Condition-Based Maintenance of Naval Propulsion Systems with Supervised Data Analysis.

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

[a]*DIBRIS - University of Genova, Via Opera Pia 13, I-16145 Genova, Italy*
[b]*Marine, Offshore and Subsea Technology Group, School of Engineering*
Newcastle University, Newcastle upon Tyne, NE1 7RU, UK

Abstract

The behavior and interaction of the main components of Ship Propulsion Systems cannot be easily modeled with a priori physical knowledge, considering the large amount of variables influencing them. Data-Driven Models (DDMs), instead, exploit advanced statistical techniques to build models directly on the large amount of historical data collected by on-board automation systems, without requiring any a priori knowledge. DDMs are extremely useful when it comes to continuously monitoring the propulsion equipment and take decisions based on the actual condition of the propulsion plant. In this paper, the authors investigate the problem of performing Condition-Based Maintenance through the use of DDMs. In order to conceive this purpose, several state-of-the-art supervised learning techniques are adopted, which require labeled sensor data in order to be deployed. A naval vessel, characterized by a combined diesel-electric and gas propulsion plant, has been exploited to collect such data and show the effectiveness of the proposed approaches. Because of confidentiality constraints with the Navy the authors used a real-data validated simulator and the dataset has been published for free use through the UCI repository.

Keywords: Data Analysis, Naval Propulsion Systems, Condition-Based

[*] Corresponding Author

Email addresses: francesca.cipollini@edu.unige.it (Francesca Cipollini), luca.oneto@unige.it (Luca Oneto), andrea.coraddu@newcastle.ac.uk (Andrea Coraddu), a.j.murphy@newcastle.ac.uk (Alan John Murphy), davide.anguita@unige.it (Davide Anguita)

Maintenance, Supervised Learning.

1. Introduction

Data Analysis (DA) is improving our ability to understand complex phenomena much more rapidly than a priori physical models have done in the past (Anguita et al., 2010; Boucheron et al., 2005; Coraddu et al., 2017). Real-world systems are usually very complex, and hard to model only relying on the a priori knowledge of the problem (Witten et al., 2016; Peng et al., 2010). On the contrary, Data-Driven Models (DDMs) are built exploiting advanced statistical techniques together with the historical data produced and stored by the logging and monitoring apparatus, without requiring any a priori knowledge of the underlining physical phenomena (Vapnik, 1998; Györfi et al., 2006). These characteristics make DDMs a suitable solution in all those contexts where large amount of historical data is available, such as manufacturing (Peng et al., 2010), communications (Nguyen & Armitage, 2008), finance (Shin et al., 2005), healthcare (Mannini & Sabatini, 2010), social networks (Pang et al., 2002), commerce (Das & Chen, 2001), and transportation (Petersen et al., 2012; Budai et al., 2006; Smith et al., 2013).

Many of these sectors traditionally based their profit on empirical experience of sector specialists, and on simplified models built upon a priori knowledge of the specific problems (Waeyenbergh & Pintelon, 2002; Peng et al., 2010). This approach obviously requires a significant amount of time and experience. At the same time, during the last decades, production plants and products have been equipped with many sensors for different purposes: automation, quality check, monitoring, and logging. The results of this process is the availability of a huge amount of historical and real-time data (Linoff & Berry, 2011). Recently, industries have realized that these data, despite their management costs, can be considered as an opportunity to improve their business since historical information can be adopted to create new services or improve the quality of their products (Linoff & Berry, 2011; Oneto et al., 2016b). In particular, they can leverage this huge amount of data thanks to the DMMs which can rapidly and effectively extract useful and actionable information (Witten et al., 2016).

In the shipbuilding industry, one of the main objectives of shipwrights companies is to improve the technological quality of their products. For example, they design more efficient hull shapes and propeller geometries, study innovative propulsion systems, and reduce the overall production costs (Carlton, 2011). Recently, many of these companies are evaluating different DA solutions for improving the quality of their products, for monitoring the equipment, and for maintenance purposes as integrative activities to their core business (Waeyenbergh & Pintelon, 2002; Wang et al., 2015). In fact, ships are already equipped with a network of sensors that collect data for security, diagnostic and monitoring purposes, which DA can directly exploit by taking advantage of these technologies (Waeyenbergh & Pintelon, 2002; Petersen et al., 2012). DA, for example, offers the possibility to extract, from the raw sensor data, useful information about the efficiency of the ship (Smith et al., 2013), to reduce the fuel consumption (Coraddu et al., 2015), and to improve maintenance activities (Coraddu et al., 2016). These data represent strategic information for shipyards, operators, ship owners, and crews, since they can be used for advisory, control, and fault detection purposes (Oneto et al., 2016a).

Among the different problems, maintenance is probably the most critical one since it could require drydocking, and the cost of retrieving a stricken vessel offshore is non-trivial (Widodo & Yang, 2007; Mobley, 2002). Correct maintenance ensures that a ship works as it was designed, with the desired level performances, without impacting the service (Management et al., 1984). Maintenance policies can be divided into two main categories (Budai-Balke, 2009; Abu-Elanien & Salama, 2010): Corrective (CM), and Preventive (PM).

In the past, the most common approach to this problem relied on the CM where maintenance is performed only after a breakdown of a component (Kothamasu & Huang, 2007). However, replacing a malfunctioning component after it has failed during service, results in exceptional costs and inevitable lower incomes. In PM, instead, a component is replaced when it reaches the end of its life cycle, which can be computed adopting many estimations. One of the historical way to perform this operation is to determine a conservative average of the component life cycle adopting the experience gained with all the components belonging to a specific class. Similarly to CM, this particular type of PM can bring unnecessary costs too, if the replaced component could have been used more than originally forecast. Moreover, this technique 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 a specification of PM, which aims at reducing both the costs of CM and non-predictive PM by relying on the exact decay state of each component and then by efficiently planning its maintenance (Mann et al., 1995; ISO BS, 2004). 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. Considering the estimated state of decay, it is possible to schedule each component's replacement before failures occur, maximizing its life cycle, according to the time required for each maintenance and to the geographical location of the ship (Mobley, 2002). As a result, the additional costs of CM and PM can be replaced with the lower ones of equipping the propulsion system with sensors and by collecting, storing, and analyzing these data for the purpose of creating effective predictive DDMs (Widodo & Yang, 2007; Mobley, 2002). It is worth noting that shipping companies already invested in maintenance to increase the reliability of their fleet and to reduce the costs. Several examples of Planned Maintenance System (PMS) are available and specifically designed for maintenance monitoring and integrated maintenance with class surveys (Danaos, 2017; DNV-GL, 2017; Spectec, 2017). Nevertheless, as reported in (Jackson et al., 2005), the use of effective predictive models is necessary to achieve reliability, availability and maintainability. For this reason, in this paper, the authors address the problem of building effective predictive models of the main components decay state in a Naval Propulsion System (NPS) for CBM purposes.Differently from the already available PMS systems, the proposed CBM approach can be adopted for real-time applications, to forecast and identify maintenance necessities. While physical models have been exploited in the past, recently, the ones based on statistical analysis have shown to outperform them (Peng et al., 2010). DDMs are adopted to correlate the available sensors measurements with the decay state of the different components of an NPS. The authors build virtual sensors able to continuously estimate the state of decay of the components based on other sensors measurements which are indirectly influenced by this decay.

Many examples of DA techniques applied to different CBM problems can be found in literature (Jardine et al., 2006). Among other, Support Vector Machines (Poyhonen et al., 2004), Hidden Markov Models (Bunks et al., 2000) and Kalman filter (Simani et al., 2003) are the most frequently used. Examples of DA approaches applied to the marine industry can be found in Palmé et al. (2011), where a standard Neural Network approach is used to improve monitoring of Gas Turbines, while Kernel based methods are applied in Singer et al. (1995), and Coraddu et al. (2016). In Akinfiev et al. (2008) and Bagavathiappan et al. (2013) image processing techniques are adopted for hull condition assessment. In Basurko & Uriondo (2015) the engine and propeller state is predicted adopting an Artificial Neural Network. A complete overview can be found in Lazakis et al. (2016). In Anantharaman (2013) a model for NPS CBM is developed adopting Reliability Block Diagrams and Fault Tree circuits, while in Michala et al. (2015) a CBM is presented together with a Decision Support System, adopting Machinery Risk Assessment techniques. In Mathur et al. (2001) and Byington et al. (2002) a review of the different techniques applied in latest years for CBM is given.

The novelty of this approach relies on the concurrent modeling of the four main components decay status of an NPS: Propeller, Hull, Gas Turbine and Gas Turbine Compressor. In fact, many works, present in literature, model just one or maximum two of these components at the same time (Palmé et al., 2011; Coraddu et al., 2016). The authors performed a regression analysis where the target is to estimate the actual decay 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 (Shawe-Taylor & Cristianini, 2004), Neural Network (Rosenblatt, 1958), Gaussian Processes (Rasmussen, 2006), Similarity Based Method (Duch, 2000), and Ensemble Methods (Breiman, 2001). Data coming from a Frigate characterized by a COmbined Diesel ELectric And Gas (CODLAG) propulsion plant have been exploited to show the effectiveness of the proposed approaches and to benchmark them in a realistic maritime application. These data will be available trough the UCI website, but they can already be downloaded at www.cbm.smartlab.ws. Confidentiality constraints with the Navy require the authors to use a real-data validated complex numerical simulator of a naval vessel (Coraddu et al., 2016). Nevertheless, even though the data acquired do not belong to any active unit in service, the simulator which was used to generate them, has been developed and tuned thanks to several years experience on numerical modeling of real propulsion plants (Figari & Altosole, 2007; Benvenuto & Campora, 2007).

The paper is organized 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 UCI. Section 4 reports the proposed DDMs. Results of the DDMs tested on these 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, characterized by a CODLAG NPS, widespread detailed in Coraddu et al. (2016). In particular, the gas turbine (GT) mechanically drives the two controllable pitch propellers (CPP) through a cross-connected gearbox. Besides, each shaft has its electric propulsion motor (EPM) mounted on the two shaft-lines. Two clutches between the gearbox (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). In this work, only the GT operating conditions have been taken into account.

2.2. Model Description

In this work, the authors consider an NPS numerical model developed in the Matlab® Simulink® software environment within many years of research (Altosole et al., 2009; Campora & Figari, 2003; Altosole et al., 2010). 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 gearbox, and the control system. In the previous literature, the authors presented a model that considers the GT and GTC decay performance (Coraddu et al., 2016; Benvenuto & Campora, 2005). The model is now further improved to take into account the performance decay of the HLL and PRP. This model is now readily to undertake an 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).

2.3. Degradation Model

The model described in Section 2.2 has been updated to take into account the performance decay of the following components: GT, GTC, HLL, and PRP. As reported in Altosole et al. (2014) and Coraddu et al. (2016), the time domain performance decay has been modeled using suitable coefficients embedded in the GTC low and high power modules of the model. This, allows quantifying the variation of performance of each component and consequently of the entire propulsion system.

GTC and GT Degradation Model

As reported in Tarabrin et al. (1998) and Meher-Homji et al. (1989), fouling of the GTC increases the specific fuel consumption and the temperature of the exhaust gas. In agreement with previous work, (Coraddu et al., 2016), 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 Coraddu et al. (2016) and Altosole et al. (2014). The authors applied a reduction factor, kM_T, to the GT flow rate to represent the effect of fouling.

PRP Degradation Model

An increase in the roughness of the blade surface is the primary cause of marine PRP performance degradation (Khor & Xiao, 2011). 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 modeled 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. (2002); Wan et al. (2002).

HLL Degradation Model

The main factors that affect hull performance are the shape, the coating used and the extent of fouling (Lindholdt et al., 2015). The resistance increase due to fouling has been modeled utilizing a correction factor kH based on the information reported in Hadler et al. (1962). 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 (Altosole et al., 2014).

3. From Data to Condition-Based Maintenance

3.1. Dataset Creation

In this work, authors will use a real-data validated complex numerical simulator of a Navy frigate described in Section 2 in order 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 of the University of California in Irvine (UCI) (Bache & Lichman, 2013). Currently, it can be downloaded from `www.cbm.smartlab.ws`.

The NPS model is characterised by the following input parameters:

- 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 analyzed, considering the linear dependency between the two variables.

- The HLL decay has been modeled according to the available literature (Hadler et al., 1962) as described in Section 2.3. The decay limit 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:

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

It is worth underlying that the performance decay functions reported in Section 2.3 are empirically derived and are only a function of time. The real degradation behavior of the physical asset should be described through specific functions able to describe the time dependency and the real operational profile. In order to overcome this issue, the authors considered each possible combination of GTC, GT, HLL, and PRP decays. Under this assumption, the authors sampled the range of decays with a uniform grid characterized by a degree of precision sufficient to have a good granularity of representation. Given the above premises, the evolution of the system can be exhaustively explored by simulating all its possible states. The space of possible states is described via the following tuple:

$$\left(lp, kK_t, kK_q, kH, kM_c, kM_t\right)_i, \quad i \in \{1, \cdots, 589223\} \tag{6}$$

since:

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

$$kK_t \in \mathcal{S}^{kK_t} = \{0.9, .9 + {}^{0.1}/_{14}, .09 + {}^{0.2}/_{14}, \cdots, 1.0\}, \tag{8}$$

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

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

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

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

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, 128 GB of RAM, 120 GB 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

Table 1: Measured values available from the continuous monitoring system

#	Variable name	Unit
1	Lever (lp)	[ ]
2	Vessel Speed	[knots]
3	GT shaft torque (GTT)	[kN m]
4	GT Speed (GT rpm)	[rpm]
5	Controllable Pitch Propeller Thrust stbd (CPP T stbd)	[N]
6	Controllable Pitch Propeller Thrust port (CPP T port)	[N]
7	Shaft Torque port (Q port)	[kN m]
8	Shaft rpm port (rpm port)	[rpm]
9	Shaft Torque stbd (Q stdb)	[kN m]
10	Shaft rpm stbd (rpm stbd)	[rpm]
11	HP Turbine exit temperature (T48)	[^{o}C]
12	Generator of Gas speed (GG rpm)	[rpm]
13	Fuel flow (mf)	[kg/s]
14	ABB TIC control signal (ABB Tic)	[ ]
15	GT Compressor outlet air pressure (P2)	[bar]
16	CGT Compressor outlet air temperature (T2)	[^{o}C]
17	External Pressure (Pext)	[bar]
18	HP Turbine exit pressure (P48)	[bar]
19	TCS TIC control signal (TCS Tic)	[ ]
20	Thrust coefficient stbd (Kt stbd)	[ ]
21	Propeller rps stbd (rps prop stbd)	[rps]
22	Thrust coefficient port (Kt port)	[ ]
23	Propeller rps port (rps prop port)	[rps]
24	Propeller Torque port (Q prop port)	[kN m]
25	Propeller Torque stbd (Q prop stbd)	[kN m]

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 can be straightforwardly mapped into a classical multi-output regression problem (Vapnik, 1998; Coraddu et al., 2016) where the aim is to predict the actual decay coefficient based on the automation data coming from the sensors installed on-board.

Since in authors' analysis four different components of an NPS are taken into account (GT, GTC, HLL, and PRP), they decided to perform an incremental analysis by breaking down the problem into simpler ones. In particular, first, only one NPS decayed component at the time is considered, then authors consider the possible combination of two NPS decayed components and so on until finally all the four NPS components are contemporarily considered. Consequently, since four NPS components are considered, $4 = \binom{4}{1}$ problems have to be solved when just one NPS decayed component at the time is considered, then $6 = \binom{4}{2}$ problems have to be solved when two NPS decayed components at the time are considered and so on. Finally when all the four NPS components are contemporarily considered $1 = \binom{4}{4}$ problem has to be solved, for a total of $15 = \sum_{i=1}^{4} \binom{4}{i}$ problems. Therefore, from the dataset described in Section 3.1, 15 sub-datasets corresponding to the cases mentioned above have been extracted. For the sake of clarity in Table 2 the 15 problems with the corresponding decay values are reported, not that in all the problems $lp \in \mathcal{S}^{lp}$.

4. Machine Learning Techniques

In this section, the authors will present the Machine Learning (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

Table 2: The 15 sub-problems corresponding to considering different decayed component at the time

Prob. #	Decayed Component Name	Non Decayed Component Name	Decayed Component	Non Decayed Component
1	GT	GTC,HLL,PRP	$kM_t{\in}\mathcal{S}^{kM_t}$	$kM_c{=}1,kH{=}1,kK_t{=}1$
2	GTC	GT,HLL,PRP	$kM_c{\in}\mathcal{S}^{kM_c}$	$kM_t{=}1,kH{=}1,kK_t{=}1$
3	HLL	GT,GTC,PRP	$kH{\in}\mathcal{S}^{kH}$	$kM_c{=}1,kM_t{=}1,kK_t{=}1$
4	PRP	GT,GTC,HLL	$kK_t{\in}\mathcal{S}^{kK_t}$	$kM_c{=}1,kM_t{=}1,kH{=}1$
5	GT,GTC	HLL,PRP	$kM_t{\in}\mathcal{S}^{kM_t},kM_c{\in}\mathcal{S}^{kM_c}$	$kH{=}1,kK_t{=}1$
6	GT,HLL	GTC,PRP	$kM_t{\in}\mathcal{S}^{kM_t},kH{\in}\mathcal{S}^{kH}$	$kM_c{=}1,kK_t{=}1$
7	GT,PRP	GTC,HLL	$kM_t{\in}\mathcal{S}^{kM_t},kK_t{\in}\mathcal{S}^{kK_t}$	$kM_c{=}1,kH{=}1$
8	GTC,HLL	GT,PRP	$kM_c{\in}\mathcal{S}^{kM_c},kH{\in}\mathcal{S}^{kH}$	$kM_t{=}1,kK_t{=}1$
9	GTC,PRP	GT,HLL	$kM_c{\in}\mathcal{S}^{kM_c},kK_t{\in}\mathcal{S}^{kK_t}$	$kM_t{=}1,kH{=}1$
10	HLL,PRP	GT,GTC	$kH{\in}\mathcal{S}^{kH},kK_t{\in}\mathcal{S}^{kK_t}$	$kM_t{=}1,kM_c{=}1$
11	GT,GTC,HLL	PRP	$kM_t{\in}\mathcal{S}^{kM_t},kM_c{\in}\mathcal{S}^{kM_c},kH{\in}\mathcal{S}^{kH}$	$kK_t{=}1$
12	GT,GTC,PRP	HLL	$kM_t{\in}\mathcal{S}^{kM_t},kM_c{\in}\mathcal{S}^{kM_c},kK_t{\in}\mathcal{S}^{kK_t}$	$kH{=}1$
13	GT,HLL,PRP	GTC	$kM_t{\in}\mathcal{S}^{kM_t},kH{\in}\mathcal{S}^{kH},kK_t{\in}\mathcal{S}^{kK_t}$	$kM_c{=}1$
14	GTC,HLL,PRP	GT	$kM_c{\in}\mathcal{S}^{kM_c},kH{\in}\mathcal{S}^{kH},kK_t{\in}\mathcal{S}^{kK_t}$	$kM_t{=}1$
15	GT,GTC,HLL,PRP		$kM_t{\in}\mathcal{S}^{kM_t},kM_c{\in}\mathcal{S}^{kM_c},kH{\in}\mathcal{S}^{kH},kK_t{\in}\mathcal{S}^{kK_t}$	

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 datasets 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 (Vapnik, 1998). Regression is one of the most popular examples of supervised ML problems (Shawe-Taylor & Cristianini, 2004) which is characterized by $\mathcal{Y} \in \mathtt{R}$.

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. It has to be clarified that $\mathcal{T}_m$ 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.

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 (Rosasco et al., 2004).

In a typical regression framework, there are two losses that are mainly used for estimating the quality of a regressor: the absolute loss $\ell_1(f(\boldsymbol{x}), y) = |f(\boldsymbol{x}) - y|$ and the squared loss $\ell_2(f(\boldsymbol{x}), y) = (f(\boldsymbol{x}) - y)^2$. Based on these losses it is possible to define different indexes of performance, which differently weight the distance between y_i^t and $f(\boldsymbol{x}_i^t)$ (Ghelardoni et al., 2013):

- the Mean Squared Error (MSE) is computed by taking the mean square loss value of f over $\mathcal{T}_m$: $\text{MSE} = {}^1\!/_m \sum_{i=1}^m \ell_2(f(\boldsymbol{x}_i^t), y_i^t)$;
- the Normalized Mean Square Error (NMSE) is similar to the MSE but a normalization term composed of the mean value of the difference between the true values and their mean is applied: $\text{NMSE} = {}^{\sum_{i=1}^m \ell_2(f(\boldsymbol{x}_i^t), y_i^t)}\!/_{\Delta}$;
- the Relative Error Percentage (REP) is similar to NMSE but the normalization term is composed of the sum of the squared true values. Then the result is square rooted and reported in percentage: $\text{REP} = 100\sqrt{{}^{\sum_{i=1}^m \ell_2(f(\boldsymbol{x}_i^t), y_i^t)}\!/_{\sum_{i=1}^m (y_i^t)^2}}$;
- the Mean Absolute Error (MAE) is similar to MSE but the absolute loss is used instead: $\text{MAE} = {}^1\!/_m \sum_{i=1}^m \ell_1(f(\boldsymbol{x}_i^t), y_i^t)$;
- the Mean Absolute Percentage Error (MAPE) can be described as the MAE expressed in percentage: $\text{MAPE} = {}^{100}\!/_m \sum_{i=1}^m {}^{\ell_1(f(\boldsymbol{x}_i^t), y_i^t)}\!/_{y_i^t}$;
- the Pearson Product-Moment Correlation Coefficient (PPMCC) is a measure of the linear dependency between $f(\boldsymbol{x}_i^t)$ and y_i^t with $i \in \{1, \cdots, m\}$: $\text{PPMCC} = {}^{\sum_{i=1}^m (y_i^t - \overline{y})(f(\boldsymbol{x}_i^t) - \hat{y})}\!/_{\sqrt{\sum_{i=1}^m (y_i^t - \overline{y})^2}\sqrt{\sum_{i=1}^n (f(\boldsymbol{x}_i^t) - \hat{y})^2}}$,

where authors have exploited the following notation

$$\overline{y} = \frac{1}{m}\sum_{i=1}^m y_i^t, \quad \Delta = \frac{1}{m}\sum_{i=1}^m (y_i^t - \overline{y})^2, \quad \hat{y} = \frac{1}{m}\sum_{i=1}^m f(\boldsymbol{x}_i^t). \tag{13}$$

4.2. Machine Learning Techniques

In this section, authors will present the supervised 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 (Kohavi et al., 1995; Bartlett et al., 2002; Anguita et al., 2012). Finally, authors will also check for possible spurious correlation in the data by performing the Feature Selection (FS) phase (Guyon & Elisseeff, 2003; Friedman et al., 2001; Yoon et al., 2005). 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 (Guyon & Elisseeff, 2003). 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 Regressive 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 (Rosenblatt, 1958), 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 back-propagation (Rumelhart et al., 1988). 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) (Cybenko, 1989; Bishop, 1995), while, if multiple layers are staked together, the architecture is defined as deep (DNN) (Hinton et al., 2006; Bengio, 2009). 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 (Huang et al., 2015, 2006; Cambria & Huang, 2013). 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 (Huang et al., 2004).

KMs are a family of ML techniques which exploits the "Kernel trick" for distances in order to extend linear techniques to the solution of non-linear problems (Cristianini & Shawe-Taylor, 2000; Scholkopf, 2001). In the case of regression, KMs select f as the function which minimizes the tradeoff between the sum of the accuracy over the data, namely the empirical error, and the complexity of the solution, namely the regularization term (Rifkin et al., 2003; Cristianini & Shawe-Taylor, 2000). Two of the most known and effective KM techniques are Kernelized Regularized Least Squares (KRLS), and Support Vector Regression (SVR). The hyperparameters of the KRLS $\mathcal{H}^{\text{KRLS}}$ are: the kernel, which is usually fixed and in this paper author chose the Gaussian Kernel for the reasons described in Keerthi & Lin (2003); Oneto et al. (2015), its hyperparameter h_1 and the regularization hyperparameter h_2. SVR, instead, is a method which roots in the Statistical Learning Theory (Vapnik, 1998) and differs from the KRLS as it introduces an ϵ insensitivity value for the errors (Shawe-Taylor & Cristianini, 2004). The hyperparameters of the SVR $\mathcal{H}^{\text{SVR}}$ are the same as the ones of KRLS plus the insensitivity value h_3.

EMs ML techniques relies on the simple fact that combining the output of several classifiers results in a much better performance than using any one of them alone (Germain et al., 2015; Breiman, 2001). Random Forest (RF) (Breiman, 2001) and Random Rotation Ensembles (RRF) (Blaser & Fryzlewicz, 2015), two popular state-of-the-art and widely adopted methods, combine many decision trees in order to obtain a effective predictors which have limited hyperparameter sensitivity and high numerical robustness (Fernández-Delgado et al., 2014; Wainberg et al., 2016). Both RF and RRF have hidden hyperparameters which are considered fixed in this work because of their limited effect (Orlandi et al., 2016).

BMs are ML techniques where, instead of choosing a particular $f \in \mathcal{F}$ a distribution for choosing $f \in \mathcal{F}$ is defined (Gelman et al., 2014). Gaussian Processes (GP) learning algorithm is a popular BM (Rasmussen, 2006) 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 $\mathcal{H}^{\mathrm{GP}}$ 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 (Duch, 2000). 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 (Cover & Hart, 1967). The hyperparameter of the KNN $\mathcal{H}^{\mathrm{KNN}}$ is the number of neighbors of $\boldsymbol{x}$ to be considered h_1.

4.2.2. Model Selection

MS deals with the problem of tuning the hyperparameters of each learning algorithm (Anguita et al., 2012). Several methods exist for MS purpose: resampling methods, like the well-known k-Fold Cross Validation (KCV) (Kohavi et al., 1995) or the nonparametric Bootstrap (BTS) approach (Efron & Tibshirani, 1994; Anguita et al., 2000) approaches, 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 $\mathcal{L}_l^r \cap \mathcal{V}_v^r = \oslash$, $\mathcal{L}_l^r \cup \mathcal{V}_v^r = \mathcal{D}_n$. Then, in order 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:

$$\mathcal{H}^* : \quad \min_{\mathcal{H} \in \mathfrak{H}} \frac{1}{n_r} \sum_{r=1}^{n_r} \frac{1}{v} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{V}_v^r} \ell(\mathscr{A}_{\mathcal{H}, \mathcal{L}_l^r}(\boldsymbol{x}_i), y_i), \tag{14}$$

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.

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 (Anguita et al., 2012). For implementing the complete k-fold cross validation, 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 (Kohavi et al., 1995; Arlot & Celisse, 2010; Anguita et al., 2012). Finally, for implementing the bootstrap, $l = n$ 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 $\mathcal{L}_l^r$ (Efron & Tibshirani, 1994; Anguita et al., 2012). Note that for the bootstrap procedure $r \leq \binom{2n-1}{n}$. In this paper the BTS is exploited because it represents the state-of-the-art approach (Efron & Tibshirani, 1994; Anguita et al., 2012).

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 (Guyon & Elisseeff, 2003; Friedman et al., 2001; Yoon et al., 2005). 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, 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 (Good, 2013), 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. Experimental results

In this section, the authors report the results obtained by the different methods applied to the CBM of the main components of an NPS, in the proposed regression framework, as described in Section 2, based on the data described in Section 3.

As described in Section 4, the authors tackle a regression problem where the actual value of the decay parameters needs to be estimated.

The different datasets considered in Section 3.2, corresponding to the 15 problems of Table 2 $\{P1, \cdots, P15\}$, were divided into training and test set, respectively $\mathcal{D}_n$ 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.

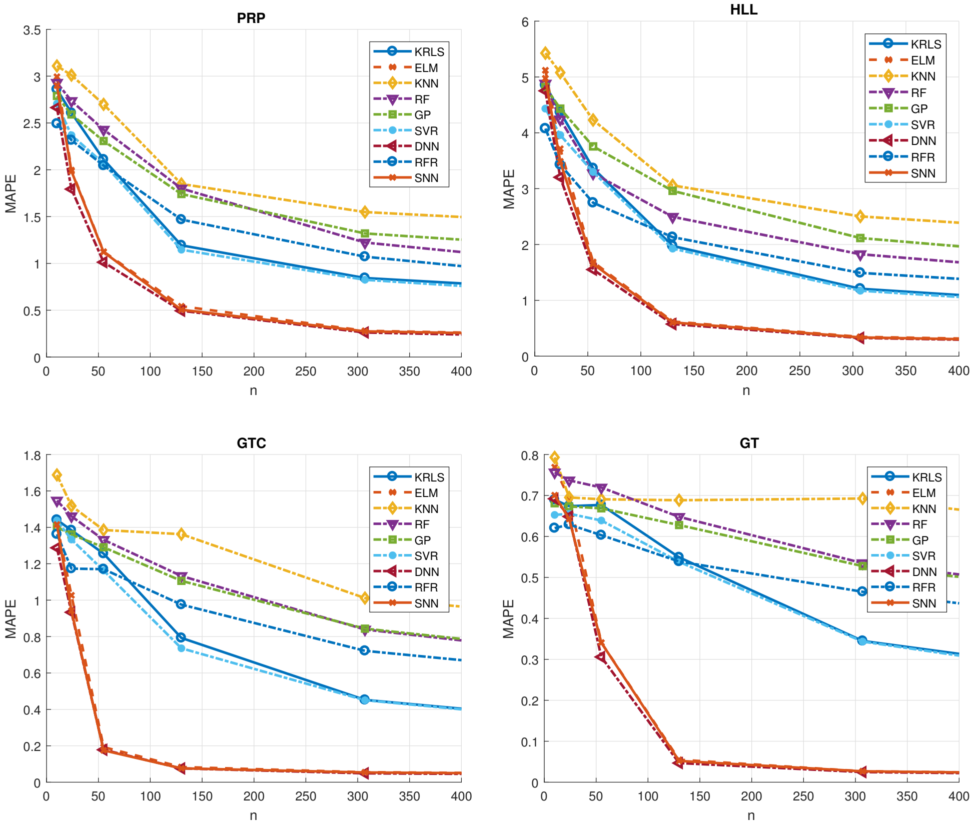

Figure 1: MAPE of the models learned with the different algorithms (DNN, SNN, ELM, SVR, KRLS, KNN, and GP) when varying n for $P15$ and for the four main NPS components.

For each supervised regression learning task and each ML technique, an MS procedure was performed, as described in Section 4.2.2. Here-below, 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. SVR: the set of hyperparameters is $\mathcal{H}^{SVR} = \{h_1, h_2, h_3\}$ and authors chose it in $\mathfrak{H}^{\text{SVR}} = \{10^{-6}, 10^{-5}, \cdots, 10^3\} \times \{10^{-2}, 10^{-1.4}, \cdots, 10^3\} \times \{10^{-2}, 10^{-1.4}, \cdots, 10^3\}$;
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 $\mathfrak{H}^{\text{GP}} = \{10^0, 10^{0.3}, \cdots, 10^3\}$;

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 (Chollet, 2015) has been exploited. For ELM, SVR, KRLS, KNN, and GKNN a custom R implementation has been developed. For RF the R package of Liaw & Wiener (2002) has been exploited. For RFE the implementation of Blaser & Fryzlewicz (2015) has been exploited. For GP the R package of Zeileis et al. (2004) has been exploited.

5.1. Regression Models Error Results

In this section, the results on the regression problem are reported. Note that, based on the problem under exam, just a subset of the components may decay. Moreover, since this dataset cardinality is limited, it is possible that a particular combination of problem and cardinality of the training set cannot be tested. For example, when just one component decays, nearly hundred samples are available (see Section 3) and then a maximum of $n = 55$. Note that, in a real-world scenario, the only useful problem is $P15$, namely when all the components decay contemporarily. In Figures 1 authors report the MAPE of the models learned with the different algorithms when varying n for $P15$ and the four NPS components. In Figures 2 authors report the MAPE of the DNN (the best performing model) when varying n for the different problems under examination and the four NPS components.

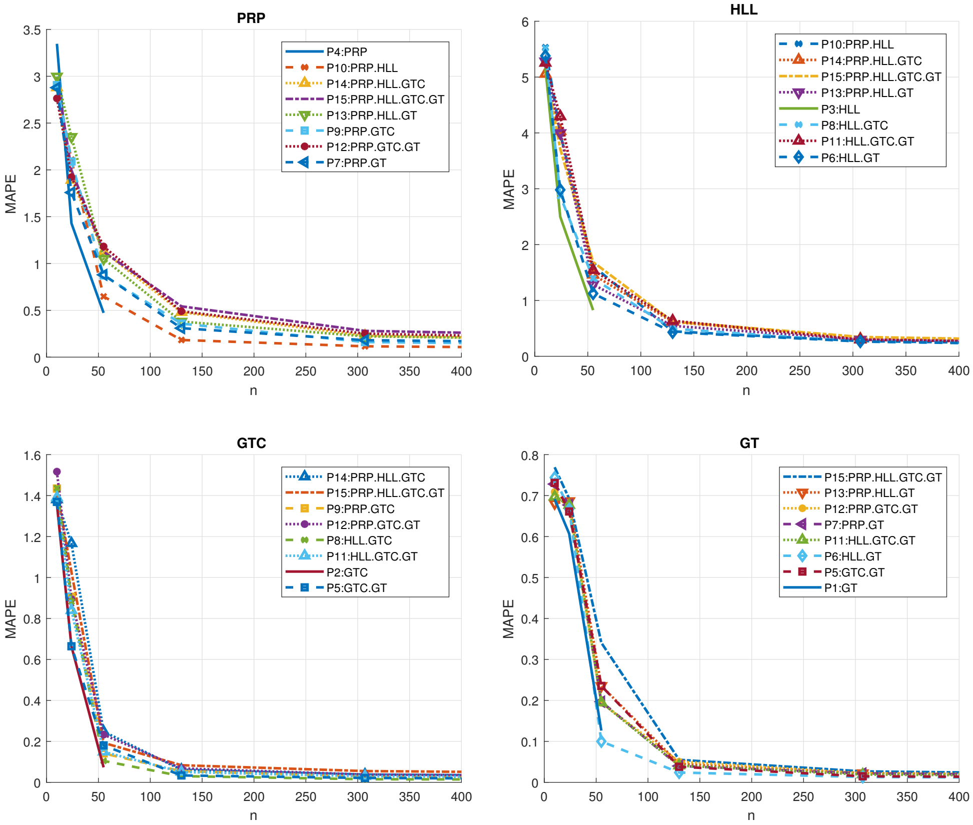

Figure 2: MAPE of the models learned with DNN when varying n for the different problems $P1, \cdots, P15$ and for the four main NPS components.

From the different figures it is possible to observe that:

- as expected the larger is n the better performances are achieved by the learned models (see Figures 2 and 1);
- the models learned with ELM, SNN, and especially DNN generally show the best performances (see Figure 1);
- the larger is the number of decaying components authors consider the lower performances are achieved by the learned models (see Figure 2);

- the most complicate decay to predict is the one of the HLL, in fact the problems where the HLL decays are the ones which show lower accuracies (see Figure 2);
- unfortunately, as it can be seen from 2 and 1, this approach requires a significant amount of labelled data, which could not be retrieved in many operational scenarios. In fact, 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. Collecting 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.

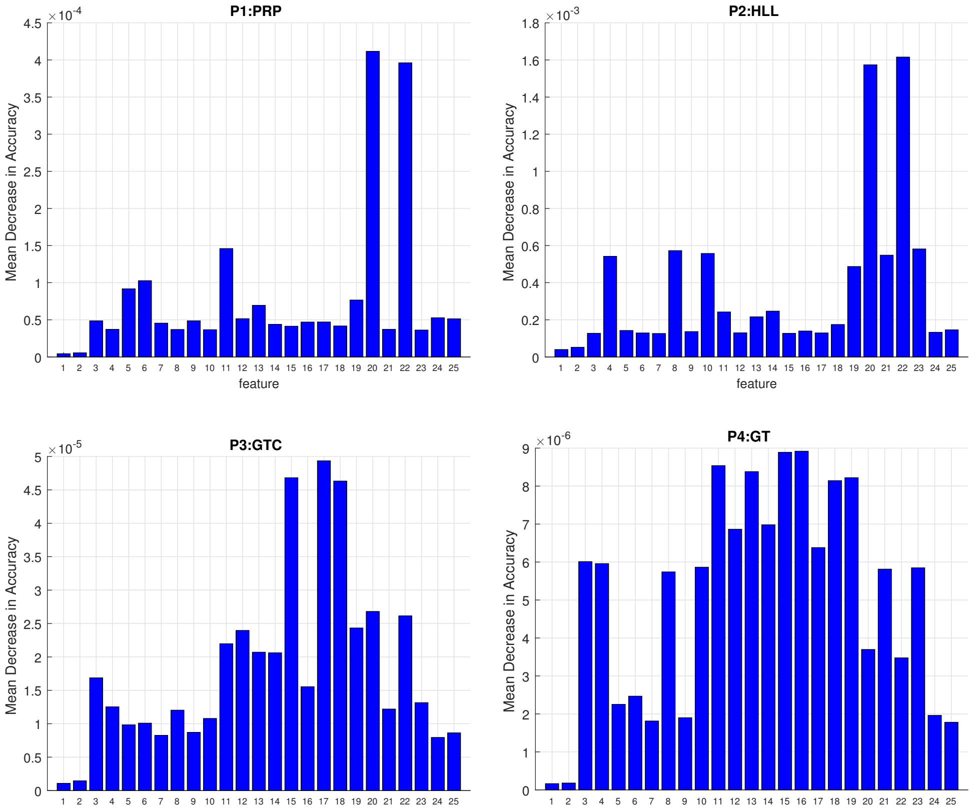

Figure 3: FS performed with RF for the four main NPS components for problem $P1$, $P2$, $P3$, and $P4$

Finally in Figure 3 and 4 the FS phase is reported, performed with RF, for

$P1$, $P2$, $P3$, and $P4$ (see Figure 3) and for $P15$ (see Figure 4). In particular, for each problem and each feature, the mean decrease in accuracy is reported as described in Section 4.3.

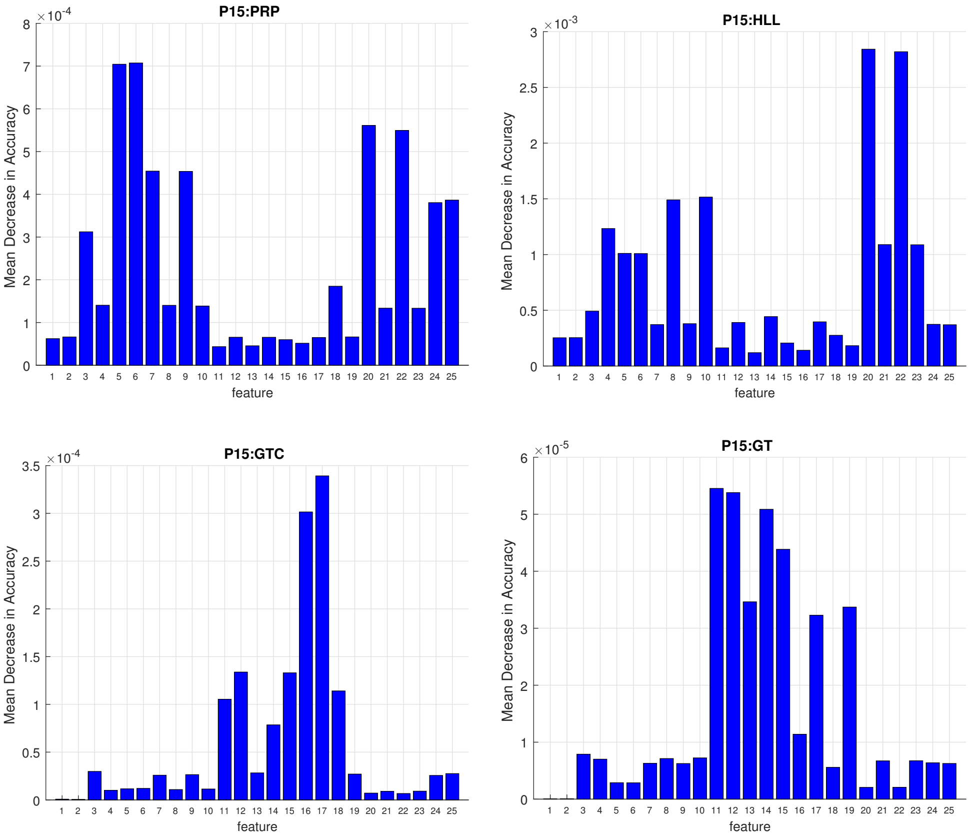

Figure 4: FS performed with RF for the four main NPS components for problem $P15$.

From Figure 3 and 4 it is possible to observe that:

- as expected, when just one component decays (see Figure 3), few predictors have strong predictive power while when all the components decay (see Figure 4) many predictors need to be considered in order to achieve satisfying accuracies;
- from Figure 3, it is possible to note that the RF model can adequately accounts for the relevant features as the outcome is consistent with the available knowledge (note that even if DNN has higher predictive capabilities with respect to RF the latter is still competitive, see Figure 1). In fact, for the $P1$ and $P2$ (PRP and HLL decays) the features

22 and 24 (Thrust coefficient stbd and port) have strong predictive power. Moreover, for the $P3$ (GTC decay) the features describing the thermodynamic process, $17, 19$, and 20 (GT Compressor outlet air pressure, External Pressure ,and HP Turbine exit pressure) have the highest predictive power. Finally, several features are necessary for the GT decay prediction in $P4$.

6. Conclusions

The behavior and interaction of the main components of Ship Propulsion Systems cannot be easily modeled with a priori physical knowledge, considering the large amount of variables influencing them. In fact, in this work, authors showed that by exploiting the most recent statistical techniques and the large amount of historical data collected by the current on-board automation systems it is possible to build effective Data-Driven Models which do not require any a priori knowledge. In particular, the developed models are able to continuously monitor the propulsion equipment to avoid Preventive or Corrective Maintenance and take decisions based on the actual condition of the propulsion plant. A naval ship, characterized by a COmbined Diesel ELectric And Gas propulsion plant, has been exploited to show the effectiveness of the proposed approaches. Because of confidentiality constraints with the Navy the authors used a real-data validated simulator and the dataset has been published for free use.

In conclusion, this study proved that it is possible to treat a Condition-Based Maintenance problem in a supervised fashion adopting regression techniques. These models can be adopted for real-time applications directly onboard, to easily and quickly identify maintenance necessities. Nevertheless, the dataset adopted for this study can difficulty be collected in a real-world scenario. In this paper, the authors propose an example of DMMS for CBM, providing evidence of the possible advantages in the experimental section with an indirect measurement of the potential savings. The proposed models are quite effective and reliable, and, for this reason, the authors' approach can correctly identify the state of decay of the system, bringing to an optimal predictive maintenance strategy with a reduction in the operational and maintenance costs. On the other hand, the authors would like to address some findings which need to be solved. First, supervised DMMS require a significant amount of labelled data to efficiently monitor the NPS components state, which would require the operational stop or even the drydocking of the vessel. Vessels are rarely available in the harbour for maintenance, and the necessary parameters cannot be easily acquired when the ship is operative. Second, the experiments carried out are relative to a specific vessel, and this work cannot be straightforwardly mapped to any new ship without a further investigation on transfer learning to understand how to use available data from other vessels. To solve the first issue, the authors suggest the possibility of adopting a relaxation on the decay value prediction requirement, by transforming the regression problem into a simpler one by predicting only the necessity of each component to be replaced or not, instead of the precise component decay. Moreover, in the future, authors will focus their efforts on the development of supervised and unsupervised methods which could solve this problem and allow a simplification of the data collection procedure.

References

Abu-Elanien, A. E. B., & Salama, M. M. A. (2010). Asset management techniques for transformers. *Electric power systems research*, *80*, 456–464.

Akinfiev, T. S., Armada, M. A., & Fernandez, R. (2008). Nondestructive testing of the state of a ship's hull with an underwater robot. *Russian Journal of Nondestructive Testing*, *44*, 626–633.

Altosole, M., Benvenuto, G., & Campora, U. (2010). Numerical modelling of the engines governors of a codlag propulsion plant. In *International Conference on Marine Sciences and Technologies*.

Altosole, M., Benvenuto, G., Figari, M., & Campora, U. (2009). Real-time simulation of a cogag naval ship propulsion system. *Journal of Engineering for the Maritime Environment*, *223*, 47–62.

Altosole, M., Campora, U., Martelli, M., & Figari, M. (2014). Performance decay analysis of a marine gas turbine propulsion system. *Journal of Ship Research*, *58*, 117–129.

Anantharaman, M. (2013). Using reliability block diagrams and fault tree circuits, to develop a condition based maintenance model for a vessel's main propulsion system and related subsystems. *Transnav: International Journal On Marine Navigation And Safety Of Sea Transportation*, *7*.

Anguita, D., Boni, A., & Ridella, S. (2000). Evaluating the generalization ability of support vector machines through the bootstrap. *Neural Processing Letters*, *11*, 51–58.

Anguita, D., Ghio, A., Greco, N., Oneto, L., & Ridella, S. (2010). Model selection for support vector machines: Advantages and disadvantages of the machine learning theory. In *International Joint Conference on Neural Networks*.

Anguita, D., Ghio, A., Oneto, L., & Ridella, S. (2012). In-sample and out-of-sample model selection and error estimation for support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, *23*, 1390–1406.

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, *4*, 40–79.

Atlar, M., Glover, E. J., Candries, M., Mutton, R. J., & Anderson, C. D. (2002). The effect of a foul release coating on propeller performance. In *Marine Science and Technology for Environmental Sustainability*.

Bache, K., & Lichman, M. (2013). UCI machine learning repository.

Bagavathiappan, S., Lahiri, B. B., Saravanan, T., Philip, J., & Jayakumar, T. (2013). Infrared thermography for condition monitoring-a review. *Infrared Physics & Technology*, *60*, 35–55.

Bartlett, P. L., Boucheron, S., & Lugosi, G. (2002). Model selection and error estimation. *Machine Learning*, *48*, 85–113.

Basurko, O. C., & Uriondo, Z. (2015). Condition-based maintenance for medium speed diesel engines used in vessels in operation. *Applied Thermal Engineering*, *80*, 404–412.

Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, *2*, 1–127.

Benvenuto, G., & Campora, U. (2005). A gas turbine modular model for ship propulsion studies. In *Symposium on High Speed Marine Vehicles*.

Benvenuto, G., & Campora, U. (2007). Performance prediction of a faulty marine diesel engine under different governor settings. In *International Conference on Marine Research And Transportation*.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

Blaser, R., & Fryzlewicz, P. (2015). Random rotation ensembles. *The Journal of Machine Learning Research*, *2*, 1–15.

Boucheron, S., Bousquet, O., & Lugosi, G. (2005). Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics*, *9*, 323–375.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Budai, G., Huisman, D., & Dekker, R. (2006). Scheduling preventive railway maintenance activities. *The International Journal of Advanced Manufacturing Technology*, *57*, 1035–1044.

Budai-Balke, G. (2009). *Operations research models for scheduling railway infrastructure maintenance*. Rozenberg Publishers.

Bunks, C., McCarthy, D., & Al-Ani, T. (2000). Condition-based maintenance of machines using hidden markov models. *Mechanical Systems and Signal Processing*, *14*, 597–612.

Byington, C. S., Roemer, M. J., & Galie, T. (2002). Prognostic enhancements to diagnostic systems for improved condition-based maintenance. In *Aerospace Conference*.

Cambria, E., & Huang, G.-B. (2013). Extreme learning machines. *IEEE Intelligent Systems*, *28*, 30–59.

Campora, U., & Figari, M. (2003). Numerical simulation of ship propulsion transients and full-scale validation. *Journal of Engineering for the Maritime Environment*, *217*, 41–52.

Carlton, J. (2011). *Marine propellers and propulsion*. Butterworth-Heinemann.

Chollet, F. (2015). Keras. `https://keras.io`.

Coraddu, A., Oneto, L., Baldi, F., & Anguita, D. (2015). Ship efficiency forecast based on sensors data collection: Improving numerical models through data analytics. In *OCEANS 2015-Genova*.

Coraddu, A., Oneto, L., Baldi, F., & Anguita, D. (2017). Vessels fuel consumption forecast and trim optimisation: A data analytics perspective. *Ocean Engineering*, *130*, 351–370.

Coraddu, A., Oneto, L., Ghio, A., Savio, S., Anguita, D., & Figari, M. (2016). Machine learning approaches for improving condition-based maintenance of naval propulsion plants. *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment*, *230*, 136–153.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*, 21–27.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, *2*, 303–314.

Danaos (2017). Planned maintenance system. `https://danaosmc.wordpress.com/2016/03/30/planned-maintenance-system`.

Das, S., & Chen, M. (2001). Yahoo! for amazon: Extracting market sentiment from stock message boards. In *Asia Pacific finance association annual conference (APFA)*.

DNV-GL (2017). Shipmanager technical. `https://www.dnvgl.com/services/planned-maintenance-system-for-technical-ship-management-shipmanager-technical-1509`.

Duch, W. (2000). Similarity-based methods: a general framework for classification, approximation and association. *Control and Cybernetics*, *29*.

Efron, B., & Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems. *The Journal of Machine Learning Research*, *15*, 3133–3181.

Figari, M., & Altosole, M. (2007). Dynamic behaviour and stability of marine propulsion systems. *Journal of Engineering for the Maritime Environment*, *221*, 187–205.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning*. Springer series in statistics Springer, Berlin.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2014). *Bayesian data analysis, vol. 2*. Chapman and Hall/CRC.

Germain, P., Lacasse, A., Laviolette, F., Marchand, M., & Roy, J. F. (2015). Risk bounds for the majority vote: From a pac-bayesian analysis to a learning algorithm. *The Journal of Machine Learning Research*, *16*, 787–860.

Ghelardoni, L., Ghio, A., & Anguita, D. (2013). Energy load forecasting using empirical mode decomposition and support vector regression. *IEEE Transactions on Smart Grid*, *4*, 549–556.

Good, P. (2013). *Permutation tests: a practical guide to resampling methods for testing hypotheses*. Springer Science & Business Media.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, *3*, 1157–1182.

Györfi, L., Kohler, M., Krzyzak, A., & Walk, H. (2006). *A distribution-free theory of nonparametric regression*. Springer Science & Business Media.

Hadler, J. B., Wilson, C. J., & Beal, A. L. (1962). Ship standardisation trial performance and correlation with model predictions. In *The Society of Naval Architects and Marine Engineers*.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*, 1527–1554.

Huang, G., Huang, G.-B., Song, S., & You, K. (2015). Trends in extreme learning machines: A review. *Neural Networks*, *61*, 32–48.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. In *IEEE International Joint Conference on Neural Networks*.

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, *70*, 489–501.

ISO BS (2004). 13372: 2012, condition monitoring and diagnostics of machines-vocabulary. In *British Standards*.

Jackson, Y., Tabbagh, P., Gibson, P., & Seglie, E. (2005). The new department of defense (dod) guide for achieving and assessing ram. In *Reliability and Maintainability Symposium*.

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, *20*, 1483–1510.

Keerthi, S. S., & Lin, C. J. (2003). Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, *15*, 1667–1689.

Khor, Y. S., & Xiao, Q. (2011). Cfd simulations of the effects of fouling and antifouling. *Ocean Engineering*, *38*, 1065–1079.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*.

Kothamasu, R., & Huang, S. H. (2007). Adaptive mamdani fuzzy model for condition-based maintenance. *Fuzzy Sets and Systems*, *158*, 2715–2733.

Lazakis, I., Dikis, K., Michala, A. L., & Theotokatos, G. (2016). Advanced ship systems condition monitoring for enhanced inspection, maintenance and decision making in ship operations. *Transportation Research Procedia*, *14*, 1679–1688.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R News*, *2*, 18–22.

Lindholdt, A., Dam-Johansen, K., Olsen, S. M., Yebra, D. M., & Kiil, S. (2015). Effects of biofouling development on drag forces of hull coatings for ocean-going ships: a review. *Journal of Coatings Technology and Research*, *12*, 315–444.

Linoff, G. S., & Berry, M. J. A. (2011). *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons.

Management, G. B. Q., Committee, S. S., & Institution, B. S. (1984). *British Standard Glossary of Maintenance Management Terms in Terotechnology*. British Standards Institution.

Mann, L., Saxena, A., & Knapp, G. M. (1995). Statistical-based or condition-based preventive maintenance? *Journal of Quality in Maintenance Engineering*, *1*, 46–59.

Mannini, A., & Sabatini, A. M. (2010). Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, *10*, 1154–1175.

Mathur, A., Cavanaugh, K. F., Pattipati, K. R., Willett, P. K., & Galie, T. R. (2001). Reasoning and modeling systems in diagnosis and prognosis. In *SPIE Aerosense Conference*.

Meher-Homji, C. B., Focke, A. B., & Wooldridge, M. B. (1989). Fouling of axial flow compressors - causes, effects, detection, and control. In *Eighteenth Turbomachinery Symposium*.

Michala, A. L., Lazakis, I., & Theotokatos, G. (2015). Predictive maintenance decision support system for enhanced energy efficiency of ship machinery. In *International Conference on Shipping in Changing Climates*.

Mobley, R. K. (2002). *An introduction to predictive maintenance*. Butterworth-Heinemann.

Nguyen, T. T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials*, *10*, 56–76.

Oneto, L., Anguita, D., Coraddu, A., Cleophas, T., & Xepapa, K. (2016a). Vessel monitoring and design in industry 4.0: A data driven perspective. In *IEEE International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow*.

Oneto, L., Fumeo, E., Clerico, G., Canepa, R., Papa, F., Dambra, C., Mazzino, N., & Anguita, D. (2016b). Advanced analytics for train delay prediction systems by including exogenous weather data. In *International Conference on Data Science and Advanced Analytics*.

Oneto, L., Ghio, A., Ridella, S., & Anguita, D. (2015). Support vector machines and strictly positive definite kernel: The regularization hyperparameter is more important than the kernel hyperparameters. In *IEEE International Joint Conference on Neural Networks*.

Orlandi, I., Oneto, L., & Anguita, D. (2016). Random forests model selection. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

Palmé, T., Breuhaus, P., Assadi, M., Klein, A., & Kim, M. (2011). New alstom monitoring tools leveraging artificial neural network technologies. In *Turbo Expo: Turbine Technical Conference and Exposition*.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *ACL-02 conference on Empirical methods in natural language processing-Volume 10*.

Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, *50*, 297–313.

Petersen, J. P., Winther, O., & Jacobsen, D. J. (2012). A machine-learning approach to predict main energy consumption under realistic operational conditions. *Ship Technology Research*, *59*, 64–72.

Poyhonen, S., Jover, P., & Hyotyniemi, H. (2004). Signal processing of vibrations for condition monitoring of an induction motor. In *Symposium on Control, Communications and Signal Processing*.

Rasmussen, C. E. (2006). Gaussian processes for machine learning. In *Gaussian Processes for Machine Learning*.

Rifkin, R., Yeo, G., & Poggio, T. (2003). Regularized least-squares classification. *Nato Science Series Sub Series III Computer and Systems Sciences*, *190*, 131–154.

Rosasco, L., De Vito, E., Caponnetto, A., Piana, M., & Verri, A. (2004). Are loss functions all the same? *Neural Computation*, *16*, 1063–1076.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*, 386.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, *5*, 1.

Scholkopf, B. (2001). The kernel trick for distances. In *Advances in neural information processing systems*.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge university press.

Shin, K., Lee, T. S., & Kim, H. (2005). An application of support vector machines in bankruptcy prediction model. *Expert Systems with Applications*, *28*, 127–135.

Simani, S., Fantuzzi, C., & Patton, R. J. (2003). *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. Springer-Verlag London.

Singer, R. M., Gross, K. C., & King, R. W. (1995). A pattern-recognition-based, fault-tolerant monitoring and diagnostic technique. In *Argonne National Lab., Idaho Falls, ID (United States)*.

Smith, T. W. P., O'Keeffe, E., Aldous, L., & Agnolucci, P. (2013). Assessment of shipping's efficiency using satellite ais data. In *International Council on Clean Transportation*.

Spectec (2017). Amos maintenance-materials-management. `http://www.spectec.net/maintenance-materials-management`.

Tarabrin, A. P., Schurovsky, V. A., Bodrov, A. I., & Stalder, J. P. (1998). An analysis of axial compressor fouling and a blade cleaning method. *Journal of Turbomachinery*, *120*, 256–261.

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley New York.

Waeyenbergh, G., & Pintelon, L. (2002). A framework for maintenance concept development. *International journal of production economics*, *77*, 299–313.

Wainberg, M., Alipanahi, B., & Frey, B. J. (2016). Are random forests truly the best classifiers? *The Journal of Machine Learning Research*, *17*, 1–5.

Wan, B., Nishikswa, E., & Uchida, M. (2002). The experiment and numerical calculation of propeller performance with surface roughness effects. *Journal of the Kansai Society of Naval Architects, Japan*, *2002*, 49–54.

Wang, H., Osen, O. L., Li, G., Li, W., Dai, H., & Zeng, W. (2015). Big data and industrial internet of things for the maritime industry in northwestern norway. In *IEEE Region 10 Conference TENCON*.

Widodo, A., & Yang, B. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical systems and signal processing*, *21*, 2560–2574.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Yoon, H., Yang, K., & Shahabi, C. (2005). Feature subset selection and feature ranking for multivariate time series. *IEEE transactions on knowledge and data engineering*, *17*, 1186–1198.

Zeileis, A., Hornik, K., Smola, A., & Karatzoglou, A. (2004). kernlab-an s4 package for kernel methods in r. *Journal of statistical software*, *11*, 1–20.