Reliability Engineering and System Safety 177 (2018) 12–23

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

Condition-based maintenance of naval propulsion systems: Data analysis with minimal feedback

Francesca Cipollini [a], Luca Oneto [a], Andrea Coraddu [b,*], Alan John Murphy [c], Davide Anguita [a]

[a] DIBRIS - University of Genova, Via Opera Pia 13, Genova, I-16145, Italy
[b] Naval Architecture, Ocean & Marine Engineering Strathclyde University, Glasgow, G1 1XW, UK
[c] Marine, Offshore and Subsea Technology Group, School of Engineering Newcastle University, Newcastle upon Tyne, NE1 7RU, UK

ARTICLE INFO

Keywords:
Data analysis
Naval propulsion systems
Condition-based maintenance
Supervised learning
Unsupervised learning
Novelty detection
Minimal feedback.

ABSTRACT

The maintenance of the several components of a Ship Propulsion Systems is an onerous activity, which need to be efficiently programmed by a shipbuilding company in order to save time and money. The replacement policies of these components can be planned in a Condition-Based fashion, by predicting their decay state and thus proceed to substitution only when really needed. In this paper, authors propose several Data Analysis supervised and unsupervised techniques for the Condition-Based Maintenance of a vessel, characterised by a combined diesel-electric and gas propulsion plant. In particular, this analysis considers a scenario where the collection of vast amounts of labelled data containing the decay state of the components is unfeasible. In fact, the collection of labelled data requires a drydocking of the ship and the intervention of expert operators, which is usually an infrequent event. As a result, authors focus on methods which could allow only a minimal feedback from naval specialists, thus simplifying the dataset collection phase. Confidentiality constraints with the Navy require authors to use a real-data validated simulator and the dataset has been published for free use through the OpenML repository.

1. Introduction

Maintenance is one of the most critical tasks to be designed and programmed by any product-selling company [1–4]. As a fact, every complex system is designed assigning it a specific life-cycle, which is influenced by different factors such as the raw materials adopted, the estimated working hours, and the environmental conditions [5,6]. Nevertheless, this time-to-live information is inevitably inaccurate as it is impossible to predict at design phase the exact working conditions in which the system will operate [7,8]. In any case, the decay of the system components will require at some point of time to be repaired or replaced, thus leading to the system halt to perform some maintenance tasks [9]. This is the reason why an efficient maintenance program can be time and costs saving since replacing a malfunctioning component after it has failed during service, results in multiple downsides for the system owner company.

The Shipbuilding industry is particularly affected by this problem, as a ship breakdown necessarily requires a drydocking, and retrieving a stricken vessel offshore is not a trivial task [10,11]. A correct maintenance program ensures that a ship works as it was designed, with the desired level performances, without impacting the service [12]. Maintenance policies can be divided into two main categories [13,14]: Corrective (CM), and Preventive (PM).

CM has been for many years the only way of performing maintenance, by replacing a component only after its breakdown, thus compromising the overall system availability and causing exceptional costs and loss in incomes [15]. In PM, instead, a component is replaced when it reaches the end of its life cycle before a possible breakdown. One of the traditional ways to perform PM is to predetermine a conservative average estimation of the component time-to-live adopting the experience gained with all the components belonging to a specific class [16]. Similarly to CM, this particular type of PM, usually called Predetermined Maintenance (PRM), can bring unnecessary costs, if the replaced component could have been used more than originally forecast. Moreover, PRM does not guarantee to limit the number of faults in a fleet, since a breakdown could still happen before the replacement takes place. In this case, there is a trade-off between the number of breakdowns and the lifetime estimation of the components, which is not easy to reach since the actual ship usage can be very different from ship to ship. Nevertheless, Condition-Based Maintenance (CBM) can be considered as another way of performing PM, which aims at reducing both the costs of CM and PRM by relying on the exact decay state of each component and

* Corresponding Author.
E-mail addresses: francesca.cipollini@edu.unige.it (F. Cipollini), luca.oneto@unige.it (L. Oneto), andrea.coraddu@strath.ac.uk (A. Coraddu), a.j.murphy@newcastle.ac.uk (A.J. Murphy), davide.anguita@unige.it (D. Anguita).

https://doi.org/10.1016/j.ress.2018.04.015
Received 25 August 2017; Received in revised form 20 December 2017; Accepted 18 April 2018
Available online 19 April 2018

then by efficiently planning its maintenance [17,18]. Note that, condition monitoring and failure prediction are two different concepts which are somehow strictly related. In fact, a failure of a component is predictable only if it is preceded by a decay in its performance or in the performance of some related component [19,20].

Since, in most cases, the decay state of each component cannot be tracked with a sensor, CBM requires a model able to predict it based on other sensors available. In fact, the decay state cannot be easily measured without an interruption of service or a drydock of the ship, situation which is usually avoided. To overcome this problem, the available on-board sensors can be used to collect a huge amount of real-time data which can be stored into historical datasets and adopted in order to formulate a statistical Data-Driven Model (DDM) to predict the exact components decay [21]. In fact, DDMs exploit advanced statistical techniques to build models directly based on the significant amount of data produced by the logging and monitoring apparatus, without requiring any a priori knowledge of the underlining physical phenomena [22–24]. Considering the estimated state of decay, it is possible to schedule each component's replacement before failures occur, maximising its life cycle, according to the time required for each maintenance [11]. As a result, the additional costs of CM and PRM can be replaced with the lower ones of equipping the propulsion system with sensors and by collecting, storing, and analysing these data for the purpose of creating effective predictive DDMs [10,11].

In this paper, authors address the problem of building effective DDMs to predict the main components decay state in a Naval Propulsion System (NPS) for CBM purposes. In particular, the decay of a vessel Gas Turbine (GT), Gas Turbine Compressor (GTC), Hull (HLL) and Propeller (PRP) is estimated. Many examples of Data Analysis (DA) techniques applied to different CBM problems can be found in literature [25]. Among other, Support Vector Machines [26], Hidden Markov Models [27] and Kalman filter [28] are the most frequently used. Examples of DA approaches applied to the marine industry can be found in [29], where a standard Neural Network approach is used to improve monitoring of Gas Turbines, while Kernel based methods are applied in [30,31]. In [32,33] image processing techniques are adopted for hull condition assessment. In [34] the engine and propeller state is predicted adopting an Artificial Neural Network. A complete overview can be found in [35].

In particular, this work can be seen as the continuation of Coraddu et al. [31], where a similar approach was attempted adopting a smaller amount of decayed NPS components and supervised Machine Learning (ML) regression models in order to predict their exact decay. Nevertheless, in [31] it was proven that a significant amount of historical data needed to be collected, together with the actual state of decay of each component. In the end, this approach resulted not feasible in a real-world scenario where the labelling process requires the intervention of an experienced operator and, in some cases, to stop the vessel or even to put the ship in a dry dock.

As a result, authors here propose a different approach where collecting labeled samples is an easier task that can be performed by less experienced operators since the raw information about the decay is requested and it can be retrieved without impacting the ship activities. Specifically, two approaches are here proposed and compared. First, authors build virtual sensors able to continuously estimate the need for replacement of the components based on other sensors measurements which are indirectly influenced by this decay. Then authors try to perform the same analysis, in conditions where only few labelled samples are present, by adopting a DDM which require a limited amount of information to achieve satisfying performance. To the best knowledge of the authors, the novelty of the proposed work relies on its ability of building a model whose accuracy is comparable with the state-of-the-art supervised learning techniques, adopting only an extremely limited number of labelled samples.

For this reason, firstly authors performed a traditional classification analysis where the target is to estimate the label state of the components described with an efficiency coefficient. The analysis has been carried out comparing different state-of-the-art methodologies such as Kernel Methods [36], Neural Network [37], Gaussian Processes [38], Similarity Based Method [39], and Ensemble Methods [40]. These binary classification techniques are adopted to predict if the efficiency coefficient is above or below a certain threshold defined by the accepted loss in efficiency of the NPS components. Secondly, the same problem has been tackled with another state-of-the-art approach which, in principle, does not need any labelled sample since it searches for novel behaviour in the data though a novelty detection algorithms [41,42]. Results show that with just a few labelled samples it is possible to fine tune this last methodology to achieve satisfying performances.

This work is the natural continuation of Coraddu et al. [31], where authors presented a dataset published trough the University of California Irvine (UCI) website of data coming from a simulator of a Frigate, characterised by a COmbined Diesel ELectric And Gas (CODLAG) propulsion plant. A similar simulator was adopted in this study, characterised by a higher amount of decaying components, and it will be published through the OpenML dataset repository [43].

The paper is organised as follows. Section 2 reports a general description of the vessel, the numerical model, and the degradation phenomena. Section 3 presents a description of the dataset extracted from the numerical simulator and published through OpenML. Section 4 reports the proposed DDMs. Results of the DDMs tested on the proposed data are reported in Section 5 with conclusions in Section 6.

2. Naval propulsion system

2.1. Vessel description

In this work authors focus on a Frigate, characterised by a CODLAG NPS, widespread detailed in [31]. In particular, the GT mechanically drives the two Controllable Pitch Propellers (CPP) through a cross-connected gearbox (GB). Besides, each shaft has its electric propulsion motor (EPM) mounted on the two shaft-lines. Two clutches between the GB and the two EPM and another clutch between the GT and the GB assure the possibility of using two different type of prime movers, i.e. EPM and GT. Finally, the electric power is provided by four diesel generators (DG). This particular GB arrangement, allows the vessel to operate under different propulsive configurations to achieve the requirements of the vessel's mission profile. The vessel is characterised by the following mission profiles: Anti-Submarine Warfare (ASW), General-Purpose (GEP) and Anti-Aircraft Warfare (AAW). In particular, for the ASW profile, the EPMs are prime movers while the GT is disconnected through the clutches. Under the GEP mission profile, the GT is the prime mover while the EPMs are working as shaft generators. Finally, for the AAW mission profile both the GT and the EPM are the prime movers. In this work, only the GT operating conditions have been taken into account.

2.2. Model description

In this work, authors consider an NPS numerical model developed in the Matlab® Simulink® software environment within many years of research [44]. The numerical model is composed of several modules each one representing a single propulsion component such as the hull, the main engines, the propellers, the rudders, the GB, and the control system. In the previous literature, authors presented a model that considers the GT and GTC decay performance [31]. The model is now further improved to take into account the performance decay of the HLL and PRP, and is now readily to undertake a holistic approach in addressing the performance decay by accounting the important components as follows:

1. Gas Turbine (GT);
2. Gas Turbine Compressor (GTC);
3. Hull (HLL);
4. Propeller (PRP).

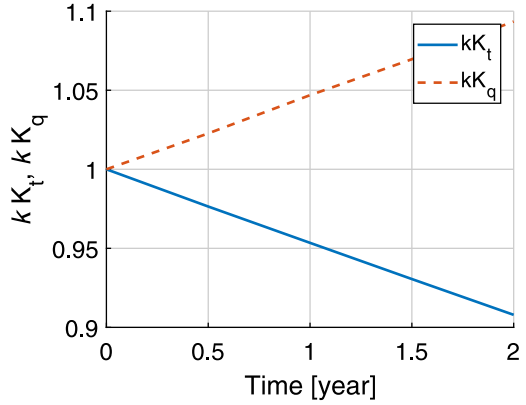

Fig. 1. Correction factors for thrust reduction, kK_t, and torque increase, kK_q.

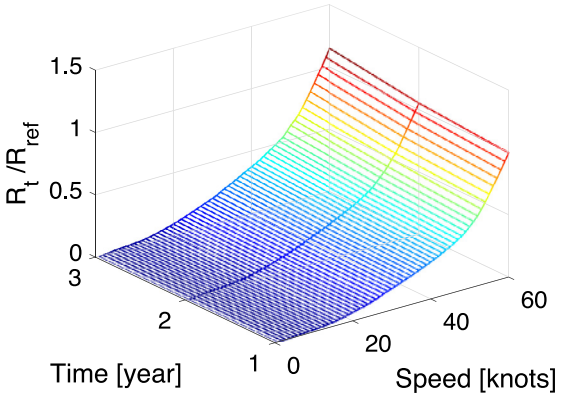

Fig. 2. Hull resistance increase behaviour over time and speed.

2.3. Degradation model

2.3.1. GTC and GT degradation model

As reported in [45,46], fouling of the GTC increases the specific fuel consumption and the temperature of the exhaust gas. In agreement with previous work, [31], the effect of the fouling is simulated by reducing the numerical values of the airflow rate M_c and of the isentropic efficiency η_c with a reduction factor kM_c and K_{η_c}. The detailed description of the GTC and GT degradation model is provided in [31,47]. The authors applied a reduction factor, kM_T, to the GT flow rate to represent the effect of fouling.

2.3.2. PRP degradation model

An increase in the roughness of the blade surface is the primary cause of marine PRP performance degradation [48]. This is caused by accretions of marine organisms to the metal, alloy erosion and corrosion, or combinations of these elements. The PRP decay status has been modelled by increasing the torque coefficient (K_q) and by reducing the thrust coefficient (K_t). The correction factors used for thrust reduction, kK_t, and torque increase, kK_q, have been derived from Atlar et al. [49] and Wan et al. [50] and reported in Fig. 1.

2.3.3. HLL degradation model

The main factors that affect HLL performance are the shape, the coating used and the extent of fouling [51]. The resistance increase due to fouling has been modelled utilising a correction factor kH based on the information reported in [52]. Using the towing tank data provided by the ship owner the resistance has been evaluated after one and two years of vessel operations in agreement with the data available in the literature [47]. In Fig. 2 the behaviour of the HLL resistance referred to the trial condition (R_t/R_{tref}) over the time and speed is reported.

3. From data to condition-based maintenance

3.1. Dataset creation

In this work, authors will use the real-data validated complex numerical simulator of a Navy frigate described in Section 2 to build a realistic set of data for designing and test purposes of DDMs. This dataset will be released for the use to the research community on the widespread well-known dataset repository OpenML [43]. Currently, it can be downloaded from https://cbm-anomaly-detection.smartlab.ws.

The NPS model input parameters are provided in [53] however, for clarity in this paper they are repeated verbatim here as follows:

- Speed: this parameter is controlled by the control lever. The latter can only assume a finite number of positions lp_i with $i \in \{0, \cdots, 9\}$, which in turn correspond to a finite set of possible configurations for fuel flow and blade position. Each set point is designed to reach a desired speed v_i with $i \in \{0, \cdots, 9\}$:

$$v_i = 3 * lp_i[\text{Knots}], \quad \forall i \in \{0, \cdots, 9\}. \tag{1}$$

  Note that, if the transients is not taken into account, lp_i and v_i are deterministically related by a linear law. In the presented analysis the transients between different speeds have been not considered.
- As reported in Section 2.3, the PRP thrust and torque decay limit over two years of operations are:

$$kK_t \in [0.9, 1.0], \quad kK_q \in [1.0, 1.1] \tag{2}$$

  kK_t and kK_q are respectively the components which define the decay of the torque and the thrust provided by the propeller in time. They are linearly correlated, since as the first decay of a certain quantity, the latter decay of the same quantity ($1 - kK_t = kK_q - 1$). For this reason only kK_t will be analysed, considering the linear dependency between the two variables.
- The HLL decay has been modelled according to the available literature [52] as described in Section 2.3. The decay limits over two years of operations are:

$$kH \in [1, 1.2] \tag{3}$$

- GTC decay:

$$kM_c \in [0.95, 1.0] \tag{4}$$

- GT decay:

$$kM_t \in [0.975, 1.0] \tag{5}$$

The performance decay functions described in Section 2.3 have been empirically derived as functions of the time variable solely. The real degradation behaviour of the physical asset should be defined through specific functions able to express the time dependency, the mutual interactions between the subsystems and the real operational profile. To overcome this issue, authors considered each possible combination of GTC, GT, HLL, and PRP decay status based on the described functions, and sampled the range of decays with a uniform grid characterised by a degree of precision sufficient to have a proper granularity of representation. Given the above premises, the evolution of the system between two important dry dock maintenance for HLL and PRP can be exhaustively and realistically explored by simulating all its possible decayed states, as all the components are decaying the same time.

The space of possible states is described via the following tuple:

$$\left(lp, kK_t, kK_q, kH, kM_c, kM_t\right)_i, \quad i \in \{1, \cdots, 455, 625\} \tag{6}$$

since:

$$lp \in S^{lp} = \{0, 3, 6, \cdots, 27\}, \tag{7}$$

$$kK_t \in S^{kK_t} = \{0.9, 0.9 + 0.1/14, 0.09 + 0.2/14, \cdots, 1.0\}, \tag{8}$$

Table 1
Measured values available from the continuous monitoring system.

#	Variable name	Unit
1	Lever position	[ ]
2	Vessel speed	[knots]
3	GT shaft torque	[kN m]
4	GT speed	[rpm]
5	PRP thrust (starboard)	[N]
6	PRP thrust (port)	[N]
7	Shaft torque (port)	[kN m]
8	Shaft speed (port)	[rpm]
9	Shaft torque (starboard)	[kN m]
10	Shaft speed (starboard)	[rpm]
11	HP GT exit temperature	[°C]
12	Gas generator speed	[rpm]
13	Fuel flow (mf)	[kg/s]
14	TIC control signal	[%]
15	GTC outlet air pressure	[bar]
16	GTC outlet air temperature	[°C]
17	External pressure	[bar]
18	HP GT exit pressure	[bar]
19	TCS TIC control signal	[ ]
20	Thrust coefficient (starboard)	[ ]
21	PRP speed (starboard)	[rps]
22	Thrust coefficient (port)	[ ]
23	PRP speed (port)	[rps]
24	PRP torque (port)	[kN m]
25	PRP torque (starboard)	[kN m]

$$kK_q = 2 - kK_t, \tag{9}$$

$$kH \in S^{kH} = \{1.0, 1.0 + 0.2/14, 1.0 + 0.4/14, \cdots, 1.2\}, \tag{10}$$

$$kM_c \in S^{kM_c} = \{0.95, 0.95 + 0.05/14, 0.95 + 0.1/14, \cdots, 1.0\}, \tag{11}$$

$$kM_t \in S^{kM_t} = \{0.975, 0.975 + 0.025/14, 0.975 + 0.05/14, \cdots, 1.0\}. \tag{12}$$

Note that the total number of samples 455,625 is the result of making a simulation for each possible combination of decay status (15 values for GTC, 15 for GT, 15 for HLL, and 15 PRP) and speed (9 values). Once these quantities are fixed, the numerical model is run until the steady state is reached. Then, the model is able to provide all the quantities reported in Table 1. These subsets of models outputs are the same quantities that the automation system installed on-board can acquire and store.

The simulator was run on a server equipped with four Intel® Xeon® CPU E5-4620 2.2 GHz, 128GB of RAM, 120GB SSD disk, and Matlab® R2016a.

3.2. Condition-based maintenance

This study aims at estimating the four decay variables described in the previous section, adopting different DA techniques. This section reports how the data generated can be used to create effective predictive DDMs for the CBM of an NPS.

The data described in Section 3.1 contain two sets of information: one regarding the quantities that the automation system installed on-board can acquire and store and the other one regarding the associated state of decay (efficiency coefficient) of the different NPS components (GT, GTC, HLL, and PRP).

This problem could have been straightforwardly mapped into a classical multi-output regression problem, as in [31], where the aim is to predict the actual decay coefficient based on the automation data coming from the sensors installed on-board [22]. Unfortunately, this approach cannot be adopted in a real operational scenario. While the sensors' data coming from the automation system are easy to collect, the information regarding the associated state of decay is not so easy to retrieve. In fact, to circumvent this challenge to prove DDMs authors exploited a numerical model for gathering all the information and build the dataset presented in Section 3.1. In practice, instead, retrieving the state of decay of the different NPS components requires the intervention of an experienced operator and, in some cases, to stop the vessel or even to put the ship in a dry dock. Moreover, data-driven regression models require a huge amount of historical data and therefore a long acquisition time.

Based on these considerations, authors decided to build simplified DDMs able to detect if the component state of decay is above or below a certain threshold. These thresholds represent the accepted loss in efficiency of the NPS components and the consequently sustainable costs of keeping a less performing vessel operative. This approach represents an abstraction of the problem which allows a more practical collection of the state of decay of the component. In fact, instead of requiring the precise state of decay, this approach only requires detecting if the decay state of the components is acceptable or not. Consequently, the collection of these data can be performed by less experienced operators since raw information about the decay is requested and can be retrieved without impacting the ship activities. This new problem can be straightforwardly mapped into a classical multi-output binary classification problem [22] where the aim is to predict if the decay state of an NPS component is acceptable or not based on the automation data coming from the sensors installed on-board.

The data described in Section 3.1 can be easily exploited to tackle this new problem as well. In fact, by thresholding kK_t, kH, kM_c, and kM_t the corresponding binary valued state of decay of the NPS components are obtained. In other words, if the efficiency coefficients are above or below a defined threshold, based on the accepted loss in efficiency of the NPS components, they will be tagged as "decayed" or "not decayed". Thresholds were fixed according to the least affordable value of decay of the single component. Defining these thresholds is not a trivial task. Authors approach is to define the maximum level of inefficiency that the operator or the shipowner is willing to tolerate before taking action and re-establish the efficiency of the system. The authors considered two years as a typical time frame between two important dry dock maintenance for HLL and PRP.

The HLL and PRP thresholds have been defined considering one year of operation. The proposed limits are just an example of the possible selection that is possible to setup to implement a CBM framework

$$kK_t \begin{cases} [0.9 - 0.95) & \text{decayed} \\ [0.95 - 1] & \text{not decayed} \end{cases} \tag{13}$$

$$kH \begin{cases} (1.1 - 1.2] & \text{decayed} \\ [1 - 1.1] & \text{not decayed} \end{cases} \tag{14}$$

As for GT and GTC, an effective time service of 2000 h per year is considered as a reasonable operating time for these vessel types. In agreement with these observations authors defined the following thresholds based on the knowledge of the time domain decay functions:

$$kM_c \begin{cases} [0.95 - 0.98) & \text{decayed} \\ [0.98 - 1] & \text{not decayed} \end{cases} \tag{15}$$

$$kM_t \begin{cases} [0.975 - 0.99) & \text{decayed} \\ [0.99 - 1] & \text{not decayed} \end{cases} \tag{16}$$

Results will show that estimating if the decay state is acceptable or not, instead of estimating its specific state, remarkably reduces the number of samples required to find accurate DDMs. However, this quantity is still too large with respect to what can be collected in a real operational scenario.

To solve this issue, authors tried to look at the same problem from another perspective. Specifically, it is reasonable to state that, for the vast majority of the time, NPS components of the ships operate in an acceptable state of decay. Consequently, most of the sensor data collected by the automation system represent ordinary operating conditions corresponding to a reasonable decay state of the NPS components (GT, GTC, HLL, and PRP). Just very few times during the ship lifetime it happens that it has to operate with over-decayed components. If, for some reasons, one or more NPS components decay too fast, the corresponding automation data measurements will deviate from their expected behaviour. This new problem can be straightforwardly mapped into a classical outlier (novelty) detection problem [36,42,54] where the aim is to detect unexpected behaviour in the sensor data collected by the automation system which may correspond to an over-decayed state of an NPS components. This method does not require to know either the actual state of decay of the components, as a regression task would do, or the less detailed information about "decayed" or "not decayed", as in the binary classification framework. In this case, the method just needs the sensor data collected by the automation system (see Table 1) without any supervision or feedback from the operator. These kinds of DDMs try to build a model of the "usual" operational profile of the ship and automatically detect if the sensor data collected by the automation system are "deviating too much" from the established behaviour. In our context "usual" means that the efficiency of GT, GTC, HLL, and PRP are in the acceptable range while "deviating too much" means that they are not in the acceptable range, according to Eqs. (13)–(16).

As for the binary classification framework, the data described in Section 3.1 can be easily exploited to tackle this problem as well. In fact, it is just necessary to keep the data corresponding to an acceptable decay state with respect to kK_t, kH, kM_c, and kM_t and in accordance with Eqs. (13)–(16). Finally, for testing and tuning the DDMs, it is possible to use just a few samples of the dataset corresponding to an unacceptable decay state. Note that these are the only samples which are costly to retrieve since they are the only ones that require the intervention of expert operators. Results will show that with just very few samples (≈ 10) of decayed state of the vessel, it is possible to obtain effective DDMs for CBM of NPS.

As a final remark, authors would like to recall that each navy frigate is characterised by different mission profiles (AAW, ASW, and GEP) as described in Section 2.1. Each mission profile is characterised by a particular use of the ship in terms of speed. CBM DDMs for NPS do not need to estimate the state of decay of the NPS components for all the possible mission profiles. In fact, the vessel operates at a cruise speed (which is approximately ≈ 15 knots) while the time spent by the vessel at different speeds is negligible. For this reason, authors conduct the same analysis described in the previous paragraph by setting $lp = 15$. Results will show that this simplification will further reduce the amount of historical data needed to build effective CBM DDMs for NPS.

4. Machine learning

In this section, authors will present the ML techniques adopted in order to build the CBM DDMs for NPS described in Section 2, based on the data outlined in Section 3.

Let authors consider an input space $\mathcal{X} \subseteq \mathbb{R}^d$ and an output space $\mathcal{Y}$. Note that, for what concerns this paper, $\mathcal{X}$ takes into account the different sensors measurements, also called features, reported in Table 1, while the output space $\mathcal{Y}$ depends on the particular problem identified in Section 3.2. ML techniques aim at estimating the unknown rule $\mu : \mathcal{X} \to \mathcal{Y}$ which associates an element $y \in \mathcal{Y}$ to an element $\boldsymbol{x} \in \mathcal{X}$. Note that, in general, μ can be non-deterministic. An ML technique estimates μ through a learning algorithm $\mathscr{A}_{\mathcal{H}} : \mathcal{D}_n \times \mathcal{F} \to h$, characterized by its set of hyperparameters $\mathcal{H}$, which maps a series of examples of the input/output relation contained in a dataset of n samples $\mathcal{D}_n : \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}$ into a function $f : \mathcal{X} \to \mathcal{Y}$ chosen in a set of possible ones $\mathcal{F}$.

When both $\boldsymbol{x}_i$ and y_i with $i \in \{1, \cdots, n\}$ are available, the problems is named supervised and consequently supervised ML technique are adopted [22]. Classification is one of the most popular examples of supervised ML problems [36]. In classification, the output space is composed of a finite set of c possibilities $\mathcal{Y} \in \{C_1, \cdots, C_c\}$. Binary classification is a particular example of classification problem where $\mathcal{Y} \in \{\pm 1\}$.

When just $\boldsymbol{x}_i$ with $i \in \{1, \cdots, n\}$ are available, which means that the associated element of the output space y_i with $i \in \{1, \cdots, n\}$ is not explicitly known, it has to be assumed that "similar" $\boldsymbol{x}_i$ are associated with "similar" y_i where the concept of similarity is something that needs to be defined based on μ. In this last case, the ML problems are called unsupervised, and consequently, unsupervised ML techniques need to be adopted [55]. Anomaly (novelty, outlier) detection is a common example of unsupervised learning problem where the unknown $y \in \mathcal{Y}$ can assume only two possible values: -1 for "non-anomaly" and $+1$ for "anomaly" [36].

The error that f commits in approximating μ is measured with reference to a loss function $\ell : \mathcal{X} \times \mathcal{Y} \times \mathcal{F} \to [0, \infty)$. Obviously, the error that f commits over $\mathcal{D}_n$, is optimistically biased since $\mathcal{D}_n$ has been used, together with $\mathcal{F}$, for building f itself. For this reason, another set of fresh data, composed of m samples and called test set $\mathcal{T}_m = \{(\boldsymbol{x}_1^t, y_1^t), \cdots, (\boldsymbol{x}_m^t, y_m^t)\}$, needs to be exploited. Note that, $\boldsymbol{x}_i^t \in \mathcal{X}$ and $y_i^t \in \mathcal{Y}$ with $i \in \{1, \cdots, m\}$, and the association of y_i^t to $\boldsymbol{x}_i^t$ is again made based on μ. Moreover, both for supervised and unsupervised problems $\mathcal{T}_m$ must contain both $\boldsymbol{x}_i^t \in \mathcal{X}$ and $y_i^t \in \mathcal{Y}$ with $i \in \{1, \cdots, m\}$ to estimate the error of f, while, for unsupervised learning problems, y_i with $i \in \{1, \cdots, n\}$ in $\mathcal{D}_n$ is unknown.

4.1. Measuring the error

In this work, many state-of-the-art ML techniques will be tested and their performances will be compared to understand what is the most suited solution for building CBM DDMs for NPS.

In order to perform this analysis, authors have to define different measures of error, also called indexes of performance, able to well characterize the quality of the different CBM DDMs for NPS. Once f has been chosen based on $\mathcal{D}_n$, it is possible to use the fresh set of data $\mathcal{T}_m$ in order to compute its error based on different losses. The choice of the loss strongly depends on the problem under examination [56].

In the classification framework, the most natural choice as loss function is the Hard loss one, which counts the number of misclassified samples $\ell_H(f(\boldsymbol{x}), y) = [f(\boldsymbol{x}) \neq y]$. Note that the Iverson bracket notation is exploited. In this work, only binary classification problems are investigated, then the Hard loss function can be expressed as $\ell_H(f(\boldsymbol{x}), y) = 1 - yf(\boldsymbol{x})/2$. Moreover, this measure will be also used for the anomaly detection problems since, also in this case, a binary output is considered (non-anomaly or anomaly).

Based on the Hard loss it is possible to define different indexes of performance [57]:

- the Average Misclassifications Rate (AMR) is the mean number of misclassified samples: $\text{AMR} = \frac{1}{m}\sum_{i=1}^{m} \ell_H(f(\boldsymbol{x}_i^t), y_i^t)$;
- the Confusion Matrix, which measures four different quantities:
  - $\text{TN} = 100/m \sum_{i=1}^{m} [f(\boldsymbol{x}_i^t = y_i^t \wedge y_i^t = -1]$ which is the percentage of true negative;
  - $\text{TP} = 100/m \sum_{i=1}^{m} [f(\boldsymbol{x}_i^t = y_i^t \wedge y_i^t = +1]$ which is the percentage of true positive;
  - $\text{FN} = 100/m \sum_{i=1}^{m} [f(\boldsymbol{x}_i^t \neq y_i^t \wedge y_i^t = -1]$ which is the percentage of false negative;
  - $\text{FP} = 100/m \sum_{i=1}^{m} [f(\boldsymbol{x}_i^t \neq y_i^t \wedge y_i^t = +1]$ which is the percentage of false positive.

4.2. Machine learning techniques

In this section, authors will present the supervised and unsupervised learning algorithms exploited in this paper for building CBM DDMs for

NPS. Moreover, authors will show how to tune their performances by tuning their hyperparameters during the so-called Model Selection (MS) phase [58–60]. Finally, authors will also check for possible spurious correlation in the data by performing the Feature Selection (FS) phase [61–65]. In fact, once f is built based on the different learning algorithm and has been confirmed to be a sufficiently accurate representation of μ, it can be interesting to investigate how the model f is affected by the different features that have been exploited to build f itself during the feature ranking procedure [61]. As authors will describe later, for some algorithms, the feature ranking procedure is a by-product of the learning process itself and allows to simply check the physical plausibility of f.

4.2.1. Supervised classification learning algorithms

Supervised ML techniques can be grouped into different families, according to the space of function $\mathcal{F}$ from which the learning algorithm chooses the particular f, the approximation of μ, based on the available data $\mathcal{D}_n$. In fact, techniques belonging to the same family, share an affine $\mathcal{F}$. Among the several possible ML families, authors choose the state-of-the-art ones which are commonly adopted in real-world application, and, in each family, the best performing techniques are selected. In particular, Neural Networks (NNs), Kernel Methods (KMs), Ensemble Methods (EMs), Bayesian Methods (BMs), and Lazy Methods (LMs) are adopted.

NNs are ML techniques which combine together many simple models of a human brain neuron, called perceptrons [37], in order to build a complex network. The neurons are organized in stacked layers connected together by weights that are learned based on the available data via backpropagation [66]. The hyperparameters of an NN $\mathcal{H}^{\text{NN}}$ are the number of layers h_1 and the number of neurons for each layer $h_{2,i}$ with $i \in \{1, \cdots, h_1\}$. Note that it is assumed that NN with only one hidden layer has $h_1 = 1$. If the architecture of the NN consists of only one hidden layer, it is called shallow (SNN) [67,68], while, if multiple layers are staked together, the architecture is defined as deep (DNN) [69–71]. Extreme Learning Machines (ELMs) are a particular kind of SNN, where the weights of the first layer are randomly chosen, while the ones of the output layers are computed according to the Regularized Least Squares (RLS) principle [72–74]. The hyperparameters of the ELM $\mathcal{H}^{\text{ELM}}$ are the number of neurons of the hidden layer, h_1, and the RLS regularization hyperparameter h_2 [75].

KMs are a family of ML techniques which exploits the "Kernel trick" for distances in order to extend linear techniques to solve non-linear problems [76,77]. In the case of classification, KMs select the function f minimizing the trade-off between the sum of the accuracy over the data, namely the empirical error, and the solution complexity, namely the regularization term [76,78,79]. The most effective KM techniques are: Kernelized Regularized Least Squares (KRLS), and Support Vector Machines (SVMs). The hyperparameters of the KRLS $\mathcal{H}^{\text{KRLS}}$ are: the kernel, which in this paper is fixed to Gaussian Kernel for the reasons described in [80,81], its hyperparameter h_1 and the regularization hyperparameter h_2. SVM, instead, is a classification method, which roots in the Statistical Learning Theory [22] and differs from the KRLS mainly because of its particular loss function [56]. The hyperparameters of the SVM are the same as the one of the KRLS.

EMs ML techniques rely on the fact that combining the output of several classifiers results in a much better performance than using any one of them alone [40,82]. Random Forest (RF) [40] and Random Rotation Ensembles (RRF) [83], two popular state-of-the-art and widely adopted methods, combine many decision trees in order to obtain effective predictors which have limited hyperparameter sensitivity and high numerical robustness [84,85]. Both RF and RRF have hidden hyperparameters which are arbitrarily fixed in this work because of their limited effect [86].

BMs are ML techniques where, instead of choosing a particular $f \in \mathcal{F}$ a distribution for choosing $f \in \mathcal{F}$ is defined [87]. Gaussian Processes (GP) learning algorithm is a popular BM [38] which employs a collection of Gaussians in order to compute the posterior distribution of the $f(\boldsymbol{x})$. In fact, this algorithm defines the probability distribution of the output values as a sum of Gaussians whose variance is fixed according to the training data. The hyperparameter of the GP $\mathcal{H}^{\text{GP}}$ is the parameter which governs the Gaussians width h_1.

LMs ML techniques are learning method in which the definition of f is delayed until $f(\boldsymbol{x})$ needs to be computed [39]. LMs approximate μ locally with respect to $\boldsymbol{x}$. K-Nearest Neighbors (KNN) is one of the most popular LM due to its implementation simplicity and effectiveness [88]. The hyperparameter of the KNN $\mathcal{H}^{\text{KNN}}$ is the number of neighbors of $\boldsymbol{x}$ to be considered h_1.

4.2.2. Unsupervised learning algorithms for anomaly detection

Similarly to their supervised counterpart, also unsupervised ML methods can be divided into different families. Since this work deals only with anomaly detection problems, authors will recall the most known and effective techniques for solving these problems according to Swersky et al. [42]. In particular Swersky et al. [42] shows that two anomaly detection methods based on SVM and KNN respectively, are the top choices in this context.

In particular One-Class SVM (OCSVM) is a boundary-based anomaly detection method, inspired by SVM, which enclose the inlier class in a minimum volume hypersphere by minimizing a Tikhonov regularization problem, similar to the one reported for SVM framework. Like traditional SVMs, OCSVM can also be extended to non-linearly transformed spaces using the "Kernel trick" for distances. The hyperparameters OCSVM $\mathcal{H}^{\text{OCSVM}}$ are the same as the ones of SVM.

The Global KNN (GKNN), inspired by the KNN, has been originally introduced as an unsupervised distance-based outlier detection method [42,89]. The hyperparameter GKNN $\mathcal{H}^{\text{GKNN}}$ is the same as the one of KNN.

4.2.3. Model selection

MS deals with the problem of tuning the hyperparameters of each learning algorithm [60]. Several methods exist for MS purpose: resampling methods, like the well-known k-Fold Cross Validation (KCV) [58] or the nonparametric Bootstrap (BTS) approach [90,91], which represent the state-of-the-art MS approaches when targeting real-world applications. Resampling methods rely on a simple idea: the original dataset $\mathcal{D}_n$ is resampled once or many (n_r) times, with or without replacement, to build two independent datasets called training, and validation sets, respectively $\mathcal{L}_l^r$ and $\mathcal{V}_v^r$, with $r \in \{1, \cdots, n_r\}$. Note that $\mathcal{L}_l^r \cap \mathcal{V}_v^r = \emptyset$, $\mathcal{L}_l^r \cup \mathcal{V}_v^r = \mathcal{D}_n$. Then, to select the best combination the hyperparameters $\mathcal{H}$ in a set of possible ones $\mathfrak{H} = \{\mathcal{H}_1, \mathcal{H}_2, \cdots\}$ for the algorithm $\mathscr{A}_{\mathcal{H}}$ or, in other words, to perform the MS phase, the following procedure has to be applied:

$$\mathcal{H}^* : \quad \min_{\mathcal{H} \in \mathfrak{H}} \frac{1}{n_r} \sum_{r=1}^{n_r} \frac{1}{v} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{V}_v^r} \ell(\mathscr{A}_{\mathcal{H}, \mathcal{L}_l^r}(\boldsymbol{x}_i), y_i), \tag{17}$$

where $\mathscr{A}_{\mathcal{H}, \mathcal{L}_l^r}$ is a model built with the algorithm $\mathscr{A}$ with its set of hyperparameters $\mathcal{H}$ and with the data $\mathcal{L}_l^r$. Since the data in $\mathcal{L}_l^r$ are independent from the ones in $\mathcal{V}_v^r$, the idea is that $\mathcal{H}^*$ should be the set of hyperparameters which allows to achieve a small error on a data set that is independent from the training set.

Note that, for the anomaly detection problem, the algorithms do not need any label in $\mathcal{L}_l^r$, consequently authors just need the labelled data for $\mathcal{V}_v^r$.

If $r = 1$, if l and v are aprioristically set such that $n = l + v$, and if the resample procedure is performed without replacement, the hold out method is obtained [60]. For implementing the complete KCV, instead, it is needed to set $r \leq \binom{n}{k}\binom{n-\frac{n}{k}}{k}$, $l = (k-2)\frac{n}{k}$, $v = \frac{n}{k}$, and $t = \frac{n}{k}$ and the resampling must be done without replacement [58,60,92]. Finally, for implementing the BTS, $l = n$ and $\mathcal{L}_l^r$ must be sampled with replacement from $\mathcal{D}_n$, while $\mathcal{V}_v^r$ and $\mathcal{T}_t^r$ are sampled without replacement from the sample of $\mathcal{D}_n$ that have not been sampled in $\mathcal{L}_l^r$ [60,90]. Note that for the BTS procedure $r \leq \binom{2n-1}{n}$. In this paper the BTS is exploited because it represents the state-of-the-art approach [60,90].

4.3. Feature selection

Once the CBM NPS models are built and have been confirmed to be sufficiently accurate representation of the real decays of the components, it can be interesting to investigate how these models are affected by the different features used in the model identification phase (see Table 1).

In DA this procedure is called FS or Feature Ranking [61–65]. This process allows detecting if the importance of those features, that are known to be relevant from a physical perspective, is appropriately described by the different CBM NPS models. The failure of the statistical model to properly account for the relevant features might indicate poor quality in the measurements or spurious correlations. FS therefore represents an important step of model verification, since it should generate consistent results with the available knowledge of the physical system under exam.

In addition to its use for classification purposes, the EMs can also be used to perform a very stable FS procedure. The procedure is a combination of EMs, together with the permutation test [93], in order to perform the selection and the ranking of the features. In details, for every tree, two quantities are computed: the first one is the error on the out-of-bag samples as they are used during prediction, while the second one is the error on the out-of-bag samples after a random permutation of the values of variable j. These two values are then subtracted and the average of the result over all the trees in the ensemble is the raw importance score for variable j (mean decrease in accuracy). This procedure was adopted since it can be easily carried out during the main prediction process inexpensively.

5. Results

In this section, authors report the results obtained by the different methods applied to the CBM of the main components of an NPS, in classification, and novelty detection frameworks, as described in Section 2, based on the data described in Section 3.

As described in Section 4, the first problem is a classification one where the label of the decay parameters needs to be estimated (CLASS-PROB). Secondly, authors attempted to solve the same problem in an unsupervised fashion by modelling the problem as a novelty detection one in order to further reduce the necessity of labelled data (ANOMALY-PROB).

The dataset considered in Section 3.2 was divided into training and test set, respectively D_n and $\mathcal{T}_m$, as reported in Section 4. Moreover, different dimensions of the training set $n \in \{10, 24, 55, 130, 307, 722, 1700, 4000\}$ were considered. These dimensions were derived by considering 8 values on a logarithmic scale from 10 to 4000, to analyse the behaviour of the predictive models in different conditions.

For each supervised classification ML technique, the BTS MS procedure was performed with $r = 1000$, as described in Section 4.2.3. Herebelow, the list of hyperparameters tested during the MS, with their respective intervals, is reported:

1. DNN: the set of hyperparameters is $\mathcal{H}^{DNN} = \{h_1, h_{2,1}, \cdots, h_{2,h_1}\}$ and authors chose it in $\mathfrak{H}^{\text{DNN}} = \{1, 3, 5, 7, 10\} \times \{10, 10^{1.2} \cdots, 10^3\} \times \cdots \times \{10, 10^{1.2} \cdots, 10^3\}$;
2. SNN: the set of hyperparameters is $\mathcal{H}^{SNN} = \{h_1\}$ and authors chose it in $\mathfrak{H}^{\text{SNN}} = \{1, 3, 5, 7, 10\}$;
3. ELM: the set of hyperparameters is $\mathcal{H}^{ELM} = \{h_1, h_2\}$ and authors chose it in $\mathfrak{H}^{\text{ELM}} = \{10, 10^{1.2}, \cdots, 10^3\} \times \{10^{-2}, 10^{-1.5} \cdots, 10^2\}$;
4. SVM: the set of hyperparameters is $\mathcal{H}^{SVM} = \{h_1, h_2\}$ and authors chose it in $\mathfrak{H}^{\text{SVM}} = \{10^{-2}, 10^{-1.4}, \cdots, 10^3\} \times \{10^{-2}, 10^{-1.4} \cdots, 10^3\}$;
5. KRLS: the set of hyperparameters is $\mathcal{H}^{KRLS} = \{h_1, h_2\}$ and authors chose it in $\mathfrak{H}^{\text{KRLS}} = \{10^{-2}, 10^{-1.4}, \cdots, 10^3\} \times \{10^{-2}, 10^{-1.4}, \cdots, 10^3\}$;
6. KNN: the set of hyperparameters is $\mathcal{H}^{KNN} = \{h_1\}$ and authors chose it in $\mathfrak{H}^{\text{KNN}} = \{1, 3, 7, 13, 27, 51\}$;
7. GP: the set of hyperparameters is $\mathcal{H}^{GP} = \{h_1\}$ and authors chose it in $\mathfrak{H}^{\text{GP}} = \{10^0, 10^{0.3}, \cdots, 10^3\}$;

When RF is exploited, also the FS phase is performed to understand how the data-driven model combines the different features in order to predict the decay state of each component.

Similarly to the supervised learning task, in the unsupervised case different dimensions of the training set were considered $n \in \{1500, 2000, 3000, 4000\}$ and the MS procedure was performed as follows:

1. OCSVM: the set of hyperparameters is $\mathcal{H}^{OCSVM} = \{h_1, h_2\}$ and authors chose it in $\mathfrak{H}^{\text{OCSVM}} = \{10^{-4}, 10^{-3.7}, \cdots, 10^3\} \times \{10^{-4}, 10^{-3.8}, \cdots, 10^{-1.0}\}$;
2. GKNN: the set of hyperparameters is $\mathcal{H}^{GKNN} = \{h_1\}$ and authors chose it in $\mathfrak{H}^{\text{GKNN}} = \{1, 3, 7, 13, 27, 51\}$;

The $\mathcal{V}_v^r$ cardinality was varied $v \in \{10, 20, 30\}$ with linear step, in order to test the possibility of building an efficient model with a few labelled samples. Note that, also in this case, the BTS MS procedure is adopted with $r = 1000$ and that the labels are only needed in $\mathcal{V}_v^r$ and not in $\mathcal{L}_l^r$ as described in Section 4.2.3.

The performances of each model are measured according to the metrics described in Section 4.1. Each experiment was performed 10 times in order to obtain statistical relevant result, and the t-student 95% confidence interval is reported when space in the table was available without compromising their readability.

For SNN and DNN the Python Keras library [94] has been exploited. For ELM, SVM, KRLS, KNN, and GKNN a custom R implementation has been developed. For RF the R package of Liaw and Wiener [95] has been exploited. For RFE the implementation of Blaser and Fryzlewicz [83] has been exploited. For GP the R package of Zeileis et al. [96] has been exploited. For OCSVM the R package of Meyer et al. [97] has been exploited.

5.1. CLASS-PROB

In this section, the results on the CLASS-PROB are reported. In Fig. 3 the AMR of the models learned with the different algorithms is reported, when varying n and for the four main NPS components. In Fig. 4 the AMR of the DNN (the best performing model) is reported, when varying n and for the four main NPS components.

From the different tables and figures it is possible to observe that:

- the larger is n the better performances are achieved by the learned models (see Fig. 3) and the models learned with ELM, SNN, and especially DNN generally show the best performances (see Fig. 3);
- as expected, to achieve a reasonable AMR a smaller number of samples is needed with respect to a regression-based approach not feasible in practice.

In Fig. 5 the FS phase for the four main NPS components is reported. Taking into account the problem $P15$, for each feature reported in Table 1, the mean decrease in accuracy as described in Section 4.3 is reported. From Fig. 5, it is possible to note that the RF model can adequately account for the relevant features as the outcome is consistent with the available knowle.g. [53]. According to Fig. 5, several features are always necessary to forecast the decay state of each component. As far as the PRP component is concerned, the thrust features (5 and 6), the shaft torque features (7 and 9) and the PRP torque features (24 and 25) have high predictive power. As expected, for the HLL component the thrust coefficients features (20 and 22) have the most significant predictive power. When it comes to the GTC component, the features describing the thermodynamic process have the highest predictive power, nominally GTC outlet air temperature, External pressure, and HP GT exit temperature features (16, 17 and 11). Finally, for the GT component prediction, several features are necessary, also this case is in line with engineering state-of-the-art knowledge [53]. These results indicate that, from a data driven perspective, the decay state of each component influences different phases of the NPS behaviour.

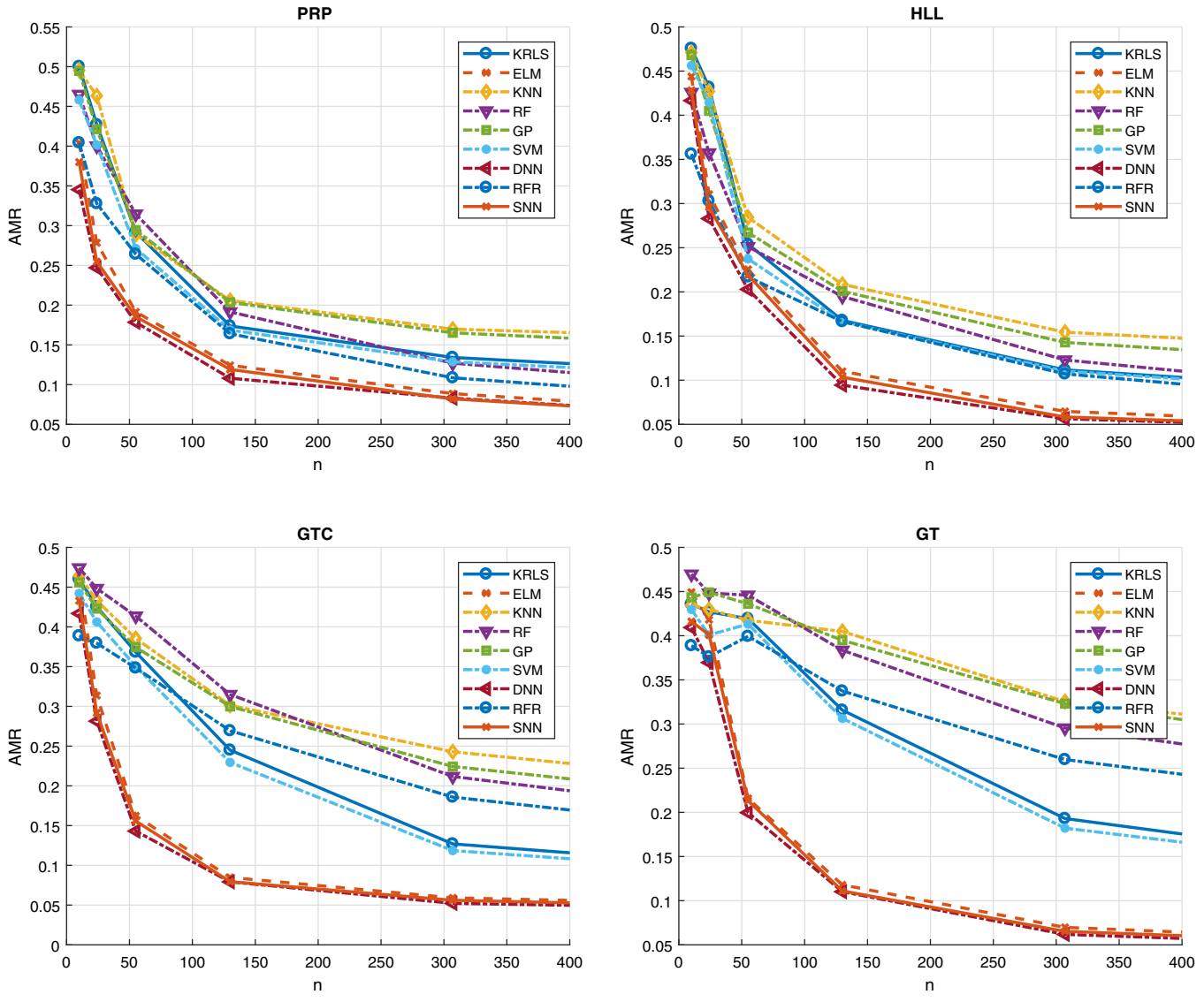

Fig. 3. CLASS-PROB: AMR of the models learned with the different algorithms when varying n and for the four main NPS components.

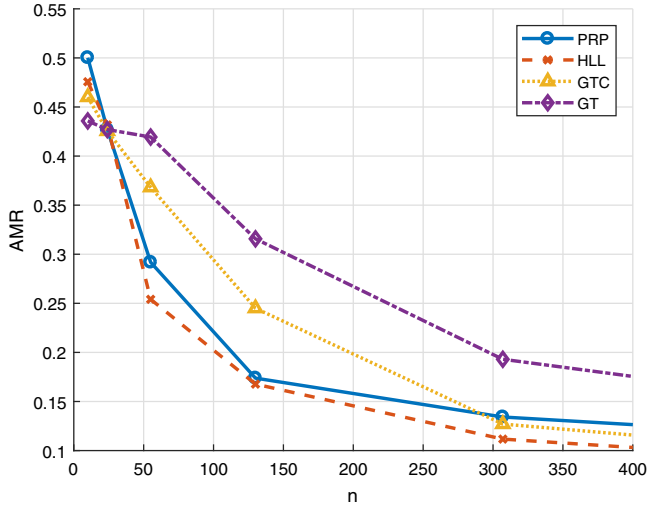

Fig. 4. CLASS-PROB: AMR of the models learned with DNN when varying n for the four main NPS components.

Fig. 5 clearly shows that the interaction between the main components cannot be easily modelled with a physical approach, considering the large number of variables that affect the final behaviour of each component. Instead, DDMs, by making use of these variables, can outperform physical models as they have the capability to take into account all the available sensors measurements to build effective and accurate predictors as reported in Fig. 3.

5.2. ANOMALY-PROB

In this section the results on the ANOMALY-PROB are reported. In Table 2, for PRP, HLL, GTC, and GT, the AMR of the models learned with the different algorithms (OCSVM and GKNN) is reported, when the number of unlabelled samples in the learning set is $l = 4000$ and when varying the number of labelled samples in the validation set $v \in \{10, 20, 30\}$ (half positively and half negatively labelled). In Table 3, respectively for PRP, HLL, GTC, and GT, the AMR of the models learned with the different algorithms is reported, when $v = 30$ and when $l \in \{1500, 2000, 3000, 4000\}$. In Table 4, for PRP, HLL, GTC, and GT, the different indexes of performances (AMR, TP, TN, FP, and FN) of the models learned with the different algorithms are reported when $n = 4000$ and $v = 30$.

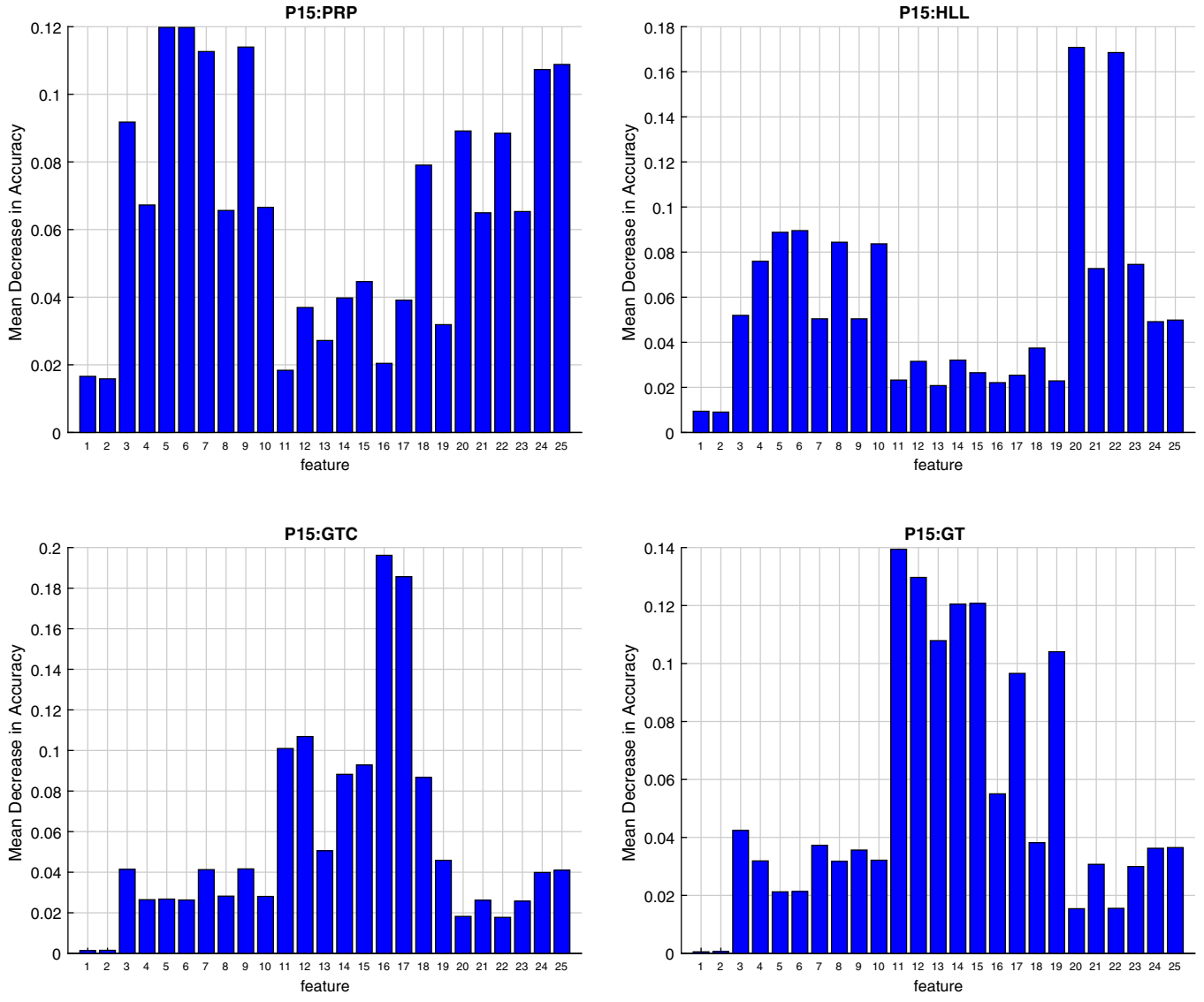

Fig. 5. CLASS-PROB: FS performed with RF for the four main NPS components.

Table 2
ANOMALY-PROB: AMR of the models learned with the different algorithms (OCSVM and GKNN) when $l = 4000$ and $v \in \{10, 20, 30\}$ and for the four main NPS components.

	v	PRP	HLL	GTC	GT
OCSVM	10	0.08 ± 0.08	0.07 ± 0.09	0.05 ± 0.07	0.11 ± 0.06
	20	0.08 ± 0.10	0.08 ± 0.07	0.10 ± 0.07	0.09 ± 0.06
	30	0.08 ± 0.07	0.12 ± 0.08	0.10 ± 0.07	0.09 ± 0.03
GKNN	10	0.07 ± 0.07	0.07 ± 0.09	0.04 ± 0.07	0.10 ± 0.06
	20	0.08 ± 0.10	0.08 ± 0.07	0.08 ± 0.07	0.07 ± 0.06
	30	0.08 ± 0.07	0.12 ± 0.08	0.08 ± 0.08	0.08 ± 0.03

Table 3
ANOMALY-PROB: AMR of the models learned with the different algorithms (OCSVM and GKNN) when $l \in \{1500, 2000, 3000, 4000\}$ and $v = 30$ and for the four main NPS components.

	n	PRP	HLL	GTC	GT
OCSVM	1500	0.09 ± 0.04	0.14 ± 0.09	0.11 ± 0.09	0.08 ± 0.03
	2000	0.12 ± 0.10	0.11 ± 0.04	0.10 ± 0.04	0.11 ± 0.13
	3000	0.08 ± 0.06	0.11 ± 0.04	0.13 ± 0.17	0.08 ± 0.05
	4000	0.08 ± 0.07	0.12 ± 0.08	0.10 ± 0.07	0.09 ± 0.03
GKNN	1500	0.09 ± 0.04	0.12 ± 0.09	0.10 ± 0.09	0.06 ± 0.03
	2000	0.11 ± 0.11	0.12 ± 0.04	0.08 ± 0.04	0.08 ± 0.13
	3000	0.07 ± 0.06	0.10 ± 0.04	0.12 ± 0.16	0.09 ± 0.04
	4000	0.08 ± 0.07	0.12 ± 0.08	0.08 ± 0.08	0.08 ± 0.03

From the tables it is possible to observe that:

- both OCSVM and GKNN perform quite well on the problem and there is no clear winner;
- changing l or v does not remarkably affect the performance of the models;
- with just few labelled samples, around 10, it is possible to obtain satisfying accuracies and this is quite a remarkable result, since 10 samples can be easily manually labelled by an expert operator;
- in some cases, the mean value of the reported AMR increases instead of decreasing when the number of training samples increases; as a fact, this value is subject to statistical variation in the data whose variance in the results can be considered acceptable from a statistical point of view. Note also that the number of validation samples is very limited and this high variance is also justified by this fact;
- FP and FN rate are quite balanced and this is a further indication of the quality of the result.

Table 4
ANOMALY-PROB: the different indexes of performances (AMR, TP, TN, FP, and FN) of the models learned with the different algorithms (OCSVM and GKNN) when $n = 4000$ and $v = 3$ and for the four main NPS components.

		PRP	HLL	GTC	GT
OCSVM	AMR	0.08 ± 0.07	0.12 ± 0.08	0.10 ± 0.07	0.09 ± 0.03
	TP	46.68 ± 8.88	45.29 ± 9.25	47.89 ± 3.17	46.59 ± 5.49
	TN	45.47 ± 3.88	42.56 ± 7.54	42.42 ± 9.36	44.56 ± 4.50
	FN	3.32 ± 8.88	4.71 ± 9.25	2.11 ± 3.17	3.41 ± 5.49
	FP	4.53 ± 3.88	7.44 ± 7.54	7.58 ± 9.36	5.44 ± 4.50
GKNN	AMR	0.08 ± 0.07	0.12 ± 0.08	0.08 ± 0.08	0.08 ± 0.03
	TP	46.68 ± 8.66	45.89 ± 9.04	48.04 ± 3.16	47.42 ± 5.41
	TN	45.24 ± 3.86	42.30 ± 7.61	44.14 ± 9.55	44.49 ± 4.47
	FN	3.32 ± 8.84	4.11 ± 9.21	1.96 ± 3.13	2.58 ± 5.54
	FP	4.76 ± 3.87	7.70 ± 7.54	5.86 ± 9.29	5.51 ± 4.56

6. Conclusions

The maintenance of the several components of a Ship Propulsion Systems is an onerous activity, which needs to be efficiently programmed by a shipbuilding company to save time and money. The replacement policies of these components can be planned in a Condition-Based fashion, by predicting their decay state and thus proceed to substitution only when needed. In this paper, authors proposed several Data Analysis supervised and unsupervised techniques for the Condition-Based Maintenance of a naval vessel, characterised by a combined diesel-electric and gas propulsion plant. The propulsion plant has been modelled using the state-of-the-art simulation techniques available in the literature [44,47]. The dataset used to benchmark the proposed data-driven approaches has been created using a realistic simulator of Frigate validated and fine-tuned during sea trials. The model has been designed to work in calm water scenario, and measurement uncertainties have not been taken into account. Within the mentioned limitations of the numerical model, the authors are confident that the results shown are in line with the real behaviour of the system. The authors considered the case in which GTC, GT, HLL, and PRP NPS components decay at the same time, to provide a realistic simulation environment.

The proposed analysis considered contexts where the collection of vast amounts of labelled data containing the exact decay state of the components is unfeasible. In fact, the collection of labelled data requires a drydocking of the ship and the intervention of expert operators, which is usually an infrequent event. As a result, authors focused on methods which could allow only a minimal feedback from naval specialists, thus simplifying the dataset collection phase. In particular, supervised Data Analysis techniques allowed to reach an average percentage error of 1% ± 2% adopting 4000 labelled samples respectively for all PRP, HLL, GTC, and GT. On the other hand, the non-supervised techniques exploited could reach an average value of 8% ± 5% error adopting only 10 labelled data overall. To reach the same average error percentage, the obtained supervised models required at least a number of labelled data between 130 and 307, thus requiring a higher amount of information for learning a performing model. Clearly, supervised models could reach a lower error rate with respect to unsupervised ones (1% ± 2%), adopting a higher amount of labelling training data, but such a difference in accuracy is obtained through higher costs for dataset collection, which cannot be sustained in most cases.

In conclusion, this study proved that it is possible to treat a Condition-Based Maintenance problem in an unsupervised fashion, with results close to the ones obtained with supervised techniques present in literature. These models can be adopted for real-time applications directly on-board, to easily and quickly identify maintenance necessities.

Appendix A. CLASS-PROB extended results

In this section, the results for the CLASS-PROB, from which Figs. 3 and 4 were derived, are extensively reported. In Table A.5, re-

Table A.5
CLASS-PROB: AMR of the models learned with the different algorithms (DNN, SNN, ELM, SVM, KRLS, KNN, and GP) when varying n for the four main NPS components.

	n	PRP	HLL	GTC	GT
KRLS	10	0.50 ± 0.06	0.48 ± 0.06	0.46 ± 0.10	0.44 ± 0.09
	24	0.43 ± 0.10	0.43 ± 0.10	0.43 ± 0.07	0.43 ± 0.07
	55	0.29 ± 0.06	0.25 ± 0.07	0.37 ± 0.06	0.42 ± 0.08
	130	0.17 ± 0.05	0.17 ± 0.06	0.25 ± 0.12	0.32 ± 0.09
	307	0.13 ± 0.02	0.11 ± 0.02	0.13 ± 0.03	0.19 ± 0.03
	722	0.10 ± 0.02	0.07 ± 0.02	0.08 ± 0.03	0.12 ± 0.03
	1700	0.06 ± 0.01	0.04 ± 0.02	0.05 ± 0.02	0.07 ± 0.02
	4000	0.03 ± 0.01	0.02 ± 0.01	0.04 ± 0.02	0.05 ± 0.01
ELM	10	0.40 ± 0.22	0.44 ± 0.13	0.46 ± 0.16	0.45 ± 0.09
	24	0.28 ± 0.14	0.31 ± 0.18	0.31 ± 0.20	0.42 ± 0.16
	55	0.19 ± 0.08	0.23 ± 0.09	0.16 ± 0.07	0.22 ± 0.07
	130	0.12 ± 0.05	0.11 ± 0.05	0.08 ± 0.03	0.12 ± 0.03
	307	0.09 ± 0.04	0.06 ± 0.03	0.06 ± 0.02	0.07 ± 0.01
	722	0.05 ± 0.04	0.04 ± 0.02	0.05 ± 0.02	0.05 ± 0.02
	1700	0.03 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	0.03 ± 0.01
	4000	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.02 ± 0.02
KNN	10	0.50 ± 0.07	0.47 ± 0.06	0.46 ± 0.10	0.43 ± 0.08
	24	0.46 ± 0.08	0.43 ± 0.15	0.43 ± 0.08	0.43 ± 0.07
	55	0.29 ± 0.07	0.28 ± 0.12	0.39 ± 0.05	0.42 ± 0.08
	130	0.21 ± 0.05	0.21 ± 0.06	0.30 ± 0.03	0.40 ± 0.07
	307	0.17 ± 0.02	0.15 ± 0.02	0.24 ± 0.04	0.33 ± 0.04
	722	0.15 ± 0.03	0.12 ± 0.02	0.18 ± 0.03	0.26 ± 0.02
	1700	0.13 ± 0.02	0.10 ± 0.02	0.13 ± 0.02	0.20 ± 0.02
	4000	0.09 ± 0.01	0.07 ± 0.01	0.10 ± 0.02	0.15 ± 0.02
RF	10	0.47 ± 0.07	0.43 ± 0.12	0.47 ± 0.09	0.47 ± 0.07
	24	0.40 ± 0.12	0.36 ± 0.15	0.45 ± 0.07	0.45 ± 0.04
	55	0.31 ± 0.08	0.25 ± 0.06	0.41 ± 0.04	0.45 ± 0.03
	130	0.19 ± 0.04	0.19 ± 0.05	0.31 ± 0.06	0.38 ± 0.04
	307	0.13 ± 0.04	0.12 ± 0.04	0.21 ± 0.05	0.30 ± 0.04
	722	0.07 ± 0.02	0.07 ± 0.02	0.13 ± 0.01	0.21 ± 0.03
	1700	0.04 ± 0.02	0.04 ± 0.01	0.09 ± 0.02	0.13 ± 0.02
	4000	0.02 ± 0.01	0.02 ± 0.01	0.06 ± 0.02	0.08 ± 0.02
GP	10	0.49 ± 0.05	0.47 ± 0.07	0.46 ± 0.07	0.44 ± 0.05
	24	0.42 ± 0.11	0.40 ± 0.11	0.42 ± 0.07	0.45 ± 0.09
	55	0.29 ± 0.05	0.27 ± 0.05	0.37 ± 0.04	0.44 ± 0.07
	130	0.20 ± 0.05	0.20 ± 0.03	0.30 ± 0.03	0.39 ± 0.08
	307	0.17 ± 0.02	0.14 ± 0.03	0.22 ± 0.02	0.32 ± 0.06
	722	0.13 ± 0.03	0.11 ± 0.03	0.15 ± 0.03	0.24 ± 0.04
	1700	0.11 ± 0.02	0.07 ± 0.02	0.10 ± 0.03	0.17 ± 0.03
	4000	0.08 ± 0.01	0.04 ± 0.01	0.07 ± 0.02	0.11 ± 0.02
SVM	10	0.46 ± 0.06	0.46 ± 0.07	0.44 ± 0.11	0.43 ± 0.10
	24	0.40 ± 0.11	0.41 ± 0.11	0.41 ± 0.07	0.40 ± 0.08
	55	0.27 ± 0.06	0.24 ± 0.07	0.35 ± 0.07	0.41 ± 0.09
	130	0.17 ± 0.06	0.17 ± 0.07	0.23 ± 0.14	0.31 ± 0.10
	307	0.13 ± 0.03	0.11 ± 0.02	0.12 ± 0.03	0.18 ± 0.04
	722	0.10 ± 0.02	0.07 ± 0.02	0.07 ± 0.03	0.11 ± 0.03
	1700	0.06 ± 0.02	0.04 ± 0.02	0.05 ± 0.02	0.07 ± 0.02
	4000	0.03 ± 0.01	0.02 ± 0.01	0.03 ± 0.02	0.05 ± 0.01
DNN	10	0.35 ± 0.12	0.42 ± 0.08	0.42 ± 0.09	0.41 ± 0.05
	24	0.25 ± 0.08	0.28 ± 0.10	0.28 ± 0.11	0.37 ± 0.09
	55	0.18 ± 0.05	0.20 ± 0.05	0.14 ± 0.04	0.20 ± 0.04
	130	0.11 ± 0.03	0.09 ± 0.02	0.08 ± 0.02	0.11 ± 0.02
	307	0.08 ± 0.02	0.06 ± 0.02	0.05 ± 0.01	0.06 ± 0.01
	722	0.04 ± 0.02	0.04 ± 0.01	0.04 ± 0.01	0.04 ± 0.01
	1700	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.03 ± 0.01
	4000	0.01 ± 0.00	0.01 ± 0.01	0.01 ± 0.00	0.02 ± 0.01
RFR	10	0.40 ± 0.04	0.36 ± 0.07	0.39 ± 0.05	0.39 ± 0.04
	24	0.33 ± 0.06	0.30 ± 0.08	0.38 ± 0.04	0.38 ± 0.02
	55	0.26 ± 0.04	0.22 ± 0.03	0.35 ± 0.02	0.40 ± 0.02
	130	0.16 ± 0.02	0.17 ± 0.03	0.27 ± 0.03	0.34 ± 0.02
	307	0.11 ± 0.02	0.11 ± 0.02	0.19 ± 0.03	0.26 ± 0.02
	722	0.06 ± 0.01	0.06 ± 0.01	0.11 ± 0.01	0.19 ± 0.02
	1700	0.04 ± 0.01	0.03 ± 0.01	0.07 ± 0.01	0.11 ± 0.01
	4000	0.02 ± 0.01	0.02 ± 0.01	0.05 ± 0.01	0.07 ± 0.01
SNN	10	0.38 ± 0.12	0.43 ± 0.08	0.43 ± 0.09	0.42 ± 0.05
	24	0.25 ± 0.08	0.30 ± 0.10	0.29 ± 0.10	0.40 ± 0.09
	55	0.19 ± 0.05	0.22 ± 0.05	0.16 ± 0.04	0.21 ± 0.04
	130	0.12 ± 0.03	0.10 ± 0.02	0.08 ± 0.02	0.11 ± 0.02
	307	0.08 ± 0.02	0.06 ± 0.02	0.06 ± 0.01	0.07 ± 0.01
	722	0.04 ± 0.02	0.04 ± 0.01	0.04 ± 0.01	0.04 ± 0.01
	1700	0.02 ± 0.01	0.02 ± 0.01	0.03 ± 0.01	0.03 ± 0.01
	4000	0.01 ± 0.00	0.01 ± 0.01	0.01 ± 0.00	0.02 ± 0.01

Table A.6
CLASS-PROB: the different indexes of performances (AMR, TP, TN, FP, and FN) of the models learned with the different algorithms (DNN, SNN, ELM, SVM, KRLS, KNN, and GP) when n is the largest possible for the four main NPS components.

		PRP	HLL	GTC	GT
KRLS	AMR	0.03 ± 0.01	0.02 ± 0.01	0.04 ± 0.01	0.05 ± 0.00
	TP	45.11 ± 1.10	45.68 ± 1.79	58.06 ± 1.67	57.29 ± 1.37
	TN	51.72 ± 1.02	52.33 ± 1.97	38.31 ± 1.45	37.77 ± 1.50
	FN	1.64 ± 0.39	0.84 ± 0.28	1.96 ± 0.74	2.81 ± 0.55
	FP	1.53 ± 0.47	1.15 ± 0.35	1.67 ± 0.49	2.13 ± 0.37
ELM	AMR	0.01 ± 0.00	0.01 ± 0.01	0.01 ± 0.00	0.02 ± 0.01
	TP	45.05 ± 1.87	46.32 ± 0.85	60.18 ± 1.14	57.51 ± 1.80
	TN	53.60 ± 1.67	52.22 ± 1.08	38.33 ± 1.23	40.20 ± 1.53
	FN	0.80 ± 0.31	0.64 ± 0.30	0.73 ± 0.24	1.94 ± 0.75
	FP	0.55 ± 0.22	0.82 ± 0.41	0.76 ± 0.31	0.35 ± 0.27
KNN	AMR	0.09 ± 0.01	0.07 ± 0.01	0.10 ± 0.01	0.15 ± 0.01
	TP	42.06 ± 0.63	43.10 ± 1.75	55.23 ± 1.68	52.65 ± 1.70
	TN	48.46 ± 1.03	49.56 ± 1.93	34.92 ± 1.53	31.96 ± 1.81
	FN	4.69 ± 0.51	3.42 ± 0.53	4.79 ± 0.71	7.45 ± 0.68
	FP	4.79 ± 0.59	3.92 ± 0.40	5.06 ± 0.79	7.94 ± 0.82
RF	AMR	0.02 ± 0.01	0.02 ± 0.01	0.06 ± 0.01	0.08 ± 0.01
	TP	45.81 ± 1.90	45.70 ± 1.47	57.17 ± 1.01	56.31 ± 1.53
	TN	52.03 ± 1.68	51.89 ± 1.26	37.31 ± 1.01	35.78 ± 1.61
	FN	1.23 ± 0.42	1.29 ± 0.37	2.52 ± 0.39	3.50 ± 0.77
	FP	0.93 ± 0.36	1.12 ± 0.48	3.00 ± 0.73	4.41 ± 0.73
GP	AMR	0.08 ± 0.01	0.04 ± 0.01	0.07 ± 0.01	0.11 ± 0.01
	TP	42.33 ± 0.84	44.17 ± 1.94	57.43 ± 1.55	57.07 ± 1.51
	TN	49.99 ± 0.91	51.55 ± 1.90	35.71 ± 1.32	32.39 ± 1.45
	FN	4.42 ± 0.67	2.35 ± 0.85	2.59 ± 0.74	3.03 ± 0.63
	FP	3.26 ± 0.86	1.93 ± 0.51	4.27 ± 0.73	7.51 ± 1.03
SVM	AMR	0.03 ± 0.01	0.02 ± 0.01	0.03 ± 0.01	0.05 ± 0.00
	TP	45.15 ± 1.21	45.70 ± 1.95	58.16 ± 1.90	57.36 ± 1.48
	TN	51.79 ± 1.13	52.43 ± 2.15	38.37 ± 1.57	37.95 ± 1.65
	FN	1.60 ± 0.42	0.82 ± 0.30	1.86 ± 0.83	2.74 ± 0.59
	FP	1.46 ± 0.53	1.05 ± 0.39	1.61 ± 0.53	1.95 ± 0.39
DNN	AMR	0.01 ± 0.00	0.01 ± 0.01	0.01 ± 0.00	0.02 ± 0.01
	TP	45.10 ± 2.08	46.40 ± 0.96	60.25 ± 1.20	57.65 ± 2.02
	TN	53.63 ± 1.83	52.30 ± 1.23	38.40 ± 1.38	40.24 ± 1.75
	FN	0.75 ± 0.35	0.56 ± 0.34	0.66 ± 0.26	1.80 ± 0.85
	FP	0.52 ± 0.25	0.74 ± 0.45	0.69 ± 0.34	0.31 ± 0.31
RFR	AMR	0.02 ± 0.01	0.02 ± 0.01	0.05 ± 0.01	0.07 ± 0.01
	TP	46.05 ± 2.12	45.96 ± 1.63	57.56 ± 1.08	56.68 ± 1.64
	TN	52.21 ± 1.88	52.05 ± 1.43	37.70 ± 1.07	36.45 ± 1.72
	FN	0.99 ± 0.48	1.03 ± 0.39	2.13 ± 0.45	3.13 ± 0.88
	FP	0.75 ± 0.40	0.96 ± 0.53	2.61 ± 0.83	3.74 ± 0.78
SNN	AMR	0.01 ± 0.00	0.01 ± 0.01	0.01 ± 0.00	0.02 ± 0.01
	TP	45.10 ± 2.00	46.33 ± 0.93	60.19 ± 1.25	57.56 ± 1.89
	TN	53.64 ± 1.90	52.25 ± 1.17	38.36 ± 1.41	40.20 ± 1.63
	FN	0.75 ± 0.34	0.63 ± 0.32	0.72 ± 0.28	1.89 ± 0.79
	FP	0.51 ± 0.25	0.79 ± 0.47	0.73 ± 0.34	0.35 ± 0.29

spectively for PRP, HLL, GTC, and GT, the AMR of the models learned with the different algorithms (DNN, SNN, ELM, SVM, KRLS, KNN, and GP) is reported, when varying n. In Table A.6, instead, respectively for PRP, HLL, GTC, and GT, the different indexes of performances (AMR, TP, TN, FP, and FN) of the models learned with the different algorithms are reported, when n is the largest possible.

References

[1] Ahmad R, Kamaruddin S. An overview of time-based and condition-based maintenance in industrial application. Comput Indus Eng 2012;63(1):135–49.
[2] Pinjala SK, Pintelon L, Vereecke A. An empirical investigation on the relationship between business and maintenance strategies. Int J Prod Econ 2006;104(1):214–29.
[3] Tsang AHC. Condition-based maintenance: tools and decision making. J Quality Maint Eng 1995;1(3):3–17.
[4] Dekker R, Scarf PA. On the impact of optimisation models in maintenance decision making: the state of the art. Reliab Eng Syst Saf 1998;60(2):111–19.
[5] Takata S, Kirnura F, van Houten FAM, Westkamper E, Shpitalni M, Ceglarek D, et al. Maintenance: changing role in life cycle management. CIRP Ann-Manuf Technol 2004;53(2):643–55.
[6] Frangopol D, Dan M, Bocchini D, Decò A, Kim D, Kwon D, et al. Integrated life-cycle framework for maintenance, monitoring, and reliability of naval ship structures. Nav Eng J 2012;124(1):89–99.
[7] Peng Y, Dong M, Zuo MJ. Current status of machine prognostics in condition-based maintenance: a review. Int J Adv Manuf Technol 2010;50(1–4):297–313.
[8] Kyrtatos NP, Tzanos E, Coustas J, Vastarouhas D, Rizos E. Shipboard engine performance assessment by comparing actual measured data to nominal values produced by detailed engine simulations. In: Proceedings; 2010.
[9] Goossens AJM, Basten RJI. Exploring maintenance policy selection using the analytic hierarchy process; an application for naval ships. Reliab Eng Syst Saf 2015;142:31–41.
[10] Widodo A, Yang B. Support vector machine in machine condition monitoring and fault diagnosis. Mech Syst Signal Process 2007;21(6):2560–74.
[11] Mobley RK. An introduction to predictive maintenance. Butterworth-Heinemann; 2002.
[12] Management GBQ, Committee SS, Institution BS. British standard glossary of maintenance management terms in terotechnology. British Standards Institution; 1984.
[13] Budai-Balke G. Operations research models for scheduling railway infrastructure maintenance. Rozenberg Publishers; 2009.
[14] Abu-Elanien AEB, Salama MMA. Asset management techniques for transformers. Electr Power Syst Res 2010;80(4):456–64.
[15] Kothamasu R, Huang SH. Adaptive mamdani fuzzy model for condition-based maintenance. Fuzzy Sets Syst 2007;158(24):2715–33.
[16] Selvik JT, Aven T. A framework for reliability and risk centered maintenance. Reliab Eng Syst Saf 2011;96(2):324–31.
[17] Mann L, Saxena A, Knapp GM. Statistical-based or condition-based preventive maintenance? J Quality Maint Eng 1995;1(1):46–59.
[18] ISO BS. 13372: 2012, condition monitoring and diagnostics of machines-vocabulary. British standards; 2004.
[19] Ghasemi A, Yacout S, Ouali MS. Parameter estimation methods for condition-based maintenance with indirect observations. IEEE Trans Reliab 2010;59(2):426–39.
[20] Jardine AKS, Banjevic D, Wiseman M, Buck S, Joseph T. Optimizing a mine haul truck wheel motors? condition monitoring program use of proportional hazards modeling. J Quality Maint Eng 2001;7(4):286–302.
[21] Linoff GS, Berry MJA. Data mining techniques: for marketing, sales, and customer relationship management. John Wiley & Sons; 2011.
[22] Vapnik VN. Statistical learning theory. Wiley New York; 1998.
[23] Györfi L, Kohler M, Krzyzak A, Walk H. A distribution-free theory of nonparametric regression. Springer Science & Business Media; 2006.
[24] Newton LR. Data-logging in practical science: research and reality. Int J Sci Educ 2000;22(12):1247–59.
[25] Jardine AKS, Lin D, Banjevic D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mech Syst Signal Process 2006;20(7):1483–510.
[26] Poyhonen S, Jover P, Hyotyniemi H. Signal processing of vibrations for condition monitoring of an induction motor. Symposium on control, communications and signal processing; 2004.
[27] Bunks C, McCarthy D, Al-Ani T. Condition-based maintenance of machines using hidden markov models. Mech Syst Signal Process 2000;14(4):597–612.
[28] Simani S, Fantuzzi C, Patton R. Model-based fault diagnosis in dynamic systems using identification techniques. Springer-Verlag London; 2003.
[29] Palmé T, Breuhaus P, Assadi M, Klein A, Kim M. New alstom monitoring tools leveraging artificial neural network technologies. Turbo expo: turbine technical conference and exposition; 2011.
[30] Singer RM, Gross KC, King RW. A pattern-recognition-based, fault-tolerant monitoring and diagnostic technique. Argonne national lab., Idaho Falls, ID (United States); 1995.
[31] Coraddu A, Oneto L, Ghio A, Savio S, Anguita D, Figari M. Machine learning approaches for improving condition-based maintenance of naval propulsion plants. Proc Inst Mech Eng Part M 2016;230(1):136–53.
[32] Akinfiev TS, Armada MA, Fernandez R. Nondestructive testing of the state of a ship's hull with an underwater robot. Russ J Nondestr Test 2008;44(9):626–33.
[33] Bagavathiappan S, Lahiri BB, Saravanan T, Philip J, Jayakumar T. Infrared thermography for condition monitoring-a review. Infrared Phys Technol 2013;60:35–55.
[34] Basurko OC, Uriondo Z. Condition-based maintenance for medium speed diesel engines used in vessels in operation. Appl Therm Eng 2015;80:404–12.
[35] Lazakis I, Dikis K, Michala AL, Theotokatos G. Advanced ship systems condition monitoring for enhanced inspection, maintenance and decision making in ship operations. Transp Res Procedia 2016;14:1679–88.
[36] Shawe-Taylor J, Cristianini N. Kernel methods for pattern analysis. Cambridge university press; 2004.
[37] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 1958;65(6):386.
[38] Rasmussen CE. Gaussian processes for machine learning. Gaussian processes for machine learning; 2006.
[39] Duch W. Similarity-based methods: a general framework for classification, approximation and association. Control Cybern 2000;29(4).
[40] Breiman L. Random forests. Mach Learn 2001;45(1):5–32.
[41] Markou M, Singh S. Novelty detection: a review. Signal Process 2003;83(12):2481–97.
[42] Swersky L, Marques HO, Sander J, Campello RJ, Zimek A. On the evaluation of outlier detection and one-class classification methods. IEEE international conference on data science and advanced analytics; 2016.
[43] Vanschoren J, van Rijn JN, Bischl B, Torgo L. Openml: networked science in machine learning. SIGKDD Explor 2013;15(2):49–60.
[44] Altosole M, Benvenuto G, Figari M, Campora U. Real-time simulation of a cogag naval ship propulsion system. J Eng Maritime Environ 2009;223(1):47–62.
[45] Tarabrin AP, Schurovsky VA, Bodrov AI, Stalder JP. An analysis of axial compressor fouling and a blade cleaning method. J Turbomach 1998;120(2):256–61.

[46] Meher-Homji CB, Focke AB, Wooldridge MB. Fouling of axial flow compressors - causes, effects, detection, and control. Eighteenth turbomachinery symposium; 1989.
[47] Altosole M, Campora U, Martelli M, Figari M. Performance decay analysis of a marine gas turbine propulsion system. J Ship Res 2014;58(3):117–29.
[48] Khor YS, Xiao Q. Cfd simulations of the effects of fouling and antifouling. Ocean Eng 2011;38(10):1065–79.
[49] Atlar M, Glover EJ, Candries M, Mutton RJ, Anderson CD. The effect of a foul release coating on propeller performance. Marine science and technology for environmental sustainability; 2002.
[50] Wan B, Nishikswa E, Uchida M. The experiment and numerical calculation of propeller performance with surface roughness effects. J Kansai Soc Naval Architects, Japan 2002;2002(238):49–54.
[51] Lindholdt A, Dam-Johansen K, Olsen SM, Yebra DM, Kiil S. Effects of biofouling development on drag forces of hull coatings for ocean-going ships: a review. J Coat Technol Res 2015;12(3):315–444.
[52] Hadler JB, Wilson CJ, Beal AL. Ship standardisation trial performance and correlation with model predictions. The society of naval architects and marine engineers; 1962.
[53] Cipollini F, Oneto L, Coraddu A, Murphy A, Anguita D. Condition-based maintenance of naval propulsion systems with supervised data analysis. Ocean Eng 2018;149:268–78.
[54] Hawkins DM. Identification of outliers. Springer; 1980.
[55] Hastie T, Tibshirani R, Friedman J. Unsupervised learning. The elements of statistical learning; 2009.
[56] Rosasco L, De Vito E, Caponnetto A, Piana M, Verri A. Are loss functions all the same? Neural Comput 2004;16(5):1063–76.
[57] Powers DM. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. J Mach Learn Technol 2011;2(1):37–63.
[58] Kohavi R, et al. A study of cross-validation and bootstrap for accuracy estimation and model selection International joint conference on artificial intelligence; 1995.
[59] Bartlett PL, Boucheron S, Lugosi G. Model selection and error estimation. Mach Learn 2002;48(1–3):85–113.
[60] Anguita D, Ghio A, Oneto L, Ridella S. In-sample and out-of-sample model selection and error estimation for support vector machines. IEEE Trans Neural Netw Learn Syst 2012;23(9):1390–406.
[61] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res 2003;3:1157–82.
[62] Friedman J, Hastie T, Tibshirani R. The elements of statistical learning. Springer series in statistics Springer, Berlin; 2001.
[63] Chang YW, Lin CJ. Feature ranking using linear svm. WCCI causation and prediction challenge; 2008.
[64] Yoon H, Yang K, Shahabi C. Feature subset selection and feature ranking for multivariate time series. IEEE Trans Knowl Data Eng 2005;17(9):1186–98.
[65] Hong SJ. Use of contextual information for feature ranking and discretization. IEEE Trans Knowl Data Eng 1997;9(5):718–30.
[66] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Cognit Model 1988;5(3):1.
[67] Cybenko G. Approximation by superpositions of a sigmoidal function. Math Control Signals Syst (MCSS) 1989;2(4):303–14.
[68] Bishop CM. Neural networks for pattern recognition. Oxford University Press; 1995.
[69] Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural Comput 2006;18(7):1527–54.
[70] Bengio Y. Learning deep architectures for ai. Found Trends® Mach Learn 2009;2(1):1–127.
[71] Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 2013;35(8):1798–828.
[72] Huang G, Huang G-B, Song S, You K. Trends in extreme learning machines: a review. Neural Netw 2015;61:32–48.
[73] Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications. Neurocomputing 2006;70(1):489–501.
[74] Cambria E, Huang G-B. Extreme learning machines. IEEE Intell Syst 2013;28(6):30–59.
[75] Huang G-B, Zhu Q-Y, Siew C-K. Extreme learning machine: a new learning scheme of feedforward neural networks. IEEE international joint conference on neural networks; 2004.
[76] Cristianini N, Shawe-Taylor J. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press; 2000.
[77] Scholkopf B. The kernel trick for distances. Advances in neural information processing systems; 2001.
[78] Rifkin R, Yeo G, Poggio T. Regularized least-squares classification. Nato Sci Ser Sub Ser III Comput Syst Sci 2003;190:131–54.
[79] Lee WS, Bartlett PL, Williamson RC. The importance of convexity in learning with squared loss. IEEE Trans Inf Theory 1998;44(5):1974–80.
[80] Keerthi SS, Lin CJ. Asymptotic behaviors of support vector machines with gaussian kernel. Neural Comput 2003;15(7):1667–89.
[81] Oneto L, Ghio A, Ridella S, Anguita D. Support vector machines and strictly positive definite kernel: The regularization hyperparameter is more important than the kernel hyperparameters. IEEE international joint conference on neural networks; 2015.
[82] Germain P, Lacasse A, Laviolette F, Marchand M, Roy JF. Risk bounds for the majority vote: from a pac-bayesian analysis to a learning algorithm. J Mach Learn Res 2015;16(1):787–860.
[83] Blaser R, Fryzlewicz P. Random rotation ensembles. J Mach Learn Res 2015;2:1–15.
[84] Fernández-Delgado M, Cernadas E, Barro S, Amorim D. Do we need hundreds of classifiers to solve real world classification problems. J Mach Learn Res 2014;15(1):3133–81.
[85] Wainberg M, Alipanahi B, Frey BJ. Are random forests truly the best classifiers? J Mach Learn Res 2016;17(110):1–5.
[86] Orlandi I, Oneto L, Anguita D. Random forests model selection. European symposium on artificial neural networks, computational intelligence and machine learning; 2016.
[87] Gelman A, Carlin JB, Stern HS, Rubin DB. Bayesian data analysis, 2. Chapman and Hall/CRC; 2014.
[88] Cover T, Hart P. Nearest neighbor pattern classification. IEEE Trans Inf Theory 1967;13(1):21–7.
[89] Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data sets. ACM sigmod record; 2000.
[90] Efron B, Tibshirani RJ. An introduction to the bootstrap. CRC press; 1994.
[91] Anguita D, Boni A, Ridella S. Evaluating the generalization ability of support vector machines through the bootstrap. Neural Process Lett 2000;11(1):51–8.
[92] Arlot S, Celisse A. A survey of cross-validation procedures for model selection. Stat Surv 2010;4:40–79.
[93] Good P. Permutation tests: a practical guide to resampling methods for testing hypotheses. Springer Science & Business Media; 2013.
[94] Chollet F. Keras. https://keras.io; 2015.
[95] Liaw A, Wiener M. Classification and regression by randomforest. R News 2002;2(3):18–22.
[96] Zeileis A, Hornik K, Smola A, Karatzoglou A. Kernlab-an s4 package for kernel methods in r. J Stat Softw 2004;11(9):1–20.
[97] Meyer D., Dimitriadou E., Hornik K., Weingessel A., Leisch F. e1071: Misc functions of the department of statistics, probability theory group (formerly: E1071), tu wien. R package; 2015. R package version 1.6–7.