017), a processing closely related to vibration signals, but applied to acoustic signals, is carried out. Thus, feature extraction from the original sound signal is done by enveloping analysis with Hilbert transform and frequency components extraction with FFT and Power Spectral Density. The study of Ullah et al. (2017) used feature extraction over images, specifically based on gray-level co-occurrence matrix to obtain contrast, correlation, homogeneity, and energy of each sample. In Hu et al. (2020), vibration signals are transformed to images and feature extraction is specific to this type of data. Thus, two different nonlinear approaches are compared. On the one hand, it works with texture features analyzing group of pixels in order to explain the visual patterns with homogeneity property. On the other hand, authors explore different DL methods, specifically based on Convolutional Neural Network (CNN), as feature extractor. The features extracted from any of the approaches are them passed to the ML models to perform the PdM task. However, the most common DL method for feature extraction is the Autoencoder (AE), in which the data are compressed or reduced to lower dimensions and then decoded to the desired dimension. Examples of AE in the context of feature extraction are found in Mishra and Huhtala (2019) and Chen, Zhu, et al. (2021), extracting in all cases operating profiles from time-series. Pillai and Vadakkepat (2021) utilize a multilayer convolutional AE based on a multiloss objective function. The relevant information maximizing encoder generates high-dimensional representation that is processed in a second stage through a depth-wise separable convolution that learns temporal features. The temporal features and encoded representation are used in the second phase of ML.

4.3.3 | Pattern augmentation

Artificial data generation has become very popular in recent years because methods such as DL require a high number of data to fit large amounts of parameters. PdM, in addition, is often based on fault learning that occur infrequently, that is, little data are available to learn critical operating moments. In this context, some works choose to use the few patterns that represent a fault to generate more patterns from modifications of their key features. One example is found in Qian et al. (2019). It generates new data in the context of signal analysis by selecting samples from the existing signals and adding Gaussian noise. However, the most common approach is based on methods of distance. Thus, in Chen, Hsu, et al. (2021) and Aqueveque et al. (2021), the imbalanced data is addressed using SMOTE selecting samples closed in the feature space. Specifically, a random example from the minority class is first chosen. Then, k of the nearest neighbors are found. A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space. Another method based on distance for textual log data is proposed in Dangut et al. (2021) as a solution for imbalanced classification: the model is based on a log-based pattern identification technique, which involves transforming and integrating well-known natural language processing techniques (TF–IDF and Word2vec). The model finds all patterns, that is, succession of failure messages related to given target, and from them, it creates all possible combinations of patterns increasing the training dataset for the given target. Other example is presented in Oh and Lee (2020). First, it estimates missing values using Gaussian Process Regression (GPR), and then, it uses DL, specifically a Generative Adversarial Network (GAN), to perform data augmentation. A domain-specific approach is followed in Wang, Bu, and He (2020). The aim is to generate synthetic data using a physical model that describes the statistical behavior of the random failure time in the problem using theory of probability and a stochastic process.

4.4 | Model building

PdM problems addressed from a DM perspective using ML algorithms can follow three main approaches depending on the type of learning used in the prediction: supervised, unsupervised, and semi-supervised learning.

Supervised learning deals with labeled data samples that give information about the system's performance. These labels can be either a continuous signal or a discrete set of categories. Tables 4 and 5 show the different supervised ML problems depending on the type of information available, and the PdM task it addresses.

The main approach in supervised learning uses discrete labels, addressing the problem as a classification task. This task is very much related to the PdM problem of health status estimation, where data used to address the PdM problem are labeled according to several categories related to one or more failures. Depending on the number of available labels, it can be a binary or multiclass classification. In addition, some works address the possibility of analyzing multiple labels simultaneously, using a multilabel learning perspective. The findings related to classification in PdM in recent years are described in Section 4.4.1. On the other hand, if the output label is continuous, the PdM problem is addressed as a regression task. This task is mainly related to the estimation of the future evolution of a continuous variable, such as a signal of interest for the system or a health index. This estimation is used to carry out a prediction of RUL of the monitored system in the time domain: remaining hours or working cycles, among others. The other great PdM task in regression field is the prediction of the behavior of the monitoring system, normally one or several continuous signals critical for the working condition. This prediction of signal evolution can serve as input to expert systems or users to trigger alarms related to patterns observed in the future behavior of the system. Section 4.4.2 describes the details of the proposals that address regression problems.

The unsupervised learning paradigm is applied when no labels are associated with the data. Thus, there is no information about what data are related to the correct operation of the system and which is not. This is an area of growing interest in the field of PdM because analyses are often based on data acquired directly from sensors used with minimal preprocessing. Thus, the main ML paradigm to address these data is the clustering task that search different ways of grouping the data. Clustering is also very much related with the task of outlier detection, which focuses on detecting those patterns in the distribution that cannot be grouped with the majority. Thus, unsupervised learning is used in PdM problems that work with data obtained during periods of different unidentified operating regimes, including malfunctions, that is, data have not associated labels.

Finally, semi-supervised learning is a hybrid approach that works with labeled and unlabeled data. In the context of PdM, this scenario has been mainly addressed from the ML paradigm of one-class classification working with data labeled only for correct operations, but some more recent articles also approach it from the point of view of the generative classification. In one-class classification, the predictive model is built from an initial dataset with common characteristics. At evaluation time, the model determines whether new incoming data fits this initial set or not. PdM problems use this task when there are not still fault records in the monitored system. Thus, the predictive model makes a region of normality from the current operating data. From there, it tries to discern whether the new data fits this region or their characteristics are sufficiently different to indicate a system malfunction. In generative classification, there is a small proportion of labeled data and a great amount of unlabeled data that it is assumed to bellow to the previous

TABLE 4 PdM problems from a supervised perspective I (classification)

DM task	Type of data	PdM problem	ML technique	References
Binary classification	Data labeled in two categories: correct or malfunction	Generic failure prediction	K-Nearest Neighbors	Susto et al. (2015), Shafi et al. (2018), Baptista et al. (2021), Cakir et al. (2021)
			Logistic Regression	Proto et al. (2020), Gohel et al. (2020)
			Decision Trees	Prytz et al. (2015), C. Yang et al. (2017), Su and Huang (2018), Lasisi and Attoh-Okine (2018), Shafi et al. (2018), Mishra and Huhtala (2019), Pałasz and Przysowa (2019), de Carvalho Chrysostomo et al. (2020), Aremu, Hyland-Wood, and McAree (2020), Proto et al. (2020), Kaparthi and Bumblauskas (2020), Hsu et al. (2020), Calabrese et al. (2020), Khorsheed and Beyca (2021), Steurtewagen and Van den Poel (2021), Patil et al. (2021), Baptista et al. (2021), Cakir et al. (2021), Dangut et al. (2021)
			Support Vector Machines	Susto et al. (2015), Lasisi and Attoh-Okine (2018), Shafi et al. (2018), Shamayleh et al. (2020), Gohel et al. (2020), Proto et al. (2020), Acernese et al. (2020), Orrù et al. (2020), Aremu, Hyland-Wood, and McAree (2020), Pałasz and Przysowa (2019), Khorsheed and Beyca (2021), Baptista et al. (2021), Cakir et al. (2021)
			Artificial Neural Network	Li et al. (2017), Orrù et al. (2020), Pałasz and Przysowa (2019), Baptista et al. (2021)
			DL—Deep Neural Networks	Oh and Lee (2020)
			DL—Convolutional Neural Networks	Schlagenhauf and Burghardt (2021)
			DL—Recurrent Neural Networks	Zhang, Liu, et al. (2018), Demidova (2020), Baptista et al. (2021)
			Hybrid models	Cakir et al. (2021)
Multiclass classification	Data labeled with distinct failure modes	Detailed failure prediction	K-Nearest Neighbors	Khodabakhsh et al. (2018), Casoli et al. (2019), Xu et al. (2021)
			Logistic Regression	Keartland and Van Zyl (2020)
			Decision Trees	Una et al. (2017), Khodabakhsh et al. (2018), Keartland and Van Zyl (2020), Casoli et al. (2019)
			Support Vector Machines	Una et al. (2017), Casoli et al. (2019), Xu et al. (2021)
			Bayesian models	Consilvio et al. (2020), Xu et al. (2021)
			Artificial Neural Networks	Rafique et al. (2018), Khodabakhsh et al. (2018), Keartland and Van Zyl (2020), Koca et al. (2020), Xu et al. (2021)
			DL—Deep Neural Networks	Zhou et al. (2017)
			DL—Convolutional Neural Networks	Zhang, Li, and Ding (2019), Z. Chen et al. (2019), Yang, Zheng, et al. (2019), Yang, Lei, et al. (2019), Song et al. (2021), C. Yang et al. (2021), Souza et al. (2021)
			DL—Recurrent Neural Networks	Wu et al. (2018), Wang et al. (2018), Nguyen and Medjaher (2019), Liang et al. (2020)
			DL—Extreme Learning Machines	Z. Chen et al. (2019), K. Li et al. (2019), Pang et al. (2020)
			Hybrid models	Zhang, Li, Wang, et al. (2019), Qian et al. (2019), Savitha et al. (2020), Liu et al. (2021), Aqueveque et al. (2021)

TABLE 4 (Continued)

DM task	Type of data	PdM problem	ML technique	References
		Prediction of gradual states of degradation	K-Nearest Neighbors	Satishkumar and Sugumaran (2017), Hu et al. (2020), Chen, Hsu, et al. (2021)
			Logistic Regression	Leahy et al. (2018), Quatrini et al. (2020)
			Decision Trees	Leahy et al. (2018), Quatrini et al. (2020), Hu et al. (2020), Panicucci et al. (2020), Chen, Hsu, et al. (2021)
			Support Vector Machines	Leahy et al. (2018), Quatrini et al. (2020), Venkataswamy et al. (2020), Cheng et al. (2020), Hu et al. (2020), Chen, Hsu, et al. (2021), Aqueveque et al. (2021)
			Bayesian models	Hu et al. (2020), Ruiz-Sarmiento et al. (2020), Aqueveque et al. (2021)
			Artificial Neural Networks	Ullah et al. (2017), Srivastava et al. (2018), Quatrini et al. (2020), Cheng et al. (2020), Hu et al. (2020)
			DL—Deep Neural Networks	Chen, Hsu, et al. (2021)
			DL—Recurrent Neural Networks	Zhang, Zhang, and Li (2019), Fernandes et al. (2020), Bampoula et al. (2021), Chen, Zhu, et al. (2021), Usuga-Cadavid et al. (2021)
Multilabel classification	Data labeled with several failures at each moment	Simultaneous failures prediction	Support Vector Machines	Oluwasegun and Jung (2020)
			Deep Learning	Wang, Zhang, et al. (2020), Pezze et al. (2021)

known labels. This approach has a double goal: (1) learning from labeled training data and generating labels for available unlabeled data and (2) generalizing to new data. In this scenario, semi-supervised learning can leverage these two type of data simultaneously in a more beneficial way than using only labeled (supervised learning) data or only unlabeled data (unsupervised learning). PdM is linked in many cases to data-intensive generation environments in a context of Industry 4.0 and sensor communication. In this context, data labels may be difficult to obtain because they require human annotators, special devices, or expensive and slow experiments, so generative classification has a great potential.

Table 6 shows unsupervised and semi-supervised paradigms related with different PdM problems, as well as the ML techniques used for each task. On the one hand, Section 4.4.3 addresses in detail the different proposals based on unsupervised learning in the last years. On the other hand, Section 4.4.4 details the reviewed PdM proposals that apply the semi-supervised learning approach.

4.4.1 | Classification in PdM

Nowadays, the most common type of PdM problem has labeled data with a discrete label that refers to the current working condition of the monitored system. This data type is addressed as a classification problem, where the main goal is to predict if the system operation is correct or, on the contrary, there is a fault in the system. Early PdM works are mainly based on binary classification. This approach aims to find a model to predict whether the system will fail. Other proposed works addressed the problem from the multi-class classification perspective. Thus, different failure modes or more than one degradation state of the same failure are studied. Finally, some approaches monitored multicomponents systems in which several failures can simultaneously occur. This context employs the paradigm of multi-label learning.

This section details the different ML techniques proposed in the context of classification applied to PdM.

TABLE 5 PdM problems from a supervised perspective II (regression)

DM task	Type of data	PdM problem	ML technique	References
Regression	Data labeled with another continuous variable	Remaining useful life estimation	Linear Regression	Saranya and Sivakumar (2020), F. K. Wang and Mamo (2019), Kang et al. (2021)
			Decision Trees	Saranya and Sivakumar (2020), Kang et al. (2021), Azab et al. (2021), Ayvaz and Alpay (2021)
			Support Vector Regressions	Yang et al. (2017), Kang et al. (2021), Ayvaz and Alpay (2021)
			Bayesian models	Benker et al. (2021)
			Artificial Neural Networks	Liao et al. (2016), Sampaio et al. (2019), Koca et al. (2020), Kang et al. (2021), Ayvaz and Alpay (2021), Bharti and McGibney (2021)
			DL—Convolutional Neural Networks	Aremu, Hyland-Wood, and McAree (2020), Aremu, Cody, et al. (2020), J. Li and He (2020), Pillai and Vadakkepat (2021), Resende et al. (2021), Ayodeji et al. (2021)
			DL—Recurrent Neural Networks	Zhang, Wang, et al. (2018), D. Wang et al. (2018), Lepenioti et al. (2020), Wang, Bu, and He (2020), C. Chen et al. (2020), Mode et al. (2020), Zschech et al. (2019), Chen, Hsu, et al. (2021), Chen, Liu, et al. (2021), Xiong et al. (2021), Kamat et al. (2021), Gribbestad et al. (2021), Chui et al. (2021)
			DL—Extreme Learning Machines	Berghout et al. (2021)
		Prediction of the evolution system behavior	Linear Regression	Zenisek, Holzinger, and Affenzeller (2019), Zenisek, Kronberger, et al. (2019), Schlagenhauf and Burghardt (2021), Chang et al. (2021)
			Symbolic Regression	Zenisek, Holzinger, and Affenzeller (2019), Zenisek, Kronberger, et al. (2019)
			Decision Trees	Zenisek, Holzinger, and Affenzeller (2019), Zenisek, Kronberger, et al. (2019)
			Artificial Neural Networks	Zhou et al. (2019), Crespo Márquez et al. (2020), de Carvalho Chrysostomo et al. (2020), Santolamazza et al. (2021)
			DL—Convolutional Neural Networks	Sun et al. (2021)
			DL—Recurrent Neural Networks	Morariu et al. (2020), Fernández-Barrero et al. (2021), Sun et al. (2021)

K-nearest neighbors

The principle behind distance-based methods is to find a predefined number of training samples closest to the new point, and predict the label from these. K-Nearest Neighbors (kNN) is a type of instance-based learning based on the k nearest neighbors of each query point, where k is a parameter that indicates the amount of neighbors to consider. In PdM, this technique has been applied in classification problems with two or more classes corresponding to different health states of the monitored systems and where all data are available at training time, since the arrival of new instances would mean recalculating the whole model. Thus, in binary scenario, kNN has been compared with other

TABLE 6 PdM problems from an unsupervised and semi-supervised learning perspectives

DM task	Type of data	PdM problem	ML technique	References
Clustering	Unlabeled data whose distribution can be divided into different categories	Identification of different working conditions	Partitional clustering	Yang et al. (2017), Cupek et al. (2018), Uhlmann et al. (2018), Casoli et al. (2019), Oliveira et al. (2020), Bekar et al. (2020), Wang, Liu, et al. (2020), Cerquitelli et al. (2021), Kamat et al. (2021), Serradilla et al. (2021), Giordano et al. (2021)
			Hierarchical clustering	Zschech et al. (2019)
		Identification of infrequent patterns that may lead to failure	Density-based clustering	Khodabakhsh et al. (2018), Hsu et al. (2020), Serradilla et al. (2021)
			Decision Trees	Aremu, Hyland-Wood, and McAree (2020), Kolokas et al. (2020)
			Deep Learning	Malawade et al. (2021)
One-class classification	The data used for training are implicitly known to correspond to the normal operation of the system	Identification of new data that does not fit the previously learned model and may lead to failure	Support Vector Machines	Ribeiro et al. (2016), Casoli et al. (2019), Oliveira et al. (2020), Aremu, Hyland-Wood, and McAree (2020), Aremu, Cody, et al. (2020), Morariu et al. (2020), Serradilla et al. (2021)
			Deep Learning	Ribeiro et al. (2016), Cho et al. (2020), Bouabdallaoui et al. (2021), Fernández-Barrero et al. (2021), Basora et al. (2021), Gribbestad et al. (2021), Ning et al. (2021), Kim et al. (2021), Serradilla et al. (2021), Fathi et al. (2021)
			Hybrid models	Yu et al. (2020), Oliveira et al. (2020), Serradilla et al. (2021)
Generative classification	Small proportion of labeled data, which is enough to characterize all the working conditions	Failure prediction	Deep Learning	Zhang et al. (2021)

ML methods to estimate the probability of failure in the next time window (Baptista et al., 2021; Susto et al., 2015), and in a multicomponent fault prediction system (Shafi et al., 2018). Other example is found in Cakir et al. (2021), where kNN for binary classification is integrated in a monitoring system for PdM is able to trigger alarms when the number of detected failures exceeds a threshold. Attending to multiclass scenario, kNN has been studied together with other methods in failure identification (Xu et al., 2021) and in the categorization of anomalies in data streams with possible failures (Khodabakhsh et al., 2018). In Casoli et al. (2019), weighted and medium kNN algorithms are compared together with other classification methods to detect the health state through the analysis of vibration signals. Also in the context of vibration signal analysis, in Hu et al. (2020), it is proposed a kNN algorithm which achieves competitive performance in detecting the degradation state. In Chen, Hsu, et al. (2021), kNN is studied together other methods to detect several degradation states over multivariate time-series. Other approximation to multistate degradation detection is found in Satishkumar and Sugumaran (2017). It applies a variation of kNN called KStar which uses entropy-based distance function to find the closest samples.

Logistic regression

Logistic Regression (LR) is a linear model in which a logistic function is trained via maximum likelihood estimation in an iterative way to produce the probabilities of data belonging to the different classes. LR has been studied together with other methods in Leahy et al. (2018), Quatrini et al. (2020), and Keartland and Van Zyl (2020) to predict the degradation state and the failure mode, respectively, among several classes. Attending to binary classification, in Proto et al. (2020), LR is compared to other methods to predict health status after the occurrence of several control events. In Gohel et al. (2020), a RUL estimation problem is transformed to classification by labeling the data according to whether the failure will occur in the next n cycles or not. Thus, this is addressed as a probabilistic problem. The effectiveness of LR is studied as a function of the value of n, that is, how far in advance the failure will be predicted.

Decision trees

A Decision Tree (DT) predicts the value of a target variable by learning simple decision rules inferred from the data features. In order to decrease the impact of over fitting, ensembles include randomness by creating multiple trees (estimators), deriving bootstrapped samples and splitting nodes among a random subset of features. DTs are very popular in classification PdM problems because of their properties related to interpretability and explainability, very useful to provide extra information about the possible causes of the predicted failure. Thus, in (de Carvalho Chrysostomo et al. (2020), a DT is applied to the output of a future signal estimation to provide an automatically decision support system. Other study case is shown in Aremu, Hyland-Wood, and McAree (2020) where the DT is compared with other models to classify normal and failure records after dimensionality reduction, achieving the best performance. In Yang et al. (2017), the classification of potential failures is carried out with a DT model that is refined with other techniques. Conditional Inference Tree (CIT) is a variation of DT whose use has also been reported in several PdM works. In CIT, a significance test is used to select input variables rather than selecting the variable that maximizes the information measure. In Khodabakhsh et al. (2018), CIT is studied together other methods achieving the best performance in the task of categorizing anomalies in data streams with possible failures. Other DT variations are studied in Patil et al. (2021) to predict the probability of failure in the next period of time evaluated. Specifically, CART and CHAID DTs are studied, characterized by using GINI index and Chi-Square, respectively, to carry out the partitions of the tree.

DTs are typically inserted in different variations of classifier ensembles that combine the results of several simple classifiers in a single output. In the context of PdM, Random Forest (RF) is the most popular type of ensemble learning. It performs an ensemble of uncorrelated DTs which are randomly generated using bagging and boosting to fit the given data. Recently, RF can be found in several PdM works of binary classification (Baptista et al., 2021; Dangut et al., 2021; Lasisi & Attoh-Okine, 2018; Mishra & Huhtala, 2019; Pałasz & Przysowa, 2019; Prytz et al., 2015; Su & Huang, 2018), multiclass classification of several degradation states (Chen, Hsu, et al., 2021; Hu et al., 2020; Leahy et al., 2018; Quatrini et al., 2020), and different failure modes (Keartland & Van Zyl, 2020), being compared with other methods and achieving the best performance in most cases.

Other common ensemble technique is Gradient Tree Boosting (GTB) where a sequence of small DTs is fitted on repeatedly modified versions of the data to arbitrary differentiable loss functions. In Proto et al. (2020), GTB is studied together with other methods in a binary classification based on predicting health status after the occurrence of several control events. In Una et al. (2017), this boosting method is also employed to detect several failure modes. Extreme Gradient Boosting (XGB) is an implementation of GTB for speed and performance which has made it suitable for PdM. Thus, in Steurtewagen and Van den Poel (2021), GTB is employed to detect if a system will fail or not. Additionally, the results of the predictive models are explained by a post analysis based on Shapley values. The Shapley value is a way to distribute the total prediction to the variables as a sort of payout, which can be very useful in black-box models in PdM to provide insight over the root cause of the failure.

Many authors focus exclusively on comparing different DT techniques in different PdM problems. Attending to binary classification (Shafi et al., 2018), DT and RF are compared in Shafi et al. (2018) and Hsu et al. (2020) together with other ML models in a multicomponent fault prediction system. DT achieves the best results. Other example is found in Cakir et al. (2021). It studies DT and RF together with other models for binary classification using a monitoring system for PdM that is able to trigger alarms when the number of detected failures exceeds a threshold. In Kaparthi and Bumblauskas (2020), CIT and RF are studied. RF achieves the best performance. In Calabrese et al. (2020), RF, GTB, and XGB are compared. XGB achieves the best accuracy; and Khorsheed and Beyca (2021) compares DT and

RF. GTB achieves the best performance. In multiclass problems, there are also problems focused on comparing DT methods: in Panicucci et al. (2020), the prediction of the degradation level is addressed with DT, RF, and GTB, being RF the most accurate model. In Casoli et al. (2019), DT and RF are compared together with other methods to detect several failure modes achieving the best performance with RF.

Support vector machines

A support vector machine (SVM) is a model-based classifier that constructs a hyperplane or set of hyperplanes in a high-dimensional space by the maximization of the distance to the nearest training-data point of any class. Many researches use SVM for PdM since could reliably separate different fault conditions and identify the severity of incipient faults. With SVM, several kernel types can be used in order to project the data in higher-dimensional spaces.

The review has shown that the most common type used in PdM is linear kernel that indicates that in most cases the different events are linearly separable. Thus, linear SVM has been used in binary classification (Cakir et al., 2021; Gohel et al., 2020; Khodabakhsh et al., 2018; Lasisi & Attoh-Okine, 2018; Pałasz & Przysowa, 2019; Proto et al., 2020; Shamayleh et al., 2020; Susto et al., 2015) to classify the events according to their proximity to the failure threshold. In multiclass scenarios, linear SVM has been studied together with other methods in both prediction of multiple degradation states (Chen, Hsu, et al., 2021; Cheng et al., 2020; Quatrini et al., 2020; Shafi et al., 2018; Venkataswamy et al., 2020), and several failure modes detection (Una et al., 2017; Xu et al., 2021). Specifically, in Cheng et al. (2020) and Shafi et al. (2018), SVM achieves the best performance.

Another type of kernel widely used in PdM is the Radial Basis Function (RBF), that it is used when data are not linearly separable. Thus, in Susto et al. (2015), Acernese et al. (2020), Orrù et al. (2020), and Baptista et al. (2021), it is tested in binary classification problems to determine whether the system will fail or not in a prediction horizon.

The polynomial kernel is another approach for nonlinearly separable problems. In this review, it is reported a work (Hu et al., 2020) that uses it for a multiclass classification. This method is compared with linear and RBF SVMs, as well as other classifiers, in combination with several feature extraction methods in a scenario with several failures degrees.

Moreover, some works present a comparative study among several kernels in SVMs. Linear and RBF kernels in binary classification (Aremu, Hyland-Wood, & McAree, 2020), and multiple failures (Casoli et al., 2019; Leahy et al., 2018) scenarios. In Aqueveque et al. (2021), linear, RBF, and polynomial SVMs are compared, together with other methods, in the multiclass classification to predict health statuses. The comparisons report the best performance for RBF in all cases. This kernel is a more powerful method when the data are not linearly separable, anything very common when there are a lot of classes.

In Oluwasegun and Jung (2020), an application of SVM to a multilabel classification is presented. There are several motion types and several variations of the monitored component, and every combination has its health status. Thus, there are three simultaneous labels so the problem of finding the working condition is addressed from a multilabel perspective. Specifically, the label powerset transformation is applied to a multiclass algorithm, SVM in this case, by assigning a unique label to every possible label combination of the problem.

Bayesian models

Bayes' theorem provides a principle based way to compute a conditional probability. The probability that an event A occurs given that another event B has already occurred is equal to the probability that B occurs given that A has already occurred multiplied by the probability of occurrence of A and divided by the probability of occurrence of B. This has a direct relationship with PdM to relate the probability of failure occurrence given the current performance of the monitored system. Thus, several type of models based on Bayes have been used in PdM. In Hu et al. (2020), Aqueveque et al. (2021), and Baptista et al. (2021), several Gaussian Naive Bayes classifiers are studied in comparison with other classifiers to predict the degradation state of the monitored systems, in the case of Hu et al. (2020) in combination with several feature extraction methods. Similarly, in Xu et al. (2021), Naïve Bayes is studied together with other methods to perform failure identification in an edge computing scenario.

Another approach for predicting the degradation state based on Bayes is proposed in Ruiz-Sarmiento et al. (2020). They present a Discrete Bayes Filter that integrates prior domain knowledge of system evolution with the real-time monitoring. Thus, it is possible to dynamically update the probabilities of each one of the degradation states considered. In Consilvio et al. (2020), rare events are identified by an anomaly detection technique, and then, they are analyzed in a Bayesian Network composed by the prior probability of the rare event occurrence. A Direct Acyclic Graph models the network structure in order to extract conditional probabilities expressing the relationship among individual components failures and the reliability of the overall monitored system. On the other hand, in Consilvio et al. (2020), another case is also studied in which Petri Nets are used to identify the sequence of events that have given risen to an anomaly.

Artificial neural networks

Feedforward Artificial Neural Networks (ANNs) are computing models based on a collection of connected units (neurons) organized in layers and whose connections do not form cycles, so the information moves only in one direction. Depending on the architecture of the output layer, an ANN can perform classification or regression tasks. Multilayer Perceptron (MLP) is the most popular approximation to ANN, usually composed from one to three layers composed by a variable number of neurons. Neurons transform the values from the previous layer with a weighted linear summation, followed by a nonlinear activation function. ANN-based classification is presented in several PdM works because of its efficiency. However, in many cases these models are underestimated for lack of explainability. Attending to binary scenario, a binary prediction example can be found in Li et al. (2017) that is addressed with a generic ANN. In the same line, in Orrù et al. (2020) and Baptista et al. (2021), a MLP is compared with other ML techniques to predict whether the system will fail or not in the next time window. Another case of binary scenario is found in Pałasz and Przysowa (2019), where ANN outperforms other techniques but finally is used in an ensemble combined with SVM and DT. Attending to multiclass scenario, in Ullah et al. (2017) and Srivastava et al. (2018), different ANNs are used to detect several degradation states in systems such as power equipment and water treatment plants. In Quatrini et al. (2020), ANN performance is compared with other classifiers in combination with different feature preprocessing methods to classify working cycles in several degradation states in a multicomponent system. It achieves the best score in one of the components. Another case of predicting several degradation states is presented in Cheng et al. (2020) as part of a decision support system in which ANN is studied among other models. In Rafique et al. (2018), an ANN model is used to detect variations in the input stream that correspond to several failure conditions in the context of optical network maintenance, achieving better performance than fixed alarms at given threshold. In Khodabakhsh et al. (2018), Keartland and Van Zyl (2020), and Xu et al. (2021), ANN performance is studied together with other models to detect several failures, achieving the best performance in most cases. In Hu et al. (2020), MLP is also studied in comparison with other classifiers in combination with several feature extraction methods in a scenario with several degrees of failure. In Koca et al. (2020), a MLP is proposed to build a mixed model that performs both classification and regression tasks. The regression neurons in the output layer estimate the time of the failures in different areas, while the classification neurons indicate different categorical information of the failure like the area or the machine identifier.

DL methods – Deep neural networks

DL methods are based on ANN, extending its architecture to more layers and neurons in order to progressively extract higher-level features from the raw input. In this sense, the most basic approach to DL is considered to be an extension of the MLP feed-forward network called Deep Neural Network (DNN), in which there is a considerable increase in both neurons and layers. Thus, the basic foundation of applying activation functions that are activated at will would remain the same, but deeper features can be extracted. Several examples of DNN employment can be found in the recently state of PdM. In Oh and Lee (2020), where a framework to estimate missing data is proposed using a combination of Gaussian process regression and GAN. DNN is applied as the last part of the framework that learns from the processed dataset to predict whether the system will fail or not. Attending to multiclass classification, in Chen, Hsu, et al. (2021), a DNN model is compared together with other methods in the context of several degradation states prediction, and it achieves the best performance in the problem. Another approach is proposed in Zhou et al. (2017) using three levels. A hierarchical DNN with the first hierarchy devised for the purpose of working mode partition, a second hierarchical level to extract features separately of different working modes and diagnose the fault source, and a third level to distinguish the severity of a certain fault in a given mode.

DL methods: Convolutional neural networks

Following the family of DL methods presented in previous paragraph, this one is focus on CNNs. CNN is a DL extension of MLP that in the hidden layers performs subsequent convolutions, a mathematical operation that applies a certain filter that modifies the input space. This feature makes them especially suitable for detecting spatial features in the data, which makes them very relevant in image analysis. In this sense, in Schlagenhauf and Burghardt (2021), a framework is developed that uses CNNs to detect if there is damage on the imaged surface. In the next phase of the model, classical vision techniques are used to establish the severity threshold of the detected damage. In a last phase, another CNN model is trained with the previously labeled data to predict damage to different degrees. Although CNNs have their origin in the field of image processing, they can also be used to process data streams. Thus, in Zhang, Li, and Ding (2019), a model based on the combination of CNN with residual learning is proposed in order to identify multiple working conditions from vibration signal analysis. In Chen et al. (2019), it is proposed a framework designed to predict several failure modes from vibration signal analysis. Specifically, the CNN part receives time–frequency representation of the signals and produces two-dimensional structural features that are used by the last part of the model, an extreme learning machine network, to perform the classification. This work also compares its proposal with both traditional ML (kNN, SVM), and DL models (Deep Belief Network [DBN], Stacked Auto-Encoder [SAE] and simple CNN). Another framework focused on failures detection from vibration data is found in (Song et al., 2021). Several DL architectures are compared achieving the best performance with the CNN-based ResNet. As for other types of data, two examples studying the application of a CNN model for the detection of various types of faults from acceleration data can be found in (C. Yang et al., 2021; Souza et al., 2021). In (Yang, Zheng, et al., 2019), it is proposed a hybrid framework involving a CNN. In this case, it is applied on the last part of the classifier performing the prediction of several failure modes after a specific preprocessing to vibration signal analysis. This work also compares its proposal with traditional ML methods (SVM and RF) in combination with different feature extractors, as well as DL methods (DBN and SAE) in combination with different signal transformation methods. In Yang, Lei, et al. (2019), a CNN is proposed as the base of a transfer learning framework that is able to learn main characteristics of an artificially generated dataset, translate them and perform classification of multiple failure modes in a real-world dataset of the same domain.

DL methods – Recurrent neural networks

Recurrent Neural Networks (RNNs) are another type of DL. Unlike feedforward ANNs, they have connections with loops between nodes, adding feedback and memory to the model over time. This memory allows this type of network to learn and generalize across sequences of inputs rather than individual patterns. Within RNN, LSTM are a very popular model, since has been shown to be particularly effective to handle sequence-dependence when stacked into a deep configuration. In the context of PdM, this implies that they will be applied in time-series problem, where the input data will be running records with an associated timestamp. In Zhang, Liu, et al. (2018), Demidova (2020), Baptista et al. (2021), studies of different RNN configurations are carried out to analyze the performance of a binary classification that determines whether the system will fail or not. Specifically, three types of RNNs are studied: simple RNN, Gated Recurrent Unit (GRU), and LSTM, with GRU and LSTM obtaining a similar performance in all cases. Attending to multiclass problems, different LSTM configurations are studied in Nguyen and Medjaher (2019), Fernandes et al. (2020), Zhang, Zhang, and Li (2019), Chen, Zhu, et al. (2021) to obtain the probability of future failure from time-series sorted operating records. Specifically, in Nguyen and Medjaher (2019), the LSTM predicts the failure probability in different time-windows, while in Fernandes et al. (2020), Zhang, Zhang, and Li (2019), Chen, Zhu, et al. (2021), it is predicted the state of degradation of the system. In Wu et al. (2018), it is proposed a hybrid DL model that combines a first stage of CNN layer that extracts spatial and short-term temporal features of the multidimensional signals inside a segmented time window, and then several LSTM layers are stacked to learn the internal long-term temporal features of each subsequence. Another hybrid proposal is presented in Liang et al. (2020) where several failure modes are identified from vibration signal analysis. It is integrated a LSTM at the beginning of the network to reduce the impact of unexpected noise, and then, two CNN-based steps are considered: a dilated convolution for information extraction, and a capsule network step for feature comprehension in sequences. The proposal of this work is compared with other DL methods such as CNN, LSTM, SAE, and CapsNet. Another hybrid approach is found in Bampoula et al. (2021). It is based on the time-series reconstruction through an AE model composed of stacked Long Short-Term Memory. The work proposes to train a single LSTM-AE for each health status, that is, each model is trained with time-series of an unique class. In the prediction phase, the time-series is reconstructed through all the models and is assigned to the class of the model with the smallest reconstruction error. Attending to GRU in multiclass classification, in Wang et al. (2018), it is used to detect several fault conditions in a configuration that includes local feature extraction and bidirectional recurrent structure. Another approach in RNN processing is explored in Usuga-Cadavid et al. (2021). It employs DL models from the natural language processing field to analyze maintenance logs provided by operators in order to predict the breakdown duration.

RNNs have been explored also in multilabel classification. Thus, in Pezze et al. (2021), it is addressed the problem of alarm forecasting in a complex system with multiple subparts that produce alarms at different time-stamps. The objective is to predict the combination of alarms that will occur in a certain future time-window. Different RNN architectures are explored: a bidirectional recurrent model based on GRU, a model that combines an attention mechanism with GRU and a model based on the Transformer networks, adapted from the field of Natural Language Processing.

DL methods – Extreme learning machines

Extreme Learning Machines (ELMs) are another type of feedforward ANN in which the training process requires to adjust not only the weights between the connections in the hidden layers, but also other parameters of the hidden nodes. In this case, not all nodes are equal in the same layer, as is the case of MLP, CNN, or RNN. In PdM, several studies can be found, all of them applied to vibration signal analysis to predict several failure modes. Thus, in Chen et al. (2019), a hybrid framework composed of a CNN and an ELM is proposed. The ELM receives the features extracted by the CNN and identifies the associated failure mode through random feature mapping and linear parameters in a more automatized way than with traditional ML methods or DL. In Li et al. (2019), a deep ELM is proposed by combining an SAE model with ELM in order to optimize the learning efficiency. Thus, firstly the compressed data representation is obtained with an ELM-based SAE, and in the second part, it is performed the classification with the classical ELM-based supervised algorithm. Moreover, in order to improve the feature extraction process, the first part is combined with sparsity and neighborhood preserving theory. In Pang et al. (2020), an ELM ensemble kernel is proposed that integrates kernel functions to improve the generalization capacity and introduces the ensemble on decision-making level.

In Wang, Zhang, et al. (2020), a double ELM model is proposed for multilabel classification in order to identify compound faults, that is, detecting simultaneously several mechanical faults in rotating machinery. Thus, the presented model is composed of two ELMs. The first one performs instance clustering by random featuring mapping, and the second one performs the multilabel classification through Gaussian style activation functions between each target into a multi-output layer, where each node corresponds to one of the label of the problem.

Hybrid models

The models categorized in this section are characterized by combining a first part of unsupervised learning that analyzes different properties and features in the original data, with a second part that maps the features learned with the labeled classes. In Zhang, Li, Wang, et al. (2019); Qian et al. (2019), two different feature distribution learning models applied to vibration data classification are presented for multiple working conditions. Both works present a complete framework that combines PCA (already presented for data preprocessing in Section 4.3.2) and other methods for feature extraction in signals, a feature distribution learning method, and a final softmax classifier that relates the learned features to the target labels to be predicted. Thus, in Zhang, Li, Wang, et al. (2019), it is used a general normalized sparse filtering that focuses on optimizing the sparsity of the learned representations, while in Qian et al. (2019), a transfer learning approach is followed using joint distribution adaptation to align both the marginal and conditional distributions of the target problem, and a previously learned dataset. Closely related to PCA, LDA was also presented in previous section. This method has been also studied for classification tasks based on the linear decision boundary generated by fitting class conditional densities to the data. Thus, LDA can be found in binary classification for generic failure detection (Cakir et al., 2021) and in multiclass classification for health status degradation (Aqueveque et al., 2021). Besides, a generalization of LDA called Quadratic Discriminant Analysis is also studied in Aqueveque et al. (2021). The study of Savitha et al. (2020) presented a Restricted Boltzmann Machine (RBM) that learns the probability distribution of the data in an online approach: the hidden layer of the network adds a neuron and/or updates the representations of the existing neurons depending on the novelty of the data streams that come. The next phase of the model uses another layer that performs supervised discriminate learning to associate the feature representation with the class labels. The Liu et al. (2021) presented a complete maintenance decision model which core is a GAN built based on LSTM. The model is composed by an encoder, a decoder, a discriminator, and an additional encoder, with the aim of learning in an unsupervised manner to reconstruct the temporal series. Thus, the components are composed of several layers of LSTMs. In a second stage, the reconstruction error in the latent features space between the two encoders is evaluated based on thresholds learned from the labeled data. Thus, different levels of error will lead to different health states. Besides, the framework maintains a GAN-LSTM for each of the subcomponents in the monitored system, and provides a general state based on the combined outputs of the models and the rules of an expert system.

4.4.2 | Regression in PdM

A classic problem in PdM is the estimation of the RUL. It is modeled as a regression problem where the average component degradation must be estimated based on the relationship between the target variable and other variables or even analyzing past states of variables. Most PdM problems from a regression perspective are characterized by estimating the temporal evolution of a variable related to the degradation state of the monitored system.

This section details the different ML techniques proposed in the context of regression applied to PdM.

Linear regression

Linear Regression (LiR) refers to a multivariate linear combination of regression coefficients (i.e., constants and weights of input variables). The coefficients are estimated by the generalized least square technique. Although LiR is a deterministic and parameterless technique, it has important advantages related to its interpretability and training speed. Thus, this technique was recently used to predict the concept drift in a data-streaming environment (Zenisek, Holzinger, & Affenzeller, 2019; Zenisek, Kronberger, et al., 2019). Another example of use can be found in Saranya and Sivakumar (2020) and Kang et al. (2021), where different multivariate LiR are employed, respectively, to estimate RUL. In Schlagenhauf and Burghardt (2021), LiR is the last step in a decision support system that estimates the evolution of a damage area in the monitored surface. In Chang et al. (2021), different regression methods are compared to analyze complex manufacturing data and predict the future behavior of the monitored machine. Specifically, LiR, Least Absolute Shrinkage Selector Operator and Ridge and Elastic Net regressors are compared.

Symbolic regression

SR refers to models in the form of a syntax tree consisting of arbitrary mathematical symbols (terminals: constants and variables, nonterminals: mathematical functions), which can be seamlessly translated to plain mathematical functions. For target estimation syntax trees are evaluated top-down. Syntax trees are developed using the stochastic genetic programming technique from the field of evolutionary algorithms. This technique is explored to predict the concept drift from a data-streaming perspective in Zenisek, Holzinger, and Affenzeller (2019) and Zenisek, Kronberger, et al. (2019).

Decision trees

DTs have been introduced in Section 4.4.1. Although these models were initially developed for classification tasks, they have been adapted to regression by performing the target estimation by averaging the individual tree estimations. Random Forest Regression (RFR) is the adaptation of RF to regression very popular in PdM. It is employed to predict RUL (Kang et al., 2021; Wang & Mamo, 2019) and also the concept drift from a data-streaming perspective (Zenisek, Holzinger, & Affenzeller, 2019; Zenisek, Kronberger, et al., 2019). In Azab et al. (2021), the performance of two ensembles is compared for RUL estimation: on the one hand RFR and on the other hand the Boosted Decision Tree, an adaptation for regression of GTB previously introduced in Section 4.4.1. Other example of DTs ensembles for RUL prediction is found in Ayvaz and Alpay (2021) that compares RFR, GTB, XGB among other ML methods. In both cases, RFR achieves the best performance. Another DT ensemble reported in the last years in the field of PdM is Extra Tree Regression (ETR). ETR splits nodes using random subdivisions of features, but with two significant variances: samples without replacement and nodes are fragmented on random splits. Thus, ETR can be found in Saranya and Sivakumar (2020) used to estimate RUL.

Support vector regressions

SVM were introduced in Section 4.4.1 as they are mainly used for classification problems. However, they can be extended to regression. In Support Vector Regression (SVR), instead of using the curve as a decision boundary, it uses the curve to find the match between the vector and position of the curve. Support vectors help in determining the closest match between the data points and the function which is used to represent them. In Kang et al. (2021) and Ayvaz and Alpay (2021), SVR is compared together with other methods to estimate RUL in different multivariate time-series problems. In Yang et al. (2017), SVR is used to refine the RUL estimation as a final step in a process that first detects if a fault can occur classifying time-series.

Bayesian models

As it was introduced in classification section (see Section 4.4.1), the Bayes approach is characterized by inferring a probability distribution over all possible values. Following this principle, the GPR is a nonparametric model that calculates the probability distribution over all admissible functions that fit the data. Thus, the model specifies a prior function, calculates the posterior function using training data, and computes the predictive posterior distribution on the points of interest. In Benker et al. (2021), it is used a GPR model with the aim of estimating RUL. The proposed approach focuses on data efficiency, needing very less data and time to achieve the competitive results compared to the state of the art.

Artificial neural networks

ANNs were introduced in Section 4.4.1, as they are able to perform classification and regression tasks depending on the configuration of the output layer. Thus, different implementations of MLP are employed in Koca et al. (2020), Sampaio et al. (2019), Kang et al. (2021), Ayvaz and Alpay (2021) to predict the time of the failure, as well as to predict the behavior of continual signals in Crespo Márquez et al. (2020), de Carvalho Chrysostomo et al. (2020), Santolamazza et al. (2021). In the case of de Carvalho Chrysostomo et al. (2020), the forecasting of the critical variables is part of a complete framework for decision support. In Bharti and McGibney (2021), MLP is used as a baseline model to implement a federated learning framework that enables several clients to train local predictive models that are updated by the other clients preserving their privacy. Beyond MLP, other types of ANN have been explored in regression task for PdM. Thus, the study of Zhou et al. (2019) used ELMs to predict the future evolution of a signal. ELMs were introduced in Section 4.4.1 both from a DL perspective and shallow learning. A RBM is another type of ANN also explored in the literature that is characterized by a particular form of log-linear Markov random field that forms a two-layer network of a visible layer and a hidden layer. The study of Liao et al. (2016) presented a variation of RBMs that adds a regularization term to automatically generate features which potentially better represent the degradation pattern of a system. Once the features are extracted, the RUL estimation is carried out with a similarity-based approach.

DL methods – Convolutional neural networks

DL models were introduced in Section 4.4.1 as an extension of ANNs family. As in the rest of ANN, CNN can perform regression or classification tasks depending on how the output layer of the model is configured. In this line, several PdM works have explored CNN-based regression to estimate RUL (Aremu, Cody, et al., 2020; Aremu, Hyland-Wood, & McAree, 2020), after using different feature engineering frameworks. Other example of RUL prediction with a CNN model is found in Resende et al. (2021). It integrates the model in a complete platform for PdM based on a modular software solution for edge computing gateways. In the case of Pillai and Vadakkepat (2021), the CNN model starts from high-dimensional data extracted with an AE. Then, the temporal features and encoded representation are concatenated, forming convolutional composite features that are used to train the model for RUL prediction. In Li and He (2020), a CNN is also used to predict RUL, introducing auto-configuration of its parameters by Bayesian optimization and adaptive-batch normalization. Auto-tune of the CNN model is also explored in Ayodeji et al. (2021), which proposes a causal, augmented CNN focus on long sequence time-series prediction and applied it to RUL prediction. The Hyperband algorithm is used for the auto-tuning process. From the perspective of behavior evolution, CNN is also employed in Sun et al. (2021) to predict the future evolution of energy consumption.

DL methods – Recurrent neural networks

RNNs were introduced in Section 4.4.1, since they are able to perform classification or regression tasks depending on the configuration of the output layer. Regression-based LSTM to predict RUL can be found in Zhang, Wang, et al. (2018), Zschech et al. (2019), Chen, Liu, et al. (2021), and Xiong et al. (2021). Two decision support systems are proposed in Lepenioti et al. (2020) and Chen, Zhu, et al. (2021) based on the RUL estimation produced by LSTM, among other components. In Kamat et al. (2021), different LSTMs configurations are studied to predict RUL, with the particularity of using only the last section of the signal from the detection of the first anomalous point, thus the computational load is reduced. Clustering analysis is employed to detect that first anomaly. In Wang, Bu, and He (2020), a slightly different approach is followed based on the way that data are presented to the model in order to predict failure times. Another approach of RUL estimation that predicts the time between failures is found in Chen et al. (2020). It highlights the need for proper preprocessing to minimize risk factors in LSTM performance such as data scatter and missing values. From the studies of Morariu et al. (2020) and Sun et al. (2021), it can be found that LSTM models learn patterns and variations in numerical measurements concerning energy consumption in order to produce forecasts. A similar approach is followed in Fernández-Barrero et al. (2021) that uses LSTMs to predict future behaviors of the monitored signals at different time horizons. In Chui et al. (2021), it is proposed a hybrid method for RUL prediction composed of a generic RNN and a LSTM whose outputs are combined using weighting factors optimized by a genetic algorithm. The aim of the proposal is to combine short- and long-term predictions. Another popular type of RNN is GRU, similar to LSTM but with less parameters (e.g. lack of output gate), which may be more effective in tasks with smaller- or less-frequent datasets over time. In Wang et al. (2018), a GRU model that includes local feature extraction and bidirectional recurrent structure is used to predict the RUL. Mode et al. (2020) evaluated the performance in RUL estimation under false data injection attacks of three DL models: GRU, LSTM, and CNN, in which GRU being the most robust method.

Transfer learning is also explored in the field of RUL prediction with LSTMs. Specifically, in Gribbestad et al. (2021), the predictive model is trained with a public dataset with more samples, and then tested in their case study.

DL methods – ELMs

ELMs were introduced in Section 4.4.1, since they are able to perform classification or regression tasks depending on the configuration of the output layer. Regression-based ELM has been recently explored to predict the RUL in Berghout et al. (2021). In this work, the proposed approach is a type of deep belief neural network based on online sequential ELM rules which has the capacity to perform convolutional mapping as well as the pooling in each single subnetwork from its hidden layers according to ELM with local receptive fields theories.

4.4.3 | Unsupervised learning in PdM

Nowadays, it is increasingly common in PdM that the data come from a real environment in which the labels are not available. Thus, it is not known a priori what anomalous behaviors or failures have to be detected, but rather the aim is to identify the different operating regimes of the monitored systems. In these situations, the ML methods that are to be applied must work without such information and should look for inherent characteristics of the data to characterize them. This is known as unsupervised learning, and in PdM it can be applied in two situations. On the one hand, separating data into a series of differentiated operating modes, related to the ML task of clustering. On the other hand, detecting infrequent patterns that may indicate anomalous behavior in data, which mainly corresponds to a normal operating regime. These patterns may indicate anomalous behavior related to an incipient failure, which is related to the outlier detection task.

This section details the different ML techniques that have been proposed in this context.

Partitional clustering

In DM, clustering is the task of grouping data into various groups in such a way that data belonging to the same cluster are as similar to each other as possible while being as distinct as possible from items in other clusters. Although there are several approaches to perform cluster analysis, in PdM the use of the K-Means algorithm, exponent of partitional clustering, has been mainly reported. In K-Means, k random centroids are chosen initially, being each data point assigned to the nearest centroid. Once each data point has been assigned to a centroid, and thus to a cluster, the positions of the k centroids are recalculated. The process ends when the centroids no longer move. The number of groups (k), which should be provided by the user, in the context of PdM means that there must be an idea of how many operating regimes have to be identified a priori. In Cupek et al. (2018), it is used over the energy consumption historical of the monitored system to find abnormal patterns that may correspond to failures. In Uhlmann et al. (2018) and Casoli et al. (2019), a number of more informative variables are selected according to the problem domain, and K-Means is applied to identify the most influential patterns in the damage of earthwork assets and laser melting machine tool, respectively. The studies of Oliveira et al. (2020) and Bekar et al. (2020) explain two examples of using K-Means over the projections of the data given by PCA analysis in order to group the data and detecting anomalies based on the least dense clusters. In Wang, Liu, et al. (2020), K-Means is used in an offline phase to group temporal series according to their vibration patterns. Then, in an online phase the centroids of the clusters are used as a classifier to identify new incoming temporal series. Similarly, in Yang et al. (2017), K-Means is used to refine a first failure detection grouping temporal series to apply over them the most suitable regression model to estimate RUL. In Kamat et al. (2021), K-Means is employed to detect the first anomalous point in the signal, that triggers the RUL prediction over the rest of the signal. Other approach to perform partitional clustering is the Gaussian Mixture Clustering (GMC) that involves the superposition of multiple Gaussian distributions and assumes that each of the distributions represents a cluster. Thus, the model tends to group the data points belonging to a single distribution together. This model is compared with K-Means and Bisecting K-Means in Cerquitelli et al. (2021) to identify different working conditions in the monitored system. After the cluster identification, several DM tasks are employed to identify the most relevant features of each cluster. Other use of GMC is explored in Serradilla et al. (2021). It applies it over previous detected anomalies to explain more about their characteristics. The study of Giordano et al. (2021) proposed a novel partitional clustering specifically focuses on multivariate time-series grouping. K-MDTSC (K-Multi-Dimensional Time-Series Clustering) is based on a generalization of K-means and a generalized notion of distance that can able to handle synchronous multidimensional time series.

It allows K-MDTSC to create clusters without the need to transform or dimensionality reduction. The proposed method is tested on a PdM real scenario to identify different working conditions.

Hierarchical clustering

In hierarchical clustering, the aim is to group data at different levels, so that the clusters formed are organized in a hierarchy whose typical representation is a dendogram. The two possible clustering strategies are agglomerative, that is, all points start in their own cluster, or divisive, that is, all points start in a single cluster. In Zschech et al. (2019), this type of clustering is employed to generate labels in the context of RUL, in such a way that characterize time series of working in order to use them train a prognostic model.

Density-based clustering

As opposed to the partitional clustering explained previously, density-based clustering does not look for dividing the available data in a number of clusters, but rather it looks for the most densely populated areas, and if there are patterns that are significantly more separated from the rest, they are not assigned to any cluster. This technique has the advantage of being halfway between clustering and outlier detection, which in PdM implies that it can be used to detect different operating patterns, as well as anomalies that may lead to failure. Density-based Spatial Clustering of Applications with Noise (DBSCAN) is the most popular implementation of density-based clustering. Thus, in PdM, it begins to be explored in some works. In Hsu et al. (2020), DBSCAN is used to identify anomalies, that is, observations in low-density areas. Then, these data are analyzed in order to obtain the most significant variables in the identification of failures. Other application is found in Khodabakhsh et al. (2018), where, as the data streams arrive, DBSCAN is applied for operating state identification. Thus, if a transient happens, DBSCAN can detect the drift by partitioning data into more than one cluster of inliers and outliers. In Serradilla et al. (2021), Ordering Points to Identify the Clustering Structure (OPTICS) is explored to visualize the characteristics of the anomalous samples previously found with different one-class classification techniques. OPTICS is closely related to DBSCAN but offers a graphic way of visualized core points of each cluster and the outliers ordering by their reachability from the clusters.

Isolation forests

Isolation Forest is an unsupervised algorithm for outlier detection inspired by the DT ensemble method RF (see Section 4.4.1). Thus, it is based on combining multiple DTs that isolate patterns in each division, so that patterns with more isolated behaviors are isolated as few divisions of the tree. Aremu, Hyland-Wood, and McAree (2020) and Kolokas et al. (2020) explored outlier detection over time series that have been preprocessed with different techniques of features reduction and variables selection.

Deep learning

The paradigm of DL has been used for supervised learning tasks in the context of PdM as previously mentioned. However, its nature makes possible to apply in other contexts such as the outlier detection. Thus, the study of Malawade et al. (2021) proposed a model of DL from an online learning perspective. It focuses on efficiently learning from a single training pass to provide real-time early outlier detection over vibration data. The proposal is based on the Hierarchical Temporal Memory (HTM) model, a sequence learning framework inspired by the structure of the neocortex in the human brain. The basic unit of HTM is a pyramidal neuron with feed-forward, feedback, and lateral components to connect to other neurons. These neurons are stacked on top of one another to form a column like the "cortical column" of the neocortex. The final HTM is a composition of many such columns. A neuron becomes active at any time only if it is in the predictive state at the previous instant. The time series is passed through a spatial pooler and a temporal pooler before outputting a prediction for the next set of column activations. The prediction and the historical distribution of anomaly scores are used to determine the anomaly likelihood at each time.

4.4.4 | Semi-supervised learning in PdM

Semi-supervised learning is halfway between supervised and unsupervised learning, characterized by having a small amount of labeled and mostly unlabeled data. This approach can be very useful in a data streaming scenario like PdM, where often there is a large amount of data without labeling entirely. In this context, generative classification can leverage both labeled and unlabeled data in a more beneficial way than using only supervised or unsupervised learning.

Thus, the aims of generative classification are (1) to learn from labeled training data and to generate labels for available unlabeled data and (2) to generalize to new data. Semi-supervised learning is also helpful when new situations that they have not been seen previously, like new system malfunctions. This approach is related to the DM task of one-class classification. It addresses the problem of identifying the boundaries that define the initial data set to distinguish whether or not the new data have the same characteristics as the initial data. A PdM problem is addressed from a one-class classification perspective when the monitored system has not yet failed. In these situations, all the data used to train the predictive model correspond to normal operation, and the objective is to identify any novelties that may be subjected to study in case they could lead to a system failure.

This section details the different ML techniques that have been proposed in this context.

Support vector machines

One-Class Support Vector Machine (OCSVM) is derived from the SVM algorithm presented in Section 4.4.1, but in this case it does not seek to separate two or more classes, but to create, from the support vectors, a boundary that delimits the distribution of the data in the training data. Thus, if new data lay within the frontier-delimited subspace, they are considered as normal, and if they do not fit the distribution, they are considered anomalies. This way of anomaly detection based on training the model only with normal data is known as novelty detection. The underlying kernel that is unanimously used for OCSVM in the context of novelty detection in PdM is the radial type RBF. The study of Ribeiro et al. (2016) presented an evolving model that apply OCSVM over time series using different sliding window strategies. In a second stage, the detected anomalous points are merged by a low-pass filter for subsequence outlier detection. In Casoli et al. (2019), OCSVM is used as baseline method to detect anomalies. The proposal is based on counting the anomalies during a fixed time window and if this number exceeds a given threshold, an abnormal event alarm is triggered. In Serradilla et al. (2021), OCSVM is studied together with other methods for novelty detection over working cycles. After the phase of novelty detection, other ML methods based on unsupervised learning are employed to explain the characteristics of the anomalies detected. In Oliveira et al. (2020), OCSVM is studied together with other methods based on clustering after signal preprocessing obtained with nondestructive ultrasounds. In Aremu, Hyland-Wood, and McAree (2020) and Aremu, Cody, et al. (2020), OCSVM is applied over time series that have been processed in order to reduce the high dimensionality of the data. In Morariu et al. (2020), a OCSVM model is combined with time forecasts to detect future anomalies in a context of production planning and maintenance scheduling.

Deep learning

RNNs are a type of DL method used for classification and regression tasks mainly, as it has been commented previously in Sections 4.4.1 and 4.4.2. However, due to their suitability for processing sequential data, they have been used for time-series forecasting in a context of novelty detection. Specifically, in Cho et al. (2020), a type of RNN called Echo State Network that has a dynamical memory to preserve in its internal state a nonlinear transformation of the input's history. Thus, this model starts with a fault-free signal that has been preprocessed with double-exponential smoothing filter, tuned with evolutionary algorithms, to generate a smoothed signal. The model is trained with this signal to generate a fault-free lower bound, and the future predictions are compared to this bound in such a way that if they diverge, there is a degradation in the system. Another DL method very popular in novelty detection is AE. It is based on compressing the input into a latent-space representation, and then reconstructing the output from this representation, that is, ideally the input and the output of the model is the same. This property can be used in the context of one-class classification to learn the characteristics of the normal data. Then, in the novelty detection phase, when anomalous sample comes and are processed by the AE, it will not correctly capture all its characteristics and there will be differences between the input and output produced. This will serve to detect the anomaly. Normally, the internal layers of the AE use CNN architectures to compress and expand the signal. This approach has been used for PdM in Ribeiro et al. (2016) and Kim et al. (2021). Time series are processed by means of sliding window strategy and find anomalies using an AE. In a second stage, the detected anomalous points are merged by a low-pass filter for subsequence outlier detection. Fathi et al. (2021) also explored the CNN-based AE to perform novelty detection over the time series, previously fragmented for batch processing. The reconstruction error obtained from the AE is passed to a minimax optimized sigmoid function to infer the health index of the monitored system, that depends on if an anomaly has been detected or not. In Serradilla et al. (2021), an AE is studied together with other methods for novelty detection over working cycles, achieving the best performance. After the phase of novelty detection, other ML methods based on unsupervised learning are employed to explain the characteristics of the anomalies detected. Other approach of AE designed to deal with time series by means of LSTM in its internal layers is explored in several works (Bouabdallaoui et al., 2021; Fernández-

Barrero et al., 2021; Ning et al., 2021). The anomalies are detected from the differences between time series at the input and at the output. In the case of Basora et al. (2021), three AE are compared to detect anomalies in complex multivariate time series: a fully connected AE, a CNN-AE, and a LSTM-AE. Finally, a health indicator is computed from the anomaly score. The concept of reconstruction error for anomaly detection is extended to more DL models in Gribbestad et al. (2021). Thus, several configurations of AEs are compared, along with other architectures like CNNs, LSTMs, and DBNs. All the architectures had the output layer of the same dimension than the input ones among which the variational AE and the LSTM model achieved the best results.

Generative classification for semi-supervised learning is an emerging trend in PdM. Thus, only one work at the end of the reviewed period has been found in this approach. In Zhang et al. (2021), two deep generative models based on variational AEs are proposed to deal with bearing fault classification with small proportion of labeled data. The model is based on the simultaneous training of two models: a variational AE for unlabeled data, and a variational AE-based classifier for labeled data.

Hybrid models

Hybrid models based on analyzing the distribution of the data and then mapping the learned features with desirable properties have been introduced in supervised learning (see Section 4.4.1), but they can be used too from an unsupervised learning perspective, mapping the features to normal or abnormal behavior instead to defined classes. In Yu et al. (2020), PCA is applied to discern which dynamics are more important in the system, which are redundant and which are noise. The results of the PCA projection are then analyzed with two statistics, T-squared and Q-squared, that detect abnormal variations in the subspace of the first principal components and in the residual subspace, respectively. Other PCA method is explored in Serradilla et al. (2021) for anomaly detection. The model focuses on the samples that cannot be explained by the principal components, that is, the anomalies. Once these samples are located, other DM methods based on unsupervised learning are employed to employ the characteristics of the anomalies detected. The study of Oliveira et al. (2020) proposed a distance-based method that computes the baricenter and the maximum dissimilarity of all normal training data based on the Euclidean distance. With these statistics, it defined a boundary that delimits whether new data are in the normality region or, on the contrary, not, that is, they are anomalies.

5 | PdM: USEFUL RESOURCES AVAILABLE

This section aims to provide the necessary tools for the implementation of PdM solutions from scratch. Thus, a list of public datasets related to different industrial domains and focused on the resolution of several tasks related to PdM have been collected. The frameworks available for programming ML algorithms for solving PdM tasks are also provided.

5.1 | Public datasets

The nature of the PdM research field is eminently applied, that is, the papers reviewed usually work on specific problems, sometimes in collaboration with companies, to provide them solutions. Thus, most reviewed papers utilize data that are not publicly available. Even so, there are several public datasets available online and related with PdM from different perspectives. Tables 7 and 8 show these datasets attending to their name, the physical domain where they has been used, a general description, the specific PdM problem, data types, the reference to its web page or reference, and the list of references that use it.

The NASA Ames Prognostics Data Repository groups are the most datasets related with PdM (Bole et al., 2014; Celaya et al., 2011; Chao et al., 2021; Lee et al., 2007; Nectoux, Gouriveau, Medjaher, Ramasso, Morello, et al., 2012; Saha & Goebel, 2007; Saxena & Goebel, 2008). The most popular dataset used in PdM during the reviewed period is also from this repository, related to turbofan engine (Saxena & Goebel, 2008). Another common domain is the ball bearing, with the Case Western Reserve University bearing dataset (*Bearing Dataset*, CSE Groups, 2008) as the most popular dataset. Component surface defect dataset (Schlagenhauf & Landwehr, 2021) and Alarms logs dataset (Tosato et al., 2020) have been generated from PdM article included in this review (Pezze et al., 2021; Schlagenhauf & Burghardt, 2021). Some of the datasets listed in Tables 7 and 8 do not report usage in the period reviewed, which is partly because they are too modern or offer similar features to more popular ones.

TABLE 7 Public dataset for PdM I

Name	Domain	Description	Target	Data type	Access	References
CMAPSSD	Turbofan	NASA turbofan engine degradation data set	RUL	Multivariate time series	Saxena and Goebel (2008)	Zhang, Wang, et al. (2018), Gohel et al. (2020), Demidova (2020), Nguyen and Medjaher (2019), Aremu, Hyland-Wood, and McAree (2020), Aremu, Cody, et al. (2020), Saranya and Sivakumar (2020), Mode et al. (2020), J. Li and He (2020), Kang et al. (2021), Chen, Zhu, et al. (2021), Pillai and Vadakkepat (2021), Benker et al. (2021), Xiong et al. (2021), Resende et al. (2021), Bharti and McGibney (2021), Ayodeji et al. (2021), Chui et al. (2021)
Case Western Reserve University Bearing Data Set	Ball bearing	Test rig operated under different load conditions	Health condition	Multivariate time series	Bearing Dataset, CSE Groups (2008)	Zhou et al. (2017), D. Wang et al. (2018), Chen et al. (2019), Yang, Zheng, et al. (2019), Yang, Lei, et al. (2019), Zhang, Li, and Ding (2019), Zhang, Li, Wang, et al. (2019), Qian et al. (2019), Souza et al. (2021), S. Zhang et al. (2021)
Bearing Data Set	Ball bearing	Vibration signal of run-to-failure experiments of bearings in a shaft.	RUL	Multivariate time series	Lee et al. (2007)	Wang and Mamo (2019), Malawade et al. (2021), S. Zhang et al. (2021)
IEEE PHM Challenge	Rolling bearing	Six rolling bearings operated under three different working conditions and 11 more for testing.	RUL	Multivariate time series	Nectoux, Gouriveau, Medjaher, Ramasso, Chebel-Morello, et al. (2012)	Liao et al. (2016), Wu et al. (2018), Zhang, Zhang, and Li (2019), Malawade et al. (2021), Kamat et al. (2021), Gribbestad et al. (2021)
MAFAULDA	Rotating motor	Machinery simulators run under different load conditions	Health condition	Multivariate time series	Ribeiro (2018)	Souza et al. (2021)
APS Failure at Scania Trucks Data Set	Truck air pressure system	Heavy trucks monitorization with a focus on APS system.	APS failure	Multivariate	Lindgren and Biteus (2016)	Oh and Lee (2020)
Bacblaze Hard Drive	Hard disk drive	Daily snapshots of the status of thousands of disks	Disk failure	Multivariate time series	Backblaze.com (2021)	Su and Huang (2018), Kaparthi and Bumblauskas (2020)
FEMTO Bearing Data Set	Ball bearing	Vibration and temperature signals of run-to-failure bearings	RUL	Multivariate time series	Nectoux, Gouriveau, Medjaher, Ramasso, Morello, et al. (2012)	Benker et al. (2021)

TABLE 8 Public dataset for PdM II

Name	Domain	Description	Target	Data type	Access	References
Industrial machine tool component surface defect dataset	Ball screw drive	Images of defects on ball screw drive spindles showing the progression of the defects on surface.	Health condition	Images with temporal order	Schlagenhauf and Landwehr (2021)	Schlagenhauf and Burghardt (2021)
Alarm Logs in Packaging Industry	Packaging machine	Sequence of alarms logged by packaging equipment in an industrial environment.	Alarm forecasting	Alarms logged with associated timestamp and piece of equipment	Tosato et al. (2020)	Pezze et al. (2021)
CMAPSSD-2	Turbofan	Updated version of NASA turbofan engine degradation data set	RUL	Multivariate time series	Chao et al. (2021)	
Battery Data Set	Li-ion battery	NASA battery degradation data set during repeated charge and discharge cycles	RUL/remaining charge	Multivariate time series	Saha and Goebel (2007)	
Randomized Battery Usage Data Set	Li-ion battery	NASA battery degradation data set using a randomized sequence of charging-discharging.	RUL	Multivariate time series	Bole et al. (2014)	
MOSFET Thermal Overstress Aging Data Set	Electrical component	Run-to-failure experiments on Power MOSFETs under thermal overstress.	RUL	Multivariate time series	Celaya et al. (2011)	
Pump data for predictive maintenance	Water pump grid	General grid status measured during a year through sensors.	Health condition/ anomaly detection	Multivariate time series	Kaggle (2019)	

5.2 | Libraries and frameworks

Some of the works reviewed in this article give details of the software they have used to develop their proposal. From these indications, it has been possible to compile the list of libraries and frameworks shown in Tables 9 and 10. Each software includes its main functions as well as a brief description; the interface of use, which is normally a programming language popular in DM methods like Python, although in some cases it is a graphic interface; if it is free software, in which case the license of use is indicated; and, finally, the reviewed works that indicate that they use the software. Most of these libraries are dedicated to the programming of ML techniques from different paradigms. Some of them combine desirable functions in PdM such as data visualization or functions for processing large amounts of data. Other libraries have different functions, such as simulated data generation or message passing between nodes.

6 | DISCUSSION AND FUTURE TRENDS

This section analyzes the current situation, which will allow us to point out future directions in PdM. First, the different DM tasks by year are analyzed. Then, a study of the different paradigms by years is carried out. Finally, considering the latest advances, the future trends are shown.

Figure 5a shows a summary of the annual publications according to DM tasks. The growing trend is appreciated in the number of proposals: three PdM proposals from a DM perspective the years 2015 and 2016, eight proposals during the year 2017, 22 during the year 2018, 32 during the year 2019, 64 for 2020 and 95 during 2021. These results follow the tendency observed during the exploratory analysis confirming that it is crucial to study these last years to get a complete picture of the state of the art in the field. A more detailed study about the different tasks shows that regression and multiclass classification are currently the most popular approaches. There are two remarkably growth. On the one hand, binary classification, there were 15 proposals between 2015 and 2019, and a substantial increase in 2020 (17 proposals) that has been maintained during 2021 (20 proposals). On the other hand, the one-class classification paradigm has also grown significantly in recent years. Thus, if by 2019, three proposals had been published in the field, in 2020, seven publications were reached and in 2021, 13 publications were reached.

According to Figure 5a, we can also extract that the most popular problems are those addressed from a supervised perspective (regression and classification DM tasks). The majority of proposals during the first reviewed years belong to these areas, with the same trend for 2020 and 2021. Specifically, multiclass classification is the most popular technique with 73 proposals reviewed, following by regression with 59 proposals and binary classification with 53. It is also noteworthy that more complex problems emerge during 2019 and continue to grow in next years, such as clustering and novelty detection through one-class classification.

Considering the different DM tasks used to solve PdM problems, an in-depth study is carried out to show the trends in the ML algorithms used. Figure 5b shows an analysis of the publications included in this review that use the regression task for solving PdM problems (normally, RUL problem). The main point to highlight is a shift toward neural networks, a more established technique in classification where it was used previously and more recently extended to regression in PdM. Thus, both traditional and DL models have grown in 2020 and 2021, as opposed to other models with lower performance, such as symbolic regression and SVM models.

Similarly, Figure 5c shows the number of reviewed publications attending to the use of the classification task for solving PdM problems (normally, predicting health status or failure modes). In this case, it can be seen that the trend is, on the one hand, the use of classical models like DT and kNN underscored by obtaining more explainable results that it is highly demanded in recent years. On the other hand, neural networks and more recently DL models maintain their influence over the reviewed period because of their high accuracy. More specifically, classical ANNs have been quite prevalent until 2020, with 11 proposals applied to classification. However, in 2021 their use has dropped to only two proposals. On the other hand DL networks have occupied this ground, increasing their use in 2021 to 14 proposals, while in the whole previous period they have been publishing about five publications per year, considering together CNN, RNN, and ELM.

Finally, Figure 5d shows the number of works that have used clustering or one-class classification tasks for solving PdM problems using unsupervised or semi-supervised learning paradigm, respectively. These tasks have been lesser used being a less mature area of application of PdM. It is remarkable that the first methods that appear (k-Means and OCSVM) to continue being quite relevant in subsequent years. Moreover, new approaches have emerged strongly in recent years. Concretely, DL models based on AE have an astounding growth during 2021 for novelty detection. Finally,

TABLE 9 Libraries and framework for implementing PdM I

Name	Description	Base usage	Open source	References
Scikit-learn	Machine learning library.	Python	New BSD License	Hsu et al. (2020), Kolokas et al. (2020), Saranya and Sivakumar (2020), Pałasz and Przysowa (2019), Ayvaz and Alpay (2021), Pezze et al. (2021)
Tensorflow	Numerical computation library focus on artificial neural networks and deep learning.	Python	Apache License 2.0	Liang et al. (2020), Schlagenhauf and Burghardt (2021), Resende et al. (2021)
Keras	Library for deep learning models over TensorFlow	Python	MIT license	Lepenioti et al. (2020), Morariu et al. (2020), Demidova (2020), Qian et al. (2019), Zhang, Zhang, and Li (2019), Pałasz and Przysowa (2019), Bampoula et al. (2021), Ayvaz and Alpay (2021), Fathi et al. (2021), Ayodeji et al. (2021)
Pytorch	Library for machine learning based on tensor computation accelerated via GPU.	Python	Modified BSD License	Song et al. (2021)
ML Packages for R: caret, ggplot2, cluster	Basic environment for machine learning in R Project	R	GNU GPL	Bekar et al. (2020), de Carvalho Chrysostomo et al. (2020), Zschech et al. (2019), Cakir et al. (2021)
Apache Spark	Framework for large-scale data processing and machine learning support	Java/Python/ Scala	Apache License 2.0	Su and Huang (2018), Zhang, Liu, et al. (2018), Morariu et al. (2020), Proto et al. (2020), Yu et al. (2020), Panicucci et al. (2020), Fernández-Barrero et al. (2021), Cerquitelli et al. (2021)
Apache Flink	Framework for data streaming processing.	Java/Python/ Scala	Apache License 2.0	Axenie et al. (2020)
Apache Kafka	Platform for high-throughput and low-latency communication through stream processing.	Java	Apache License 2.0	Lepenioti et al. (2020)
H2O.ai	Platform for data modeling and general computing in a distributed and parallel manner.	Java/Python	Apache License 2.0	Calabrese et al. (2020)
BURLAP	Library for single and multiagent planning and learning and reinforcement learning	Java	Apache License 2.0	Lepenioti et al. (2020)
KNIME	Analytic platform based on modular data pipeline concept.	Graphical interface	GNU GPL	Orrù et al. (2020)
HeuristicLab	Library for heuristic and evolutionary algorithm development from a graph-based approach	Graphical interface	GNU GPL	Zenisek, Holzinger, and Affenzeller (2019), Zenisek, Kronberger, et al. (2019)

TABLE 10 Libraries and framework for implementing PdM II

Name	Description	Base usage	Open source	References
Predictive Maintenance Toolbox	Matlab toolbox that combines several functions specific to PdM: sensor management, health status definition and analytics.	Matlab	No	Ullah et al. (2017), Srivastava et al. (2018), Li et al. (2019), Sampaio et al. (2019), Zhou et al. (2019), Song et al. (2021)
Vensim	Simulation software for continuous synthetic data generation and discrete event and agent-based modeling capabilities.	Graphical interface	No	Crespo Márquez et al. (2020)
Anomaly Detection Toolbox	Ten popular algorithms for anomaly detection for multivariate data.	Matlab	No	
ELKI	KDD-applications with focus on unsupervised learning, clustering and outlier detection.	Java	AGPLv3	
PyOD	Toolkit for anomaly detection in multivariate data including 30 algorithms and ensembles.	Python	BSD 2	
PySAD	Library for online anomaly detection for streaming data.	Python	BSD 3	
AnomalyDetection6	Package for anomaly detection in a robust statistical manner in time series with seasonality and underlying trend.	R	Yes	
TODS	Library for automated time-series outlier detection including data processing, ML-processing, feature analysis and reinforcement learning.	Python	Apache 2	
DeepADoTS	Library for deep anomaly detection based on PyTorch	Python	MIT	
Merlion	End-to-end ML framework for univariate and multivariate time-series. PythonBSD 3			

with respect to clustering methods, partitional clustering, with K-Means as its maximum exponent, is the method most consistently used throughout the reviewed period, mainly due to its interpretability and speed of response. Thus, there are 12 proposals that use it. On the other hand, density-based methods seem to have some potential in recent years for their ability to detect outliers, with an annual publication since 2018. Finally, hierarchical clustering has hardly been explored.

Analyzing these trends, one can conclude that DL has emerged as an important technique to address PdM problems in either supervised, semi-supervised, or unsupervised learning. This type of models requires a large amount of data and computational capability for training. They are usually black-box models, that is, it is not possible to know which characteristics of the human domain of the problem best explain their outputs. However, they are characterized by their high precision, being, therefore, very relevant today. On the other hand, traditional ML techniques based on DTs and SVM are very relevant as well, mainly in discrete problems such as classification, clustering, or novelty detection. These techniques are easier to train, require less data, and can provide more explainable results. They also obtain reasonably accurate results, depending on the problem and its complexity.

Regarding new challenges, it is observed that in the area of PdM there is still scope to explore several ML techniques with potential in this area of applicability. As noted in the previous analysis, the least explored field is unlabeled or weakly labeled data, which leads to fewer proposals on clustering and novelty detection. In this line, other learning paradigms that deal with the absence of labels could be introduced in PdM. Specifically, reinforcement learning and active learning could be beneficial in contexts where it is not feasible or practical to label large amounts of data, but only the most relevant of both normal operation and faults. In this way, the learning process could be optimized in big data

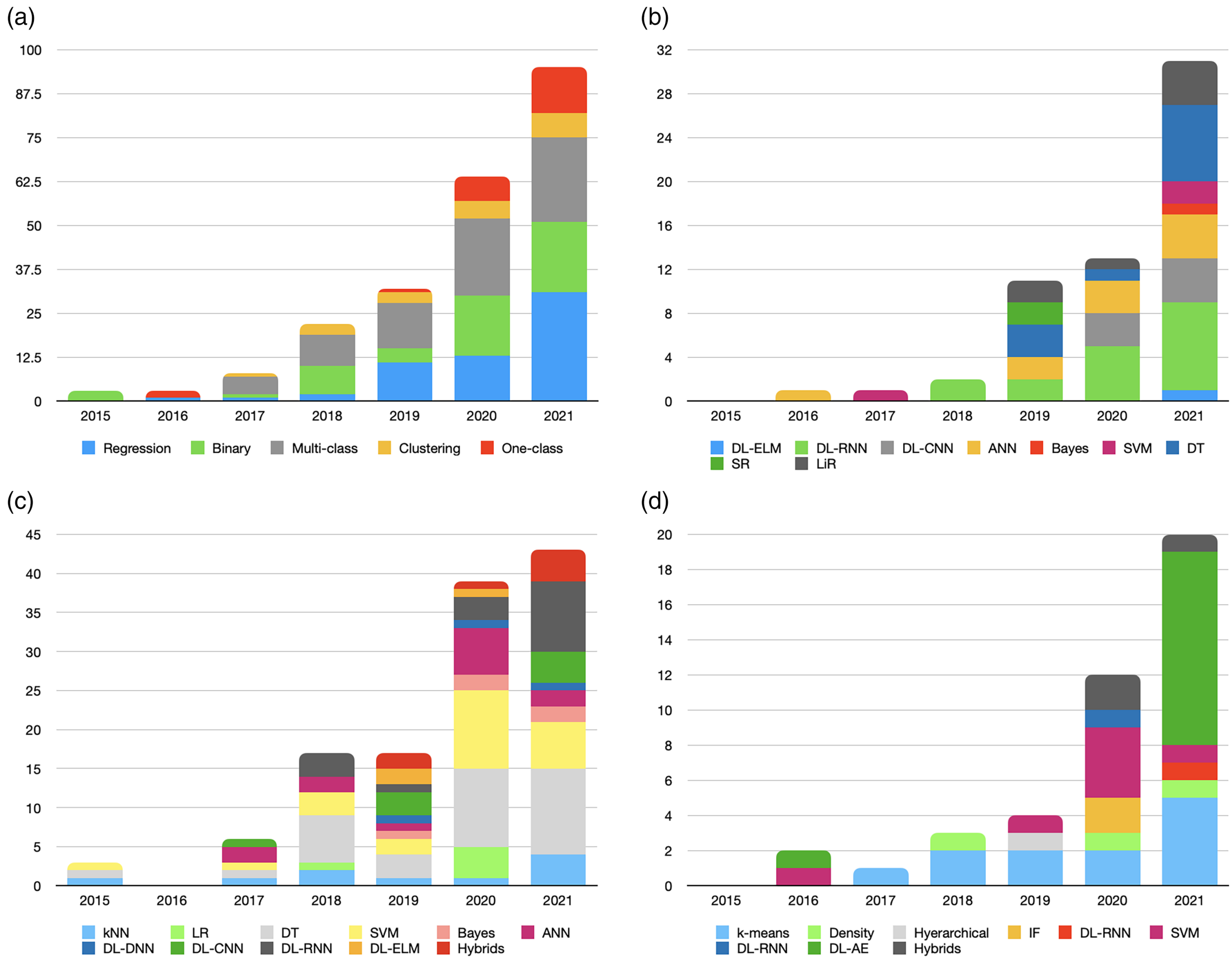

FIGURE 5 Analysis of the number of proposals reviewed. (a) Proposals per year and DM tasks; (b) Proposals per year and regression task; (c) Proposals per year and classification task; (d) Proposals per year and unsupervised or semi-supervised learning

environments or those with few failures. Therefore, little data could be needed to prepare predictive models for these failures.

Finally, looking at flexible information representation techniques, it is observed that no works are using the multi-instance representation paradigm, and there are very few using multilabel. These approaches could be the great support in PdM to work with weakly labeled data or in complex systems with several kinds of failures to detect. Thus, applying these learning paradigms, separately or in combination, could help to improve accuracy and interpretability of prediction models due to more optimized representations. This area will be one of the most important promoted by the next industrial evolution, Industry 5.0. The main objective of Industry 5.0 is to use the creativity of human experts in collaboration with efficient, intelligent, and accurate machines to obtain resource-efficient and user-preferred solutions. Thus, interpretability will be a crucial factor.

7 | CONCLUSIONS

This work presents a novel systematic review of the latest trends in PdM from a DM perspective. The review includes 132 articles published in high-impact journals between the period from 2015 to 2021. Those works have been analyzed exhaustively, leading to a taxonomy that presents three main steps concerning DM to implement a complete PdM solution: data acquisition, data preparation, and the building of the predictive model. In addition, a complete list of open and public datasets has been compiled, as well as libraries and frameworks that can be used to implement PdM solutions.

During the realization of this review, it has been possible to verify that PdM is an emerging and very active field, with many publications that grow exponentially thanks to the advances in monitoring brought about by the Industry 4.0 paradigm, and advances in predictive models and computing capacity. Thus, it has established several industrial problems to address from PdM, which are determined by the type of data available: whether data are labeled, whether they contain failure events, or whether they are continuous or categorical to determine the possibilities of the PdM solutions to implement, as well as the DM task and the ML paradigm to implement. Once the targets of the PdM, the DM task and ML paradigm, have been established, there is a range of ML techniques that can be used to implement the final PdM solution. The reviewed period for this work, 2015–2021, offers a current and complete overview of the latest trends in PdM, where knowledge is more mature, and proposals are becoming more complex than in the previous decade. It is observed that the field is currently at a turning point in which, on the one hand, there are works based on classic ML and, on the other hand, there are many other works based on DL. Both proposals can be found in all tasks and problems studies. Although DL can achieve more accurate results, the transparency, and interpretability of the obtained results by other classic techniques is also important. Most DL-based works do not use data preprocessing. However, in articles based on classical ML, preprocessing is often used, especially techniques that project attributes into another dimensional space to reduce the volume of data to be processed.

Based on the results obtained, current trends have been analyzed and future challenges have been presented. Thus, DL, among other paradigms, is having a high impact in the last publications because of its high precision. However, companies need to rely on the predictions of the systems. In this line, one can find new trends about interpretability and explainability. These advances bring end-users closer to the DM model, as well as flexible representations focus on the most relevant data to simplify model building. It can be concluded that in the area of PdM there is still scope for exploring several DM techniques with potential in the optimization of the available information. Thus, semi-supervised learning paradigms, able to deal with unbalanced classes or large amounts of unlabeled data, are a growing field in PdM. On the other hand, flexible information representation techniques, like the multi-instance and multilabel paradigms, can be useful in processing a large amount of data efficiently. Moreover, these paradigms obtain simpler, and therefore, more interpretable models for the explanation of predicted future failures.

AUTHOR CONTRIBUTIONS

Aurora Esteban: Investigation (equal); visualization (equal); writing – original draft (equal). **Amelia Zafra:** Methodology (equal); supervision (equal); writing – original draft (equal); writing – review and editing (equal). **Sebastián Ventura:** Conceptualization (lead); funding acquisition (lead); methodology (lead); resources (lead); supervision (lead); writing – original draft (equal); writing – review and editing (equal).

ACKNOWLEDGMENTS

This research was supported by the Spanish Ministry of Science and Innovation and the European Regional Development Fund, project PID2020-115832GB-I00, and by the Spanish Ministry of Science and Innovation, doctoral grant FPU19/03924. The authors are also grateful to the University of Córdoba, Spain/CBUA for funding open access publication charges.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

DATA AVAILABILITY STATEMENT

Data sharing does not apply to this article as no new data were created or analyzed in this study.

ORCID

Aurora Esteban https://orcid.org/0000-0002-6290-5353
Amelia Zafra https://orcid.org/0000-0003-3868-6143
Sebastián Ventura https://orcid.org/0000-0003-4216-6378

RELATED WIREs ARTICLES

An overview on the evolution and adoption of deep learning applications used in the industry

REFERENCES

Acernese, A., Del Vecchio, C., Tipaldi, M., Battilani, N., & Glielmo, L. (2020). Condition-based maintenance: An industrial application on rotary machines. *Journal of Quality in Maintenance Engineering*, *27*(4), 565–585.

Angelopoulos, A., Michailidis, E. T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S., & Zahariadis, T. (2020). Tackling faults in the industry 4.0 era—A survey of machine-learning solutions and key aspects. *Sensors (Switzerland)*, *20*(1), 1–34.

Aqueveque, P., Radrigan, L., Pastene, F., Morales, A. S., & Guerra, E. (2021). Data-driven condition monitoring of mining mobile machinery in non-stationary operations using wireless accelerometer sensor modules. *IEEE Access*, *9*, 17365–17381.

Aremu, O. O., Cody, R. A., Hyland-Wood, D., & McAree, P. R. (2020). A relative entropy based feature selection framework for asset data in predictive maintenance. *Computers and Industrial Engineering*, *145*(May), 106536.

Aremu, O. O. O., Hyland-Wood, D., & McAree, P. R. P. (2020). A machine learning approach to circumventing the curse of dimensionality in discontinuous time series machine data. *Reliability Engineering and System Safety*, *195*, 106706.

Axenie, C., Tudoran, R., Bortoli, S., Hassan, M. A. H., Wieder, A., & Brasche, G. (2020). SPICE: Streaming PCA fault identification and classification engine in predictive maintenance. *Communications in Computer and Information Science*, *1168*, 333–344.

Ayodeji, A., Wang, Z., Wang, W., Qin, W., Yang, C., Xu, S., & Liu, X. (2021). Causal augmented ConvNet: A temporal memory dilated convolution model for long-sequence time series prediction. *ISA Transactions*, *123*, 1–18.

Ayvaz, S., & Alpay, K. (2021). Predictive maintenance systemfor production lines in manufacturing: Amachine learning approach using IoT data in real-time. *Expert Systems with Applications*, *173*(January), 1–10.

Azab, E., Nafea, M., Shihata, L. A., & Mashaly, M. (2021). A machine-learning-assisted simulation approach for incorporating predictive maintenance in dynamic flow-shop scheduling. *Applied Sciences (Switzerland)*, *11*(24), 1–17.

Backblaze.com. (2021). *Hard drive dataset*. https://www.backblaze.com/b./hard-drive-test-data.html

Bampoula, X., Siaterlis, G., Nikolakis, N., & Alexopoulos, K. (2021). A deep learning model for predictive maintenance in cyber-physical production systems using LSTM autoencoders. *Sensors (Switzerland)*, *21*(3), 1–14.

Baptista, M. L., Henriques, E. M., & Prendinger, H. (2021). Classification prognostics approaches in aviation. *Measurement*, *182*(April), 1–10.

Basora, L., Bry, P., Olive, X., & Freeman, F. (2021). Aircraft fleet health monitoring with anomaly detection techniques. *Aerospace*, *8*(4), 1–33.

Bekar, E. E. T., Nyqvist, P., & Skoogh, A. (2020). An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study. *Advances in Mechanical Engineering*, *12*(5), 1–14.

Benker, M., Bliznyuk, A., & Zaeh, M. F. (2021). A Gaussian process based method for data-efficient remaining useful life estimation. *IEEE Access*, *9*, 137470–137482.

Berghout, T., Mouss, L. H., Bentrcia, T., Elbouchikhi, E., & Benbouzid, M. (2021). A deep supervised learning approach for condition-based maintenance of naval propulsion systems. *Ocean Engineering*, *221*, 1–11.

Bharti, S., & McGibney, A. (2021). Privacy-aware resource sharing in cross-device federated model training for collaborative predictive maintenance. *IEEE Access*, *9*, 120367–120379.

Bole, B., Kulkarni, C., & Daigle, M. (2014). *Randomized battery usage data set*. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#batteryrnddischarge

Bouabdallaoui, Y., Lafhaj, Z., Yim, P., Ducoulombier, L., & Bennadji, B. (2021). Predictive maintenance in building facilities: A machine learning-based approach. *Sensors (Switzerland)*, *21*(4), 1–15.

Burkov, A. (2019). *The hundred-page machine learning book*. Andriy Burkov.

Cakir, M., Guvenc, M. A., & Mistikoglu, S. (2021). The experimental application of popularmachine learning algorithms on predictive maintenance and the design of IIoT based condition monitoring system. *Computers and Industrial Engineering*, *151*, 1–14.

Calabrese, M., Cimmino, M., Fiume, F., Manfrin, M., Romeo, L., Ceccacci, S., Paolanti, M., Toscano, G., Ciandrini, G., Carrotta, A., Mengoni, M., Frontoni, E., & Kapetis, D. (2020). SOPHIA: An event-based IoT and machine learning architecture for predictive maintenance in industry 4.0. *Information (Switzerland)*, *11*(4), 202.

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. *Computers and Industrial Engineering*, *137*(August), 1–10.

Casoli, P., Pastori, M., Scolari, F., & Rundo, M. (2019). A vibration signal-based method for fault identification and classification in hydraulic axial piston pumps. *Energies*, *12*(5), 1–18.

Celaya, J. R., Saxena, A., Saha, S., & Goebel, K. (2011). *MOSFET thermal overstress aging data set*. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#mosfet

Cerquitelli, T., Ventura, F., Apiletti, D., Baralis, E., Macii, E., & Poncino, M. (2021). Enhancing manufacturing intelligence through an unsupervised data-driven methodology for cyclic industrial processes. *Expert Systems with Applications*, *182*(March), 1–10.

Chang, R. I., Lee, C. Y., & Hung, Y. H. (2021). Cloud-based analytics module for predictive maintenance of the textile manufacturing process. *Applied Sciences (Switzerland)*, *11*(21), 1–22.

Chao, M., Kulkarni, C., Goebel, K., & Fink, O. (2021). *Aircraft engine run-to-failure dataset under real flight conditions*. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#turbofan-.

Chen, C., Liu, Y., Sun, X., Cairano-Gilfedder, C. D., & Titmus, S. (2021). An integrated deep learning-based approach for automobile maintenance prediction with GIS data. *Reliability Engineering and System Safety*, *216*(January), 1–16.

Chen, C., Liu, Y., Wang, S., Sun, X., Di Cairano-Gilfedder, C., Titmus, S., & Syntetos, A. A. (2020). Predictive maintenance using cox proportional hazard deep learning. *Advanced Engineering Informatics*, *44*, 101054.

Chen, C., Zhu, Z. H., Shi, J., Lu, N., & Jiang, B. (2021). Dynamic predictive maintenance scheduling using deep learning ensemble for system health prognostics. *IEEE Sensors Journal*, *21*(23), 26878–26891.

Chen, H., Hsu, J. Y., Hsieh, J. Y., Hsu, H. Y., Chang, C. H., & Lin, Y. J. (2021). Predictive maintenance of abnormal wind turbine events by using machine learning based on condition monitoring for anomaly detection. *Journal of Mechanical Science and Technology*, *35*(12), 5323–5333.

Chen, Z., Gryllias, K., & Li, W. (2019). Mechanical fault diagnosis using convolutional neural networks and extreme learning machine. *Mechanical Systems and Signal Processing*, *133*, 106272.

Cheng, J. C., Chen, W., Chen, K., & Wang, Q. (2020). Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms. *Automation in Construction*, *112*, 103087.

Cho, A. D., Carrasco, R. A., Ruz, G. A., & Ortiz, J. L. (2020). Slow degradation fault detection in a harsh environment. *IEEE Access*, *8*, 175904–175920.

Chui, K. T., Gupta, B. B., & Vasant, P. (2021). A genetic algorithm optimized RNN-LSTM model for remaining useful life prediction of turbofan engine. *Electronics (Switzerland)*, *10*(3), 1–15.

Çinar, Z. M., Nuhu, A. A., Zeeshan, Q., Korhan, O., Asmael, M., & Safaei, B. (2020). Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability (Switzerland)*, *12*(19), 1–42.

Consilvio, A., Solís-Hernández, J., Jiménez-Redondo, N., Sanetti, P., Papa, F., & Mingolarra-Garaizar, I. (2020). On applying machine learning and simulative approaches to railway asset management: The earthworks and track circuits case studies. *Sustainability (Switzerland)*, *12*(6), 1–24.

Crespo Márquez, A., Crespo Del Castillo, A., & Gómez Fernández, J. F. (2020). Integrating artificial intelligent tech niques and continuous time simulation modelling. Practical predictive analytics for energy effciency and failure detection. *Computers in Industry*, *115*, 103164.

CSE Groups. (2008). *Bearing dataset*. https://csegroups.case.edu/bearingdatacenter/home

Cupek, R., Ziebinski, A., Zonenberg, D., & Drewniak, M. (2018). Determination of the machine energy consumption profiles in the mass-customised manufacturing. *International Journal of Computer Integrated Manufacturing*, *31*(6), 537–561.

Dangut, M. D., Skaf, Z., & Jennions, I. K. (2021). An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset. *ISA Transactions*, *113*, 127–139.

de Carvalho Chrysostomo, G. G., de Aguiar Vallim, M. V. B., da Silva, L. S., Silva, L. A., & de Aguiar Vallim Filho, A. R. (2020). A framework for big data analytical process and mapping—BAProM: Description of an application in an industrial environment. *Energies*, *13*(22), 1–28.

Demidova, L. A. (2020). *Recurrent neural Networks' configurations in the predictive maintenance problems* [Conference presentation]. Iop Conference Series: Materials Science and Engineering, Moscow, Russia, 1–9.

Fathi, K., van de Venn, H. W., & Honegger, M. (2021). Predictive maintenance: An autoencoder anomaly-based approach for a 3 dof delta robot. *Sensors*, *21*(21), 1–23.

Fernandes, S., Antunes, M., Santiago, A. R. A., Barraca, J. J. P., Gomes, D., & Aguiar, R. R. L. (2020). Forecasting appliances failures: A machine-learning approach to predictive maintenance. *Information (Switzerland)*, *11*(4), 1–13.

Fernández-Barrero, D., Fontenla-Romero, O., Lamas-López, F., Novoa-Paradela, D., R-Moreno, M. D., & Sanz, D. (2021). Soprene: Assessment of the spanish armada's predictive maintenance tool for naval assets. *Applied Sciences (Switzerland)*, *11*(16), 1–18.

Giordano, D., Mellia, M., & Cerquitelli, T. (2021). K-mdtsc: K-multi-dimensional time-series clustering algorithm. *Electronics (Switzerland)*, *10*(10), 1–21.

Gohel, H. A., Upadhyay, H., Lagos, L., Cooper, K., & Sanzetenea, A. (2020). Predictive maintenance architecture development for nuclear infrastructure using machine learning. *Nuclear Engineering and Technology*, *52*(7), 1436–1442.

Gribbestad, M., Hassan, M. U., Hameed, I. A., & Sundli, K. (2021). Health monitoring of air compressors using reconstruction-based deep learning for anomaly detection with increased transparency. *Entropy*, *23*(1), 1–27.

Hanga, K. M., & Kovalchuk, Y. (2019). Machine learning and multi-agent systems in oil and gas industry applications: A survey. *Computer Science Review*, *34*, 1–9.

Hashemian, H. M., & Bean, W. C. (2011). State-of-the-art predictive maintenance techniques. *IEEE Transactions on Instrumentation and Measurement*, *60*(10), 3480–3492.

Hoffmann, M. W., Wildermuth, S., Gitzel, R., Boyaci, A., Gebhardt, J., Kaul, H., Amihai, I., Forg, B., Suriyah, M., Leibfried, T., Stich, V., Hicking, J., Bremer, M., Kaminski, L., Beverungen, D., Zur Heiden, P., & Tornede, T. (2020). Integration of novel sensors and machine learning for predictive maintenance in medium voltage switchgear to enable the energy and mobility revolutions. *Sensors (Switzerland)*, *20*(7), 1–24.

Hsu, J. H. Y. J. Y., Wang, Y. F., Lin, K. C., Chen, M. Y., & Hsu, J. H. Y. J. Y. (2020). Wind turbine fault diagnosis and predictive maintenance through statistical process control andmachine learning. *IEEE Access*, *8*, 23427–23439.

Hu, Q., Ohata, E. F., Silva, F. H., Ramalho, G. L., Han, T., & Rebouças Filho, P. P. (2020). A new online approach for classification of pumps vibration patterns based on intelligent IoT system. *Measurement: Journal of the International Measurement Confederation*, *151*, 107138.

Kaggle. (2019). *Pump data for predictive maintenance*. https://www.kaggle.com/nphantawee/pump-sensor-data?select=sensor.csv

Kamat, P. V., Sugandhi, R., & Kumar, S. (2021). Deep learning-based anomaly-onset aware remaining useful life estimation of bearings. *PeerJ Computer Science*, *7*, 1–28.

Kang, Z., Catal, C., & Tekinerdogan, B. (2021). Remaining useful life (Rul) prediction of equipment in production lines using artificial neural networks. *Sensors (Switzerland)*, *21*(3), 1–20.

Kaparthi, S., & Bumblauskas, D. (2020, feb). Designing predictive maintenance systems using decision tree-based machine learning techniques. *International Journal of Quality and Reliability Management*, *37*(4), 659–686.

Karimzadeh, A., & Shoghli, O. (2020). Predictive analytics for roadway maintenance: A review of current models, challenges, and opportunities. *Civil Engineering Journal*, *6*(3), 602–625.

Keartland, S., & Van Zyl, T. (2020). *Automating predictive maintenance using oil analysis and machine learning* [Conference presentation]. 2020 International Saupec/Robmech/Prasa Conference, Saupec/Robmech/Prasa 2020, Cape Town, South Africa, 1–6.

Khodabakhsh, A., Ari, I., Bakir, M., & Ercan, A. O. (2018). Multivariate sensor data analysis for oil refineries and multi-mode identification of system behavior in real-time. *IEEE Access*, *6*, 63489–64405.

Khorsheed, R. M., & Beyca, O. F. (2021). An integrated machine learning: Utility theory framework for real-time predictive maintenance in pumping systems. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *235*(5), 1–15.

Kim, D., Lee, S., & Kim, D. (2021). An applicable predictive maintenance framework for the absence of run-to-failure data. *Applied Sciences (Switzerland)*, *11*(11), 1–17.

Koca, O., Kaymakci, O., & Mercimek, M. (2020). Advanced predictive maintenance with machine learning failure estimation in industrial packaging robots [Conference presentation]. 2020 15th International Conference Development and Application Systems (DAS 2020) - Proceedings, Suceava, Romania, 1–6.

Kolokas, N., Vafeiadis, T., Ioannidis, D., & Tzovaras, D. (2020). A generic fault prognostics algorithm for manu facturing industries using unsupervised machine learning classifiers. *Simulation Modelling Practice and Theory*, *103*(May), 102109.

Lasisi, A., & Attoh-Okine, N. (2018). Principal components analysis and track quality index: A machine learning approach. *Transportation Research Part C: Emerging Technologies*, *91*, 230–248.

Leahy, K., Gallagher, C., O'Donovan, P., Bruton, K., & O'Sullivan, D. T. (2018). A robust prescriptive framework and performance metric for diagnosing and predicting wind turbine faults based on SCADA and alarms data with case study. *Energies*, *11*(7), 1–21.

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

Lepenioti, K., Pertselakis, M., Bousdekis, A., Louca, A., Lampathaki, F., Apostolou, D., Mentzas, D., & Anastasiou, S. (2020). Machine learning for predictive and prescriptive analytics of operational data in smart manufacturing. In *Lecture notes in business information processing* (Vol. 382, pp. 5–16). LNBIP.

Li, J., & He, D. (2020). A Bayesian optimization AdaBN-DCNN method with self-optimized structure and Hyperpa rameters for domain adaptation remaining useful life prediction. *IEEE Access*, *8*, 41482–41501.

Li, K., Xiong, M., Li, F., Su, L., & Wu, J. (2019). A novel fault diagnosis algorithm for rotating machinery based on a sparsity and neighborhood preserving deep extreme learning machine. *Neurocomputing*, *350*, 261–270.

Li, Z., Wang, Y., & Wang, K. S. (2017). Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario. *Advances in Manufacturing*, *5*(4), 377–387.

Liang, Y., Li, B., & Jiao, B. (2020). A deep learning method for motor fault diagnosis based on a capsule network with gate-structure dilated convolutions. *Neural Computing and Applications*, 33(5), 1401–1418.

Liao, L., Jin, W., & Pavel, R. (2016). Enhanced restricted Boltzmann machine with prognosability regularization for prognostics and health assessment. *IEEE Transactions on Industrial Electronics*, *63*(11), 7076–7083.

Lindgren, T., & Biteus, J. (2016). *APS failure at Scania trucks data set*. https://archive.ics.uci.edu/ml/datasets/APS+Failure+at+Scania+Trucks

Liu, C., Tang, D., Zhu, H., & Nie, Q. (2021). A novel predictive maintenance method based on deep adversarial learning in the intelligent manufacturing system. *IEEE Access*, *9*, 49557–49575.

Malawade, A. V., Costa, N. D., Muthirayan, D., Khargonekar, P. P., & Al Faruque, M. A. (2021). Neuroscience-inspired algorithms for the predictive maintenance of manufacturing systems. *IEEE Transactions on Industrial Informatics*, *17*(12), 7980–7990.

Merkt, O. (2020). Predictive models for maintenance optimization: An analytical literature survey of industrial maintenance strategies. In E. Ziemba (Ed.), *Information Technology for Management: Current Research and Future Directions. AITM ISM 2019 Lecture notes in business information processing* (Vol. 380, pp. 135–154). Springer.

Mishra, K. M., & Huhtala, K. (2019). Elevator fault detection using profile extraction and deep autoencoder feature extraction for acceleration and magnetic signals. *Applied Sciences (Switzerland)*, *9*, 1–15.

Mobley, R. K. (2002). *An introduction to predictive maintenance* (2nd ed.). Elsevier.

Mode, G., Calyam, P., & Hoque, K. (2020). Impact of false data injection attacks on deep learning enabled predictive analytics [Conference presentation]. Proceedings of IEEE/IFIP Network Operations and Management Symposium 2020: Management in the Age of Softwarization and Artificial Intelligence, NOMS 2020, Budapest, Hungary, 1–7.

Morariu, C., Morariu, O., Raileanu, S., & Borangiu, T. (2020, sep). Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems. *Computers in Industry*, *120*, 103244.

Namuduri, S., Narayanan, B. N., Davuluru, V. S. P., Burton, L., & Bhansali, S. (2020). Review—Deep learning methods for sensor based predictive maintenance and future perspectives for electrochemical sensors. *Journal of the Electrochemical Society*, *167*, 1–12.

Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., & Varnier, C. (2012). *PRONOSTIA: An experimental platform for bearings accelerated degradation tests* [Conference presentation]. IEEE International Conference on Prognostics and Health Management, PHM'12, Denver, CO, 1–8.

Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Morello, B., Zerhouni, N., & Varnier, C. (2012). *FEMTO bearing data set*. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#femto

Nguyen, K. T., & Medjaher, K. (2019, aug). A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliability Engineering and System Safety*, *188*(February), 251–262.

Ning, S., Sun, J., Liu, C., & Yi, Y. (2021). Applications of deep learning in big data analytics for aircraft complex system anomaly detection. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, *235*(5), 923–940.

Oh, E., & Lee, H. (2020, apr). An imbalanced data handling framework for industrial big data using a gaussian process regression-based generative adversarial network. *Symmetry*, *12*(4), 1–19.

Olesen, J. F., & Shaker, H. R. (2020). Predictive maintenance for pump systems and thermal power plants: State-of-the-art review, trends and challenges. *Sensors (Switzerland)*, *20*(8), 1–24.

Oliveira, M. A., Simas Filho, E. F., Albuquerque, M. C., Santos, Y. T., da Silva, I. C., & Farias, C. T. (2020). Ultrasound based identification of damage inwind turbine blades using novelty detection. *Ultrasonics*, *108*(March), 106166.

Oluwasegun, A., & Jung, J. C. (2020, oct). The application of machine learning for the prognostics and health management of control element drive system. *Nuclear Engineering and Technology*, *52*(10), 2262–2273.

Orrù, P. F., Zoccheddu, A., Sassu, L., Mattia, C., Cozza, R., & Arena, S. (2020). Machine learning approach using MLP and SVM algorithms for the fault prediction of a centrifugal pump in the oil and gas industry. *Sustainability (Switzerland)*, *12*(11), 1–15.

Pałasz, P., & Przysowa, R. (2019, sep). Using different ML algorithms and hyperparameter optimization to predict heat meters' failures. *Applied Sciences (Switzerland)*, *9*, 1–18.

Pang, S., Yang, X., Zhang, X., & Lin, X. (2020). Fault diagnosis of rotating machinery with ensemble kernel extreme learning machine based on fused multi-domain features. *ISA Transactions*, *98*, 320–337.

Panicucci, S., Nikolakis, N., Cerquitelli, T., Ventura, F., Proto, S., Macii, E., Makris, S., Bowden, D., Becker, P., O'Mahony, N., Morabito, L., Napione, C., Marguglio, A., Coppo, G., & Andolina, S. (2020). A cloud-to-edge approach to support predictive analytics in robotics industry. *Electronics (Switzerland)*, *9*(3), 1–22.

Patil, S. D., Mitra, A., Tuggali Katarikonda, K., & Wansink, J. D. (2021). Predictive asset availability optimization for underground trucks and loaders in the mining industry. *Opsearch*, *58*(3), 751–772.

Pezze, D. D., Masiero, C., Tosato, D., Beghi, A., & Susto, G. A. (2021). FORMULA: A deep learning approach for rare alarms predictions in industrial equipment. *IEEE Transactions on Automation Science and Engineering*, 1–12.

Pillai, S., & Vadakkepat, P. (2021). Two stage deep learning for prognostics using multi-loss encoder and convolutional composite features. *Expert Systems with Applications*, *171*(January), 114569.

Proto, S., Di Corso, E., Apiletti, D., Cagliero, L., Cerquitelli, T., Malnati, G., & Mazzucchi, D. (2020). REDTag: A predictive maintenance framework for parcel delivery services. *IEEE Access*, *8*, 14953–14964.

Prytz, R., Nowaczyk, S., Rögnvaldsson, T., & Byttner, S. (2015). Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. *Engineering Applications of Artificial Intelligence*, *41*, 139–150.

Qian, W., Li, S., Yi, P., & Zhang, K. (2019). A novel transfer learning method for robust fault diagnosis of rotating machines under variable working conditions. *Measurement: Journal of the International Measurement Confederation*, *138*, 514–525.

Quatrini, E., Costantino, F., Pocci, C., & Tronci, M. (2020). Predictive model for the degradation state of a hydraulic system with dimensionality reduction. In *International conference on industry 4.0 and smart manufacturing (ism 2019)* (Vol. 42, pp. 516–523). Elsevier B.V.

Rafique, D., Szyrkowiec, T., Grießer, H., Autenrieth, A., & Elbers, J. P. (2018). Cognitive assurance architecture for optical network fault management. *Journal of Lightwave Technology*, *36*(7), 1443–1450.

Resende, C., Folgado, D., Oliveira, J., Franco, B., Moreira, W., Oliveira, A., Cavaleiro, A., & Carvalho, R. (2021). TIP4.0: Industrial internet of things platform for predictive maintenance. *Sensors*, *21*, 1–23.

Ribeiro, F. M. L. (2018). *MAFAULDA: Machinery fault database*. http://www02.smt.ufrj.br/{~}offshore/mfs/page{_}01.html

Ribeiro, R. P., Pereira, P., & Gama, J. (2016). Sequential anomalies: A study in the railway industry. *Machine Learning*, *105*(1), 127–153.

Ruiz-Sarmiento, J. R., Monroy, J., Moreno, F. A., Galindo, C., Bonelo, J. M., & Gonzalez-Jimenez, J. (2020). A predictive model for the maintenance of industrial machinery in the context of industry 4.0. *Engineering Applications of Artificial Intelligence*, *87*, 1–15.

Saha, B., & Goebel, K. (2007). *Battery data set*. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#battery

Sampaio, G. S., Filho, A. R. d. A. V., da Silva, L. S. L. A., & da Silva, L. S. L. A. (2019, oct). Prediction of motor failure time using an artificial neural network. *Sensors (Switzerland)*, *19*(19), 1–17.

Santolamazza, A., Dadi, D., & Introna, V. (2021). A data-mining approach for wind turbine fault detection based on scada data analysis using artificial neural networks. *Energies*, *14*(7), 1–25.

Saranya, E., & Sivakumar, P. B. (2020). *Data-driven prognostics for run-to-failure data employing machine learning models* [Conference presentation]. Proceedings of the 5th International Conference on Inventive Computation Technologies, ICICT 2020, Coimbatore, India, 528–533.

Satishkumar, R., & Sugumaran, V. (2017). Remaining life time prediction of bearings using K-star algorithm–A statistical approach. *Journal of Engineering Science and Technology*, *12*(1), 168–181.

Savitha, R., Ambikapathi, A. M., & Rajaraman, K. (2020). Online RBM: Growing restricted Boltzmann machine on the fly for unsupervised representation. *Applied Soft Computing Journal*, *92*, 106278.

Saxena, A., & Goebel, K. (2008). *Turbofan engine degradation simulation data set*. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository#turbofan

Schlagenhauf, T., & Burghardt, N. (2021). Intelligent vision based wear forecasting on surfaces of machine tool elements. *SN Applied Sciences*, *3*(12), 1–13.

Schlagenhauf, T., & Landwehr, M. (2021). Industrial machine tool component surface defect dataset. *Data in Brief*, *39*, 107643.

Serradilla, O., Zugasti, E., de Okariz, J. R., Rodriguez, J., & Zurutuza, U. (2021). Adaptable and explainable predictive maintenance: Semi-supervised deep learning for anomaly detection and diagnosis in pressmachine data. *Applied Sciences (Switzerland)*, *11*(16), 1–20.

Shafi, U., Safi, A., Shahid, A. R., Ziauddin, S., & Saleem, M. Q. (2018). Vehicle remote health monitoring and prognostic maintenance system. *Journal of Advanced Transportation*, *2018*, 1–10.

Shamayleh, A., Awad, M., & Farhat, J. (2020). IoT based predictive maintenance management of medical equipment. *Journal of Medical Systems*, *44*(4), 72.

Song, L., Wang, L., Wu, J., Liang, J., & Liu, Z. (2021). Integrating physics and data driven cyber-physical system for condition monitoring of critical transmission components in smart production line. *Applied Sciences*, *11*(19), 1–20.

Souza, R. M., Nascimento, E. G., Miranda, U. A., Silva, W. J., & Lepikson, H. A. (2021). Deep learning for diagnosis and classification of faults in industrial rotating machinery. *Computers and Industrial Engineering*, *153*, 1–12.

Srivastava, S., Vaddadi, S., Kumar, P., & Sadistap, S. (2018). Design and development of reverse osmosis (RO) plant status monitoring system for early fault prediction and predictive maintenance. *Applied Water Science*, *8*(6), 1–10.

Steurtewagen, B., & Van den Poel, D. (2021). Adding interpretability to predictive maintenance by machine learning on sensor data. *Computers and Chemical Engineering*, *152*, 1–8.

Su, C. J., & Huang, S. F. (2018). Real-time big data analytics for hard disk drive predictive maintenance. *Computers and Electrical Engineering*, *71*(July), 93–101.

Sun, L., Liu, T., Xie, Y., Zhang, D., & Xia, X. (2021). Real-time power prediction approach for turbine using deep learning techniques. *Energy*, *233*, 1–18.

Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, *11*(3), 812–820.

Tosato, D., Dalle Pezze, D., Masiero, C., Susto, G. A., & Beghi, A. (2020). *Alarm logs in packaging industry (ALPI)* (Technical Report). Università degli Studi di Padova. https://ieee-dataport.org/open-access/alarm-logs-packaging-industry-alpi

Uhlmann, E., Pontes, R. P., Geisert, C., & Hohwieler, E. (2018). Cluster identification of sensor data for predictive maintenance in a selective laser melting machine tool. In: B. Denkena, K. D. Thoben, & A. Trächtler (Eds.), *Procedia manufacturing* (Vol. 24, pp. 60–65). Elsevier B.V.

Ullah, I., Yang, F., Khan, R., Liu, L., Yang, H., Gao, B., & Sun, K. (2017). Predictive maintenance of power substation equipment by infrared thermography using a machine-learning approach. *Energies*, *10*(12), 1–13.

Una, M., Sahin, Y., Onat, M., Demetgul, M., & Kucuk, H. (2017). Fault diagnosis of rolling bearings using data mining techniques and boosting. *Journal of Dynamic Systems, Measurement and Control*, *139*(2), 1–12.

Usuga-Cadavid, J. P., Lamouri, S., Grabot, B., & Fortin, A. (2021). Using deep learning to value free-form text data for predictive maintenance. *International Journal of Production Research*, 1–29.

Venkataswamy, R., Uma Rao, K., & Meena, P. (2020). *Internet of things based metaheuristic reliability centered maintenance of distribution transformers* [Conference presentation]. Iop Conference Series: Earth and Environmental Science, Vol. 463, Bangkok, Thailand, 1–9.

Wang, D., Yan, R., Member, S., Mao, K., Shen, F., & Wang, J. (2018). Feature-based gated recurrent unit networks. *IEEE Transactions on Industrial Electronics*, *65*(2), 1539–1548.

Wang, F. K., & Mamo, T. (2019). Hybrid approach for remaining useful life prediction of ball bearings. *Quality and Reliability Engineering International*, *35*(7), 2494–2505.

Wang, K. (2016). Intelligent predictive maintenance (IPdM) system – Industry 4.0 scenario. *WIT Transactions on Engineering Sciences*, *113*, 259–268.

Wang, Q., Bu, S., & He, Z. (2020, oct). Achieving predictive and proactive maintenance for high-speed railway power equipment with LSTM-RNN. *IEEE Transactions on Industrial Informatics*, *16*(10), 6509–6517.

Wang, Q., Liu, J., Wei, B., Chen, W., & Xu, S. (2020, oct). Investigating the construction, training, and verification methods of k-means clustering fault recognition model for rotating machinery. *IEEE Access*, *8*, 196515–196528.

Wang, X. B., Zhang, X., Li, Z., & Wu, J. (2020). Ensemble extreme learning machines for compound-fault diagnosis of rotating machinery. *Knowledge-Based Systems*, *188*, 105012.

Wu, Z., Guo, Y., Lin, W., Yu, S., & Ji, Y. (2018). A weighted deep representation learning model for imbalanced fault diagnosis in cyber-physical systems. *Sensors (Switzerland)*, *18*(4), 1–15.

Wuest, T., Weimer, D., Irgens, C., & Thoben, K. D. (2016). Machine learning in manufacturing: Advantages, challenges, and applications. *Production and Manufacturing Research*, *4*(1), 23–45.

Xiong, M., Wang, H., Fu, Q., & Xu, Y. (2021). Digital twin–driven aero-engine intelligent predictive maintenance. *International Journal of Advanced Manufacturing Technology*, *114*(11–12), 3751–3761.

Xu, Y., Nascimento, N. M. M., de Sousa, P. H., Nogueira, F. G., Torrico, B. C., Han, T., Jia, C., & Rebouças Filho, P. P. (2021). Multi-sensor edge computing architecture for identification of failures short-circuits in wind turbine generators. *Applied Soft Computing*, *101*, 1–12.

Yang, B., Lei, Y., Jia, F., & Xing, S. (2019). An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings. *Mechanical Systems and Signal Processing*, *122*, 692–706.

Yang, C., Letourneau, S., Liu, J., Cheng, Q., & Yang, Y. (2017). Machine learning-based methods for TTF estimation with application to APU prognostics. *Applied Intelligence*, *46*(1), 227–239.

Yang, C., Sun, Y., Ladubec, C., & Liu, Y. (2021). Article developing machine learning-basedmodels for railway inspection. *Applied Sciences (Switzerland)*, *11*(1), 1–15.

Yang, Y., Zheng, H., Li, Y., Xu, M., & Chen, Y. (2019). A fault diagnosis scheme for rotating machinery using hierarchical symbolic analysis and convolutional neural network. *ISA Transactions*, *91*, 235–252.

Yu, W., Dillon, T., Mostafa, F., Rahayu, W., & Liu, Y. (2020, jan). A global manufacturing big data ecosystem for fault detection in predictive maintenance. *IEEE Transactions on Industrial Informatics*, *16*(1), 183–192.

Zenisek, J., Holzinger, F., & Affenzeller, M. (2019). Machine learning based concept drift detection for predictive maintenance. *Computers and Industrial Engineering*, *137*(August), 1–13.

Zenisek, J., Kronberger, G., Wolfartsberger, J., Wild, N., & Affenzeller, M. (2019). *Concept drift detection with variable interaction networks* [Conference presentation]. International Conference on Computer Aided Systems Theory, Las Palmas de Gran Canaria, Spain, 296–303.

Zhang, B., Zhang, S., & Li, W. (2019). Bearing performance degradation assessment using long short-termmemory recurrent network. *Computers in Industry*, *106*, 14–29.

Zhang, J., Wang, P., Yan, R., & Gao, R. X. (2018). Deep learning for improved system remaining life prediction. In *Procedia cirp* (Vol. 72, pp. 1033–1038). Elsevier B.V.

Zhang, S., Liu, C., Su, S., Han, Y., & Li, X. H. (2018). A feature extraction method for predictive maintenance with time-lagged correlation–based curve-registration model. *International Journal of Network Management*, *28*(5), 1–15.

Zhang, S., Ye, F., Wang, B., & Habetler, T. G. (2021). Semi-supervised bearing fault diagnosis and classification using variational autoencoder-based deep generative models. *IEEE Sensors Journal*, *21*(5), 6476–6486.

Zhang, W., Li, X., & Ding, Q. (2019). Deep residual learning-based fault diagnosis method for rotating machinery. *ISA Transactions*, *95*, 295–305.

Zhang, W., Yang, D., & Wang, H. (2019). Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, *13*(3), 2213–2227.

Zhang, Z., Li, S., Wang, J., Xin, Y., & An, Z. (2019). General normalized sparse filtering: A novel unsupervised learning method for rotating machinery fault diagnosis. *Mechanical Systems and Signal Processing*, *124*, 596–612.

Zhou, F., Gao, Y., & Wen, C. (2017). A novel multimode fault classification method based on deep learning. *Journal of Control Science and Engineering*, *2017*, 3583610.

Zhou, K. B., Zhang, J. Y., Shan, Y., Ge, M. F., Ge, Z. Y., & Cao, G. N. (2019). A hybrid multi-objective optimization model for vibration tendency prediction of hydropower generators. *Sensors (Switzerland)*, *19*(2055), 1–21.

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. *Computers and Industrial Engineering*, *150*(October), 1–17.

Zschech, P., Heinrich, K., Bink, R., & Neufeld, J. S. (2019, jun). Prognostic model development with missing labels: A condition-BasedMaintenance approach UsingMachine learning. *Business and Information Systems Engineering*, *61*(3), 327–343.

How to cite this article: Esteban, A., Zafra, A., & Ventura, S. (2022). Data mining in predictive maintenance systems: A taxonomy and systematic review. *WIREs Data Mining and Knowledge Discovery*, *12*(5), e1471. https://doi.org/10.1002/widm.1471