Deep Learning for Cavitating Marine Propeller Noise Prediction at Design Stage

1st Luca Oneto, 2ndFrancesca Cipollini
DIBRIS - University of Genoa
{luca.oneto, francesca.cipollini}@unige.it

3rd Leonardo Miglianti, 4th Giorgio Tani,
5th Stefano Gaggero, 7th Michele Viviani
DITEN - University of Genoa
{leonardo.miglianti, giorgio.tani}@unige.it
{stefano.gaggero, michele.viviani}@unige.it

6th Andrea Coraddu
Naval Architecture,
Ocean & Marine Engineering
Strathclyde University
andrea.coraddu@strath.ac.uk

***Abstract*—Reducing the noise impact of ships on the marine environment is one of the objectives of new propellers designs, since they represent the dominant source of underwater radiated noise, especially when cavitation occurs. Consequently, ship designers require new predictive tools able to verify the compliance with noise requirements and to compare the effectiveness of different design solutions. In this context, tools able to provide a reliable estimate of propeller noise spectra based just on the information available at design stage represent a fundamental tool to speed up the design process avoiding model scale tests. This work focus on developing such a tool, adopting methods coming from the world of Machine Learning and Deep Neural Networks, in order to create a model able to predict the cavitating marine propeller noise spectra. For this purpose authors will make use of a dataset collected by means of dedicated model scale measurements in a cavitation tunnel combined with the detailed flow characterization obtainable by calculations carried out with a Boundary Element Method. The performance of the proposed approaches are analyzed considering different definitions of the input and output variables used during the modelization.**

***Index Terms*—Marine Propeller, Cavitation Noise, Noise Prediction, Machine Learning, Deep Learning, Hybrid Models.**

I. Introduction

Marine propellers are the first contributor to ships radiated noise in water and, in case of cavitation, the noise levels increase abruptly [1]. In order to reduce the amount of noise produced, international organisations and class societies started to emanate non-mandatory guidelines to reduce the noise emissions [2], [3].

Within the different noise mitigation strategies that can be adopted by the shipping industry, the analysis of design solutions aimed at the reduction of radiated noise for new ships is of utmost importance. This task requires the availability of tools and procedures for the prediction of the propeller radiated noise even at the design stage. Model Scale Tests (MSTs) are traditionally considered the most reliable method for cavitation noise prediction, as they are performed in cavitation tunnels by testing a model of the propeller in the full scale working conditions. This approach represents a valuable tool both for the final verification of the propeller design and for the comparison of few alternative solutions, although scale effects must be carefully taken into account to retrieve the full scale noise from MSTs. However, MSTs are time-consuming, expensive, and their inclusion in the early stages of the design procedure is unpractical.

In this context, the availability of a propeller noise predictive tool, based on the information available at the design stage, is of paramount importance to reach cost and time effectiveness.

A. State-of-the-art

A possible approach is represented by the use of Physical Models (PMs) combined with empirical relations. In general, the functional form of these formulations is derived from the physical equations describing the phenomenon under some simplifying assumptions. The formulations are successively tweaked on available experimental data to adapt them to practical cases. As an example, a formulation for the prediction of the characteristic frequency of noise generated by cavitating vortices is presented in [4]; although this formulation succeeds in describing the behavior of an isolated vortex, it may require case-by-case tuning, especially when different geometries and wakes are considered. However, as claimed by the author, the method is not able to model the phenomenon considering all the whole dynamics, as for instance the interactions with other phenomena, if present. Moreover, a method for the prediction of tip vortex noise through an empirical relation was presented in [5]. Eventually, a simple empirical formulation for the noise generated by thruster propellers has been presented in [6] where a simple relation between the amplitude of noise and the area of sheet cavitation is provided.

An attractive alternative consists in tackling the problem utilising Data Driven Models (DDMs), as proposed in present work. DDMs approximate the relations existing between some target quantities (the targets) and the available data on quantities and phenomena influencing the targets (the features). This is achieved through robust statistical inference procedures and data collected in previous experiments, including both features and targets, to make predictions about previously unseen cases. These methods do not need any a-priory knowledge about the mathematical expression governing the physical system. As a consequence, the DDMs can also model complex propeller cavitation patterns, including possible interactions between different phenomena, without the need to consider any simplifying assumptions.

DDMs have proved to be valuable instruments in many marine applications, such as fuel consumption and efficiency prediction or ship components condition-based maintenance via status prediction [7]–[9], or to determine the best propeller design given operational requirements and constraints [10]. The application of DDMs in the field of ship radiated noise is mostly limited to classification problems [11], [12], while there is a lack of work regarding their application to the problem of ship and propeller noise modelling. Recently [13] proposed an Artificial Neural Network approach for the prediction of the propeller cavitation noise given a number of design parameters and an extensive collection of noise samples from cavitation tunnel tests.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

In the view of developing DDMs, the employment of MSTs to collect data by means of systematic investigations of propeller noise is very desirable. Nevertheless, also in the case of MSTs, collecting a very large set of test cases could require significant effort in terms of cost and time. Consequently, the available data is typically not as large as required. In this context, the capability of DDMs, which usually produce black-box (non-parametric) models, may be limited in terms of generality.

In order to improve the performance of the DDMs, the knowledge of the physical phenomena included in the PMs can be exploited. This idea is the basis of Hybrid Models (HMs), which are developed to take advantage of the best characteristics of both PMs and DDMs by combining them together. The application of HMs to the problem of propeller noise modelling based on data obtained through MSTs has been presented in [14] with satisfactory results.

B. Contribution

Models developed in the aforementioned works consider as features only the main characteristics and functioning parameters of the propeller. However, these quantities are generally not sufficient to describe the whole phenomenon that a propeller undergoes. Actually, cavitation noise depends on the whole propeller geometry, and on the flow field, including the effects of non-uniform propeller inflow, as in realistic conditions. Furthermore, if the considered features include only global parameters, it is not possible to appreciate the local effects of specific design solutions, which may be remarkable for custom designed propellers. With the purpose to address these limits, the approach presented in this work exploits as features the detailed results of hydrodynamic computations, carried out by means of an in-house developed Boundary Element Method (BEM) [15], [16]. These features provide an accurate characterisation of the hydrodynamic field of the propeller at an acceptable computational cost, implicitly including the effects of the complete propeller geometry.

The hydrodynamic quantities computed by the BEM (i.e. surface pressure distributions over the blades, and bound circulation) include highly structured data in the form of 3D and 2D tensors. In order to fully exploit all the information contained in these tensors, some advanced techniques coming from the world of Deep Learning [17] have been proposed. These techniques allow first to find a rich representation of the tensors by finding all the information describing the cavitation phenomena and then to exploit it to make an effective and sufficient prediction of the propeller cavitation noise.

These models have been compared with the conventional models previously presented in [14], where the 1D features were manually selected based on the theories on cavitation noise. It is demonstrated that the advanced models outstand the performance of the conventional ones if a high dimensional and complex dataset is considered. On the contrary, if a low dimensional dataset composed only by simple variables is taken into account, the performance of the advanced model is close to that of the more conventional models. As a result, the proposed approach is proven to bring significant improvements with respect to more traditional techniques if a complex set of features is provided.

Furthermore, with respect to [14], the whole feature set has been rearranged in order to include only data easily obtainable at the early stage of propeller design. Different combinations of input variables and targets have been considered, allowing to analyze the relative merits of each.

II. Problem definition

The present work aims to model the unknown relation existing among some input variables describing the propeller, its functioning points and the wake inflow (the features) and the corresponding cavitation radiated noise (the targets). This relation has to be learned from the previous observations of the targets which are collected in dedicated MST at the UNIGE cavitation tunnel (described in detail in [14]). Conversely, the features are obtained both from design papers and from BEM calculations. Surveys have been performed on controllable pitch propellers considering pitch settings, wake configurations and loads, chosen to provide an exhaustive characterisation of cavitation noise. The tests and the subsequent noise analyses have been performed in compliance with the International Towing Tank Conference (ITTC) regulations [18]. In particular for each working point (sample) the cavitation pattern and the noise emission have been assessed. Many cavitation types exist, but the purpose of this study is focused on the most common cavitation types, i.e. the tip vortex cavitation and the suction side sheet cavitation. As a result, the final dataset is composed by 258 samples, each sample corresponding to a row of the dataset.

In the modelisation phase, five alternative set of features have been accounted:

- Feature Set 1 (FS1), consisting in a set of scalar parameters providing a global description of the propeller geometry and its functioning conditions (see [14], [18], [19]);
- Feature Set 2 (FS2), consisting in the point by point intensity of the design ship inflow wake, evaluated on a discrete combination of radials and angular positions;
- Feature Set 3 (FS3), consisting in the punctual values of the geometric angle of attack, that is the relative angle between blade and flow, evaluated on the same locations as FS2 (according to formula in [19]);
- Feature Set 4 (FS4), consisting in the distribution on the blades of pressure coefficient, computed for a finite number of locations on the blade surface (identified by the radial and chordwise coordinate), for varying angular positions of the blade (see [20]);
- Feature Set 5 (FS5), consisting in the complete tensor of velocity circulation around the blade, computed for a finite number of radial locations and for varying angular positions of the blade (see [20]).

FS4 and FS5 are obtained by BEM, FS1, FS2 and FS3 are design requirements.

Two different parametrizations of the noise signals have been defined as output targets in order to highlight the most important spectral characteristics and allow an easy identification of the modelisation error:

- Noise Spectra Parametrization 1 (NSP1), consisting in the central frequency and level of the peak in the noise spectrum associated to the resonance of the cavitating vortices [1], [14];
- Noise Spectra Parametrization 2 (NSP2), consisting in the radiated noise levels in one-third octave band representation [18].

In brief, NSP1 only account for one of the most important characteristics of the noise spectrum from a practical point of view, instead NSP2 is the whole spectrum in one-third octave bands. The different FS and NSP are summarized in Table I.

[1] See Section III-B

TABLE I
DATASET VARIABLES DESCRIPTION.

Symbol	um	Description	Tensor size if not scalar	FS or NSP	PM[1]
P/D		Pitch ratio		Geometry (FS1)	
D	m	Diameter			FS0
BAR		Blade area ratio			
Z		Number of blades			FS0
c/D		Chord ratio at 0.7R			FS0
$t_{\max}/c$		Blade maximum thickness at 0.7R			
f_{max}/c		Blade maximum camber at 0.7R			
α_S	deg	Shaft inclination angle		Working Parameters (FS1)	
J		Advance coefficient			
K_T		Thrust coefficient			
$10K_Q$		Torque coefficient			
V_a	m/s	Advance velocity			FS0
n		Propeller rotation			FS0
σ_v		Cav. index ref. to advance velocity			
σ_n		Cav. index ref. to rotational speed			FS0
σ_{tip}		Cav. index ref. to resultant speed at blade tip			
$\overline{w}$		Average w		Wake Parameter (FS1)	
max w_{07}		Wake maximum at 0.7R			
Wwd_{07}		Wake width at 0.7R			
$D_\theta W\|_{07}^{-}$		Left wake gradient at 0.7R			
$D_\theta W\|_{07}^{+}$		Right wake gradient at 0.7R			
max w_{09}		Wake maximum at 0.9R			
Wwd_{09}		Wake width at 0.9R			
$D_\theta W\|_{09}^{-}$		Left wake gradient at 0.9R			
$D_\theta W\|_{09}^{+}$		Right wake gradient at 0.9R			
$\overline{\alpha}_{G07}$	deg	Circumferential average α_G at 0.7R		Angle of Attack (FS1)	
min α_{G07}	deg	Minimum α_G at 0.7R			
max α_{G07}	deg	Maximum α_G at 0.7R			
$\theta\|max\ \alpha_{G07}$	deg	Angular position of maximum α_G at 0.7R			
$\overline{\alpha}_{G09}$	deg	Circumferential average α_G at 0.9R			
min α_{G09}	deg	Minimum α_G at 0.9R			
max α_{G09}	deg	Maximum α_G at 0.9R			
$\theta\|max\ \alpha_{G09}$	deg	Angular position of maximum α_G at 0.9R			
w		Axial wake	360×31	Wake (FS2)	
α_G	deg	Geometric angle of attack	360×31	Angle of Attack (FS3)	
C_P		Coefficient of pressure on blade	$44\times25\times60$	BEM (FS4)	FS0[2]
$\mathcal{C}$	m^2/s	Blade circulation	60×25	BEM (FS5)	FS0[2]
f_c	Hz	Central peak frequency		NSP1	✓
RNL_c	dB	Noise level at f_c		NSP1	✓
$RNL_{1/3}$	dB	Noise Levels in one-third octave bands	24	NSP2	✓[3]

III. MODELIZATION

In the proposed context, namely the estimation of the variables of the different noise spectra parametrizations based on a series of input variables characterising the propeller (see Table I), a general modelization framework can be defined, characterised by an input space $\mathcal{X}$, an output space $\mathcal{Y}$, and an unknown relation $\mu : \mathcal{X} \rightarrow \mathcal{Y}$ to be learned. In the specific case, $\mathcal{X}$ is composed by the FS reported in Table I. In particular, the FS is composed by a series of scalars (FS1), a series of two-dimensional tensors (FS2, FS3, and FS5) and a three-dimensional tensor (FS4). On the other hand, the output space $\mathcal{Y}$, depends on the chosen parametrization (NSP1 and NSP2). In this context, the authors define as model $h : \mathcal{X} \rightarrow \mathcal{Y}$ an artificial simplification of μ. The model h can be obtained with different kinds of techniques, for example requiring some physical knowledge of the problem, as in PMs, or the acquisition of large amount of data, as in DDMs, or both of them, as in HMs.

A. Performance Measures

Independently of the adopted technique, any model h requires some data to be tuned (or learned) on the problem specificity and to be validated (or tested) on a real-world scenario. For these purposes, two separate sets of data $\mathcal{D}_n = \{(X_1, Y_1), \cdots, (X_n, Y_n)\}$ and $\mathcal{T}_m = \{(X_1^t, Y_1^t), \cdots, (X_m^t, T_m^t)\}$, where $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, need to be exploited, to respectively tune h and evaluate its performances. It is important to note that $\mathcal{T}_m$ is needed since the error that h would commit over $\mathcal{D}_n$ would be too optimistically biased since $\mathcal{D}_n$ has been used to tune h.

Note that, in this specific case, $Y \in \mathcal{Y}$ is not a simple scalar but it is a more complex structure since it is a characterisation of the noise spectra. Let us suppose, for a moment, that $\mathcal{Y} \subseteq \mathbb{R}$, namely $Y \in \mathcal{Y}$ is a scalar. In this case, the error that h commits on $\mathcal{T}_m$ in approximating the real process is usually measured with reference to different indexes of performance [14]

- the Mean Absolute Error (MAE) is computed by taking the absolute loss value of h over $\mathcal{T}_m$
$$\text{MAE}(h) = \frac{1}{m} \sum_{i=1}^{m} |h(X_i^t) - Y_i^t|;$$
- the Mean Absolute Percentage Error (MAPE) is computed by taking the absolute loss value of h over $\mathcal{T}_m$ in percentage
$$\text{MAPE}(h) = \frac{100}{m} \sum_{i=1}^{m} \left| \frac{h(X_i^t) - Y_i^t}{Y_i^t} \right|;$$
- the Pearson Product-Moment Correlation Coefficient (PPMCC) measures the linear dependency between $h(X_i^t)$ and Y_i^t with $i \in \{1, \cdots, m\}$
$$\text{PPMCC}(h) = \frac{\sum_{i=1}^{m} (Y_i^t - \overline{Y})(h(X_i^t) - \hat{\overline{Y}})}{\sqrt{\sum_{i=1}^{m} (Y_i^t - \overline{Y})^2} \sqrt{\sum_{i=1}^{m} (h(X_i^t) - \hat{\overline{Y}})^2}},$$
where $\overline{Y} = \frac{1}{m} \sum_{i=1}^{m} Y_i^t$ and $\hat{\overline{Y}} = \frac{1}{m} \sum_{i=1}^{m} h(X_i^t)$.

Other state-of-the-art measures of error exist (such as R-squared and or the Mean Square Error) but, from a physical point of view, they give a complete description of the quality of the model and adding more measures would make the results less readable.

Unfortunately, in the proposed application, $\mathcal{Y} \subseteq \mathbb{R}^p$ is a vector representing a parametrization of the cavitation noise spectra. For this reason, to provide a value which reasonably represents the error that h commits on $\mathcal{T}_m$ when $Y \in \mathcal{Y}$ is a parametrization of the cavitation noise spectra, it is necessary to use a more suitable error measure which targets one of the particular parametrizations (NSP1 and NSP2).

For what concerns NSP2 the authors will redefine the MAPE as the average MAPE among the different parameters which compose the parametrizations. This can be done since NSP2 count homogeneous quantities (see Section II) and the average MAPE well describes the quality of h, since it represents the average difference between actual and predicted spectra.

For what concerns NSP1, it represents just a point of the spectrum, therefore it is not possible to define the MAPE based on the distance between actual and predicted spectra for this target. In this case, the MAPE is defined as the average of the MAPE computed for the two parameters composing NSP1, namely the frequency and the level of the spectral hump.

B. Physical Models (PMs)

PMs are derived from the physical theories and relevant equations describing the phenomenon of interest, under certain assumptions making the solution of the equations affordable without the need for sophisticated computational tools. Although the structure of these formulations is derived from physical equations, experimental data are used to tune the formulations to deal with real world problems. For this reason, these methods are usually referred to also as semi-empirical methods. Two main physical models are considered in present

[2] The variable has been used to compute a feature for a PM

[3] In this case just the last elements of the vectors can be estimated with the PMs, see Section III-B.

work: the first one models the noise generated by the resonance of a cavitating tip vortex, the second regards the noise due to sheet cavitation.

a) Cavitating vortices noise: The resonance of the tip vortex cavity is assumed to be the main responsible for the low frequency hump typically observed in propeller noise spectra, when vortex cavitation is present. Therefore, this model is used to predict the central frequency f_c and noise level RNL_c of the centre peak (NSP1).

The formulation adopted to describe the relationship between the pulsation frequency of a cavitating vortex and its radius can be derived from the analysis of the pulsation of a single bubble of gas immersed in an infinite fluid domain. The relation takes the following form

$$\frac{f_\mathrm{c}}{nZ} = c_1 \frac{1}{r_c/D} \frac{\sqrt{\sigma_{tip}}}{Z} + c_2 \,, \tag{1}$$

where r_c is the vortex radius, Z is the number of blades, σ_{tip} is the cavitation number evaluated at the blade tip, c_1 and c_2 are unknown constants whose values can be determined by fitting on experimental data, as done in [21].

The amplitude of the noise generated by vortex pulsation depends again on the cavity radius. The formulation used in present work has been derived following an approach similar to that proposed by [5]. The original formulation gives the noise level in dB as a function of some parameters describing the propeller and its functioning condition. In the formulation used in present work, the noise level in dB is given as a function of the vortex cavity radius, the propeller diameter and the number of blades, as shown in Eq. (2)

$$\mathrm{RNL}_\mathrm{c} = a_p + 20\log_{10}\left[\left(\frac{r_c}{D}\right)^k \sqrt{Z}\right] \,, \tag{2}$$

where a_p and k are again unknown constants to be found through data fitting.

Both formulations used to compute the frequency and the noise amplitude of vortex pulsation require the knowledge of the cavity radius. This radius can be found, in first approximation, as the radial distance r from the vortex axis, where the local pressure equals the water vapour pressure; the local pressure can be determined if the velocity field around the vortex is known and this requires the use of a vortex model.

Within the several vortex models available in the literature, the one described in [22] has been used to estimate the radial distribution of the azimuthal velocity component. This model is based on the assumptions of 2-D and axisymmetric vortical flow, implying constant axial velocity, negligible radial velocity with respect to the tangential velocity, and constant velocity distribution in the azimuthal direction. According to this model, the azimuthal velocity distribution is given as $v(r) = f(\mathcal{C}_\infty, \beta, r_\nu, p, D, \zeta, r)$, where $\mathcal{C}_\infty$ is the vortex strength, ζ is a prescribed parameter taking the value of 1.2564, and p has been fixed at 0.75 as suggested by [22].

The viscous core radius r_ν is an input and differ for every propeller and functioning condition. Since direct measurements are not available for present cases, literature values reported in [23] have been exploited. These values have been scaled according to the procedure proposed by [24] to take into account the different Reynolds number.

Eventually, the vortex strength $\mathcal{C}_\infty$ is needed to compute the vortex velocity distribution using Proctor's formula. In this work, this quantity has been borrowed by BEM bound circulation in correspondence to the blade section at $r/R = 0.95$. The coefficient β has to be found by least square fitting on the experimentally measured f_c.

At this point, the azimuthal velocity distribution in the vortex radial direction is known. Under the previously mentioned hypotheses, the pressure distribution can be predicted integrating the momentum equation in the radial direction [25], therefore the cavitating radius can be found.

In summary, the structure of the whole physical model for the tip vortex noise is represented in Figure 1.

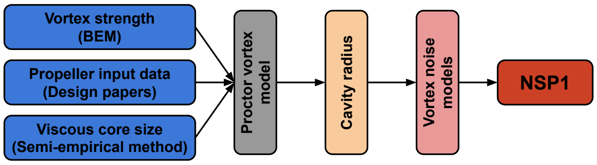

Vortex strength (BEM)
Propeller input data (Design papers)
Viscous core size (Semi-empirical method)
Proctor vortex model
Cavity radius
Vortex noise models
NSP1

Fig. 1. Schema of the tuning algorithm for the prediction of the vortex peak.

b) High frequency noise: The second model is used to predict the contribution of sheet cavitation that is assumed to dominate measured noise spectra in the frequency range 10 kHz to 80 kHz. The model chosen is Brown's empirical formula discussed in [6].

$$L = K + \eta \,\left[10\, log_{10}(ZD^4 n^3 f^{-1}) + 10\, log_{10}\left(\frac{A_C}{A_D}\right)\right] \,, \tag{3}$$

where A_D is the propeller disk area, f is the frequency and L is the noise dB level. The cavitating area A_C has been derived from the distribution on the blades of the pressure coefficient C_p, assuming that cavitation is present where the drop in local pressure is greater than the reserve of static pressure ($C_p < -\sigma_n$). This approximation neglects cavities dynamics and the real extent of cavitation, therefore the predicted area is expected to be underestimated.

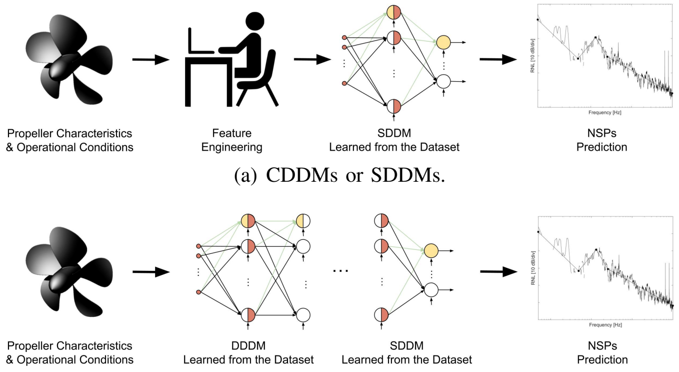
In the formula, the unknown parameters are the constants K and η which have been found by fitting to the experimental RNLs.

C. Data Driven Models (DDMs)

In this section the authors will present the proposed DDMs for predicting the different NSPs based on the different FSs (see Table I) exploiting the data presented in Section II.

Even if the scenario of this paper is slightly different with respect to the one of [14] (see Section I), a first idea could be to exploit the same methodology for defining new DDMs and HMs and adapt it to the scope of this work. Unfortunately, for the reasons that will be clarified in this section [17], [26], this approach would result in very low performance, in terms of accuracy, as will be shown in the experimental results of Section IV-B. The reasons behind this decay in performance need to be searched in the philosophy behind the methodology proposed in [14] that, from now on, will be named as Conventional DDMs (CDDMs) or Shallow DDMs (SDDMs). CDDMs relies on the simple pipeline presented in Figure 2(a):

- from the available inputs, i.e. the propeller characteristics and operational conditions, the raw information about the FSs of Table I are extracted;
- from the raw FSs, experts of the problem together with data scientists extract a series of rich features, that should be able to provide all the information about the desired



(a) CDDMs or SDDMs.

(b) ADDMs or DDDMs.

Fig. 2. Conventional (Shallow) DDMs vs Advanced (Deep) DDMs.

output, in this case the different parameters of the NSPs (this process is called Feature Engineering);

- a functional form of the predictive model, the SDDM, is defined by the data scientist. Then the parameters of the SDDMs are learned from the input/output samples, called dataset, where the input is coded with the features engineered in the previous step;
- finally the learned model can be exploited to make prediction about the NSPs.

This approach is very effective under a simple, but quite strict, assumption: the feature engineered by the experts should be rich enough to describe the phenomena, but characterised by a cardinality not too high compared to the number of samples of the dataset [17], [26]. If only FS1 is available, somehow analogously to the work of [14], then CDDMs would be the correct choice. As a matter of fact, in this case also FS2, FS3, FS4, and FS5 are available, and exploiting the CDDMs of [14] would result in an exploding number of features, because of the FSs intrinsic cardinality, and the model would be not able to learn the correct model with a dataset of limited cardinality like the one available for this study (see Section II). Moreover, the extraction of rich and representative features from the FS2, FS3, FS4, and FS5 is a complex task not suited for a human expert.

For these reasons, in this paper, the authors will exploit an Advanced DDMs (ADDMs) or Deep DDMs (DDDMs). ADDMs rely on the pipeline presented in Figure 2(b)

- as for the CDDMs, from the available inputs, (propeller characteristics and operational conditions), the raw information about the FSs of Table I are extracted;
- contrarily to the CDDMs, experts of the problem together with data scientists do not perform a Feature Engineering phase, but they define a functional form of the model, namely a structure of the model, to be learned from the data. This structure is composed of two levels: a first level (DDDMS) is dedicated to learning the features to be provided to the same SDDM exploited for the CDDMs;
- from the dataset, both the SDDM and the DDDM parameters are learned;
- finally the learned model can be exploited to make prediction about the NSPs.

The main differences between the CDDMs and the ADDMs rely on the fact that, in the ADDMs there is just minimal intervention of experts and data scientists in the definition of the model. In fact, in ADDMs, as it will be shown in this section, just the functional form of the features must be designed, while in CDDMs the features are basically handmade, everything else is learned from the dataset.

In this work, the CDDMs is not described in details, nevertheless, all the relevant details are reported in the original work of [14]. For completeness, the authors just recall that the CDDMs proposed in [14] is a combination of features engineered by experts [14] plus a Kernel Regularised Least Squares (KRLS) [27] model plus a Feature Reduction phase [28] plus an advance Model Selection phase [29].

Instead, from now on, a detailed description of the proposed ADDMs is reported, starting from the basic principles that guided the definition of the proposed functional form until the final proposed model. In particular, the authors will first explain the building blocks of the proposed ADDM, and then they will show how to combine them to derive the proposed architecture and solve the problem faced in this work. For simplicity, if not specified otherwise, the proposed ADDMs is simply referred as DDMs since, as it will be shown in Section IV-B, the ADDMs are the most effective ones for the purpose of this paper.

The architecture of the proposed DDMs will be built incrementally to explain the different choices.

For what concerns the FS1, this FS is somehow analogous to the one of [14] and for this reason a simple shallow neural network is enough [17] since it has the same expressivity of the approach defined in [14]. In order to limit the number of weights to be learned, the hidden layer will be a simple Random Layer, namely a sort of Extreme learning machine (ELM) [17], [30], [31], with a hyperbolic tangent activation function to provide the necessary non-linearity, and the output layer will be a simple Dense Layer with linear activation function and the L2 regularisation to limit the overfitting [17]. This structure basically emulates, with much fewer parameters to tune, the one proposed in [14]. The hyperparameter to be tuned are just the number of neurons of the hidden layer $n_{\text{RL}} \in 2^{\{2,4,6,8,10\}}$ and the amount of regularisation defined by $\lambda \in 10^{\{-4.0,-3.5,\cdots,+3.0\}}$ in the output layer since the number of inputs is defined by FS1 and the number of output neurons is defined by the particular NSP to be predicted. The initialisation of the output Dense Layer is a simple zero-valued initialisation.

For what concerns instead FS2, FS3, FS4, and FS5, the process is a bit more complicated. The authors will first present the proposed method for dealing with the 2D-tensors (FS2, FS3, and FS5) and the treatment of the 3D-tensor (FS4) will be just summarised because analogous. As already mentioned before and deeply explained in [17], the 2D-tensors cannot be simply stacked with FS1 by means of a Concatenation Layer [17] and fed to a classical neural network. For this reason, a more condensed representation of these FSs needs to be learnt, and, for this purpose, the convolution layers is the best choice [17]. The only problem of the Convolutional Layers is that, based on the setting of their parameters, they are designed to react to just a particular scale of dimension and for this reason it would be good to have more layers which react to different scales [17]. The solution that has been adopted in this paper is to use and Inception Layer [32] composed of three parallel Convolutional Layers (equipped with linear activation functions to mitigate the gradient vanishing effect, that will be clarified later, and no regularisation because of the intrinsic sparsity of the architecture) reacting to different scales. In order to limit the number of weights to be learned, one of the three Convolutional Layers is a simple 2D Max Pooling Layer [17]. Then, in order to agglomerate all the information at different scales and produce a condensed representation, the outputs of the two 2D Convolutional and the 2D Max Pooling Layers are combined adopting a Concatenation Layer and then exploiting a Dense Layer (equipped with linear activation functions [17], again to mitigate the gradient vanishing effect [17], and dropout as regulariser [17]). This building block is depicted in Figure 3.

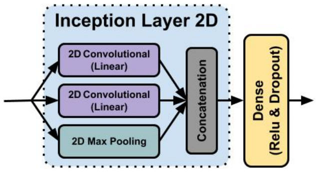


Fig. 3. Proposed architecture for extracting a good representation from the two-dimensional tensors (see FS2, FS3, and FS5 in Table I) in the dataset described in Section II.

The convolution layer has multiple hyperparameters that have to be tuned [17]. Suppose that the input space, being a 2D-tensor, is a matrix of size $\mathbb{R}^{q_1 \times q_2}$. The patch, or the size of the filter to be learned, is $\mathbb{R}^{\lfloor r_f q_1 \rfloor_{\text{odd}} \times \lfloor r_f q_2 \rfloor_{\text{odd}}}$ where $r_f \in (0, 1)$ is an hyperparameter which regulates the ratio between the size of the 2D-input tensor and the filter while $\lfloor \cdot \rfloor_{\text{odd}}$ represents the closer smaller odd number. The padding is the addition of elements at the border of the tensor to mitigate the edge effects, and its size. The stride is the movement step of the filter on the tensor which is $\lfloor r_s q_1 \rfloor$ along the first dimension of the tensor and $\lfloor r_s q_2 \rfloor$ along the second dimension, $r_s \in (0, 1)$ is and hyperparameter which regulates this movement. The dilation is a further sparsity capability of the filter reaction, which is $\lfloor r_d q_1 \rfloor$ along the first dimension of the tensor and $\lfloor r_d q_2 \rfloor$ along the second dimension, $r_s \in (0, 1)$ is and hyperparameter which regulates it. In our case, for the padding, a zero padding has been exploited. It is now necessary to tune for the 2D Convolutional (C2D_1, C2D_2, and MP2D_1) and the 2D Max Pooling Layers the r_f ($r_f^{\text{C2D}_1}, r_f^{\text{C2D}_2}, r_f^{\text{MP2D}_1} \in \{0.1, 0.2, 0.4\}$), the r_s ($r_s^{\text{C2D}_1}, r_s^{\text{C2D}_2}, r_s^{\text{MP2D}_1} \in \{0.1, 0.2, 0.4\}$), and the r_d ($r_d^{\text{C2D}_1}, r_d^{\text{C2D}_2}, r_d^{\text{MP2D}_1} \in \{0.1, 0.2, 0.4\}$). Then, for the dense layer, it is necessary to tune the number of neurons $n_{\text{DL}} \in 2^{\{2,4,6,8,10\}}$ and the dropout rate $r_d \in 10^{\{-3,-4,-2,-1\}}$, namely the number of neurons to randomly deactivate during training [17]. The problem of this architecture is its initialization phase since a deterministic or random initialization would be not sufficient to guarantee good performances [17]. For this reason the authors initialize, or more precisely pre-trained, the architecture of Figure 3 with a surrogate problem, using the autoencoders approach [17]. Basically, since the output of the dense layer in Figure 3 should be a good and condensed representation of the FSs (FS2, FS3, and FS5), based on that representation it should be possible to retrieve the original FSs. Subsequently, the weights have been initialised using the approach proposed in [33], the authors attach to the Dense Layer of Figure 3 another Dense Layer where the outputs are the same FS provided to the block as input, and finally the network is trained using the algorithms that will be explained later in this section. The architecture of the autoencoder for pre-training the block of Figure 3 is depicted in Figure 4. After this pre-training phase the last Dense Layer added for the pre-training is removed, and the Inception Layer plus the dense layer after that have been kept.

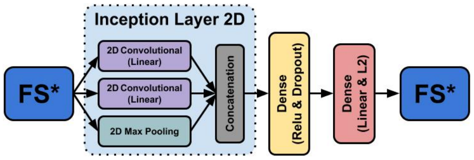


Fig. 4. Architecture of the autoencoder for initialising the architecture presented in Figure 3.

The extension of this 2D block defined for FS2, FS3, and FS5 can be trivially extended to the case of FS4 where a 3D block needs to be developed.

At this point, it is possible to combine all outputs of the blocks developed for FS2, FS3, FS4, and FS5 in a Concatenation Layer together with FS1 and fed them to the same SDDMs described for FS1. It is possible to do perform this action since FS1 plus the outputs of the blocks developed for FS2, FS3, FS4, and FS5 is an informative and condensed information about all the features. The resulting architecture, namely the proposed DDM, is depicted in Figure 5.

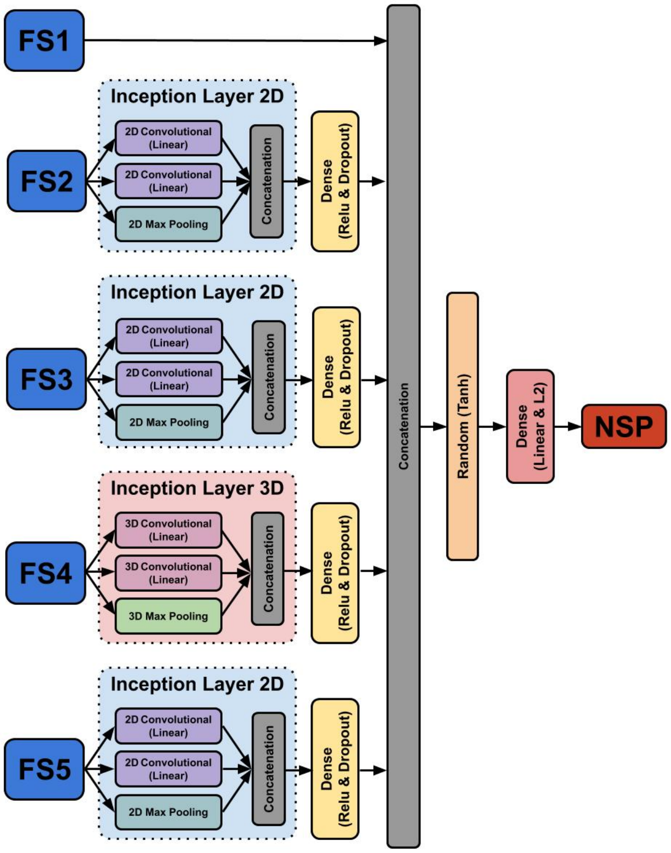


Fig. 5. Proposed DDM architecture.

What still needs to be described is how the network has been trained (or pre-trained the blocks just described for FS2, FS3, FS4, and FS5). As described before, many gradient descend-based algorithms [17] (e.g. SGD, RMSprop, Adagrad, Adadelta, Adam, etc.) exist for solving the problem. The only issue of these algorithms is the Gradient Vanishing effect [17], namely the fact that in deep network the gradient tends to go to zero exponentially in the number of layers.

For this reason, in the proposed architecture, the authors have exploited, in all the trained layers, linear or RELU activation functions which mitigate this problem [17]. Then, the Mini Batch Stochastic Gradient Descent (SGD) algorithm has been used, characterised by three hyperparameters [17]: learning rate of the gradient, momentum that accelerates SGD in the relevant direction, and batch-size of each iteration.

The last problem that authors need to solve is how to tune the hyperparameters of the proposed architecture. Since all DDMs are characterised by a set of hyperparameters $\mathcal{H}$ influencing their ability to estimate μ, a proper Model Selection (MS) procedure, namely the process of tuning them to achieve optimal performances, needs to be performed [29]. Several methods exist for MS purpose but resampling methods, like the well-known k-Fold Cross Validation (KCV) or the nonparametric Bootstrap (BTS) approaches represent the state-of-the-art MS approaches when targeting real-world applications [29]. Resampling methods rely on a simple idea: the original dataset $\mathcal{D}_n$ is resampled once or many (s) times, with or without replacement, to build two independent datasets called training, and validation sets, respectively $\mathcal{L}_l^i$ and $\mathcal{V}_v^s$, with $i \in \{1, \cdots, s\}$. Note that $\mathcal{L}_l^i \cap \mathcal{V}_v^i = \oslash$ and $\mathcal{L}_l^i \cup \mathcal{V}_v^i = \mathcal{D}_n$. 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 proposed architecture or, in other words, to perform the MS phase, the following procedure has to be applied

$$\mathcal{H}^* : \ \arg\min_{\mathcal{H} \in \mathfrak{H}} \quad \frac{1}{s} \sum_{i=1}^{s} \hat{\mathrm{L}} \left(h^*_{\{\mathcal{H}, \mathcal{L}_l^i\}}, \mathcal{V}_v^i \right) \tag{4}$$

where $h^*_{\{\mathcal{H}, \mathcal{L}_l^i\}}$ is the model with its set of hyperparameters $\mathcal{H}$ learned with with the data $\mathcal{L}_l^i$. Since the data in $\mathcal{L}_l^i$ are independent of the ones in $\mathcal{V}_v^i$, the idea is that $\mathcal{H}^*$ should be the set of hyperparameters which allows achieving a small error on a data set that is independent of the training set. In this work, the authors will exploit the BTS procedure and consequently $s = 100$, if $l = n$ and the resampling must be done with replacement [29].

D. Hybrid Models (HMs)

The problem that authors want to address here is how to construct a model able to both take into account the physical knowledge about the problem encapsulated in the PMs of Section III-B and the information hidden in the available data described in Section II as the DDMs of Section III-C. For this reason the proposed HM is a combination of the PMs and the DDMs.

In order to reach this goal different approaches exist (see e.g. [14], [34], [35]) but all these methods have been developed in the context of conventional DDMs (like the KLRS mentioned above) and not for advance DDMs (es the one based on DNNs described in Section III-C). In fact, for conventional DDMs there are many ways of including the knowledge encapsulated in the PMs. For example, in [34], [35] authors simply add to the input space of the DDMs the prediction of the PMs, while in [14], [35] authors tried to build a model able to contemporary learn the target task and how the PMs behave.

These different flavours of HMs, for conventional DDMs, are due to the fact that the model functional form for conventional DDMs cannot be arbitrary modified without compromising their ability to effectively and efficiently learn from data or weakening they theoretical properties [17], [26]. Vice versa, the architecture of the advanced DDMs based on DNNs described in Section III-C can be easily and almost arbitrary modified to meet the requirements of the particular application. Moreover, different ways of changing the architecture may results in the same effect because of the nature of the functional form of these DDMs, and for this reason the simplest solution can be chosen.

For example, in the case under examination, it could be possible to change the architecture of the proposed DDM depicted in Figure 5 using the two main different philosophies introduced in [14], [34], [35] which consists in

I changing the FS, namely the input space, or
II force the DDMs to learn contemporary the NSPs and the PMs, namely change the output space.

For what concerns the Option II the modification is trivial while the Option I is not as much trivial as it may seem since it is required to define where and how the prediction of the PMs should be fed to the DDM. Since the PMs already provide a good approximation of the propeller characteristic, in this particular case an actual NSP approximation, the most natural choice would be to consider this information at the same level of the FS1 that need to be fed to the layer which condensates all the information about the different FSs in order to improve its representativity. But such a choice is somehow equivalent to change the output space of the DDM since this would result in a consistent change of the last layers of the DDMs (in particular the expressivity, of size, of the random layer) [17]. Since these two modifications, in the proposed DDM, would have a similar effect, the authors decide to use the Option I since it affects more directly the last layers, not influencing the other ones. The result is the HM architecture depicted in Figure 6, where the authors just underline, for simplicity, the differences between the DDMs of Figure 5 and the proposed HM.

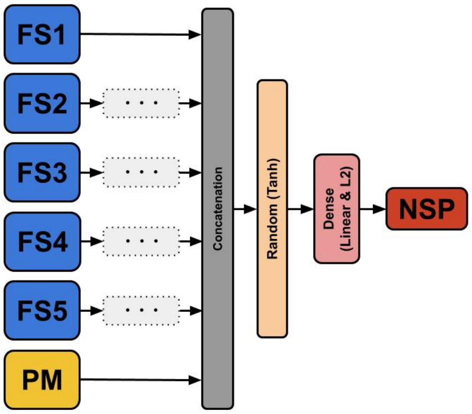


Fig. 6. DNN-based HM architecture (see Figure 5 for the missing pieces).

Note that the HMs can be built just for the NSP for which a PM is provided, able to estimate all, or just a subset, of the parameters of the NSP. Hence, in this case, the HM can be defined just for all the NSPs (see Table I).

Note also that the hyperparameters of the architecture, also for the HMs, need to be tuned with the same procedure described in Section III-C for the DDMs.

IV. Experimental Results

In this section, the performances of the PMs, DDMs, and HMs, (see Sections III-B, III-C, and III-D) are tested and compared by means of the data described in Section II and the performance measures described in Section III-A. With respect to [14], just the extrapolation scenario is considered since it is the most interesting and challenging.

For what concerns the PM, please recall that PMs are just able to predict a subset of the parameters of the different NSPs, while the DDMs are able to predict all the targets. HMs differ from the DDMs every time the PMs are able to predict the spectral parametrization or a part of it. The set of hyperparameters tuned during the MS phase are the same as those of the DDMs.

All the tests have been repeated 30 times and the average results are reported, together with their t-student 95% confidence interval, in order to ensure the statistical consistency of the results.

A. Scenario

The extrapolation scenario has been studied [14]: in this scenario, the models try to predict the propeller noise in groups of loading conditions characterized by different cavitation extents with respect to those exploited for building the model. These groups of loading conditions are defined based on the combination of thrust coefficient K_T and cavitation number σ_n, as exemplified in Figure 7 on a typical cavitation bucket: the cavitation extent of interest grows moving down from the Suction Side (S.S.) vortex inception line and, for the same σ_n, increasing the K_T. Basically the scenario defines how $\mathcal{D}_n$ and $\mathcal{T}_m$ are built, that are the subset of data exploited for building and testing the models. The extrapolation scenario tests the capability of the models to predict radiated noise for cases not included in the variable domain of the data used to build them. The practical application of this scenario is the prediction of noise for those full scale propeller operational conditions which cannot be consistently reproduced at model scale due to viscous scale effects, as explained in [14]. In order to obtain an indication of the extrapolation performance, the authors included in $\mathcal{D}_n$ samples of only five of the six groups of different operational conditions and use the sixth group as $\mathcal{T}_m$.

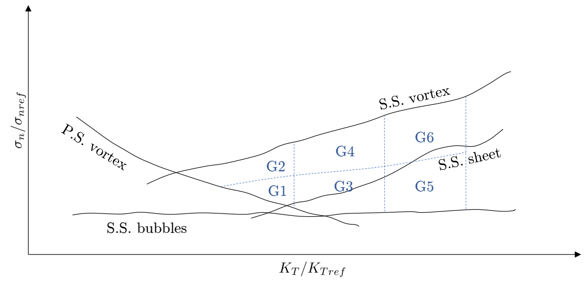


Fig. 7. Extrapolation groups.

B. CDDMs vs ADDMs

In order to compare the CDDMs (like the KRLS-based DDMs introduced in [14] and recalled in Section III-C) and the ADDMs proposed in this work, the authors have first reported in Table II the errors measured with the MAPE in the proposed extrapolation scenario with different FSs for the different NSPs.

From the results reported in Table II, the CDDMs show comparable performances to ADDMs only when the FS1 is considered alone, hence when the cardinality of the FS is not too high with respect to the number of samples. If multidimensional features are considered, the error for CDDMs is usually higher with respect to ADDMs. The worst performances for the CDDMs are attained for FS1+FS4 (FS4 is the single feature with the higher cardinality) and for FSAll. The lowest error for the CDDMs, when multidimensional features are considered, is in the case FS1+FS5 probably because FS5 (the blade circulation) account for important information on blade loading with smaller dimensions than FS4. Conversely, the minimum prediction error is obtained, for the ADDMs, when all the features are considered.

In summary, results suggest that only by using the ADDMs it is possible to improve the performance of the model exploiting more rich and complex features. On the other hand, it must be noted that the errors obtained with the CDDMs and the simplest feature set FS1 are only a few percentages higher than the minimum errors obtained with the ADDMs. Notwithstanding this, the improvements achieved exploiting the ADDMs with more complete and physically meaningful features are significant.

C. PMs vs DDMs vs HMs

From now on, for the sake of simplicity, the proposed ADDMs will be referred as DDMs since, as reported in Section IV-B, the ADDMs are the most effective ones for the scopes of this paper. In order to compare the PMs, DDMs, and HMs Table III reports the errors measured with the MAPE with different FSs for the different NSPs. Note that the PMs are able to fully predict only the NSP1 and it always uses only FS0. Instead, since the PMs is able to predict part of all the NSP2 it is always possible to build the HMs.

The results highlight the limits of the PMs, which are not able to accurately predict all the trends present in the experimental data. Nevertheless, the information enclosed in these simple formulations allows to improve the performance of DDMs, as confirmed by the results obtained for the HMs. The improvement with respect to the simple DDMs is more significant when all the features are considered where the MAPE for the HMs is about 2% lower than the DDMs. This result agrees with the higher capabilities of HMs to generalise, thanks to the information encapsulated in the PMs.

D. The Effect of Using the Different FSs on the DDMs and the HMs

From the results reported in Table III, it is also possible to understand the effect of using the different FSs on the DDMs and the HMs.

The DDMs seem to be able to well exploit these multidimensional inputs. When one multidimensional feature (FS2, FS3, FS4, FS5) is added to FS1, results are generally improved. Surprisingly, the effects of the different multidimensional features are all rather similar, preventing to rank these features based on their importance. For the DDMs, the best performances are achieved when all the possible features are considered. However, the absolute improvement with respect to the use of one single multidimensional feature added to FS1 is not large given the low errors.

The same patterns are reflected in the HMs (Table III). If all the features are considered, the HMs show the best performances among all the considered models.

E. The Best PMs, DDMs, and HMs

In order to better detail the quality of the best PMs, DDMs, and HMs in predicting the different parameters of the different NSPs, the errors measured with the MAE, MAPE, and PPMCC, with the best FS according to Table III are reported in Tables IV- V. The word "best" is used to intend the model which produces the best accuracy, or lower error, for each type, according to Table III.

The best MAE for $\mathrm{f_c}$ is below 60 Hz and for $\mathrm{RNL_c}$ it is lower than 1 dB, both for DDMs and HMs. In Figure 8 the comparison reported shows the PM, and the best DDM and HM predicting the NSP1. Looking more in detail, it can be noticed that the DDMs/HMs are able to predict cases in which the PM fails, i.e. when FS0 is not sufficient to characterise the samples. Some outliers are visible in the DDMs and HMs: these samples should be investigated and properly treated, e.g. adjusting the position of the peak or removing them from the dataset.

The NSP2 (noise corrected for spherical propagation) is visible in Tables V. Each column refers to a one-third octave band (from 1 to 24) in the range 0.4 kHz to 80 kHz for the DDM/HM, instead the PM is available only for the range 1 kHz to 80 kHz. The different levels are well predicted and no particular trends seem to exist among the different parts of the spectrum.

TABLE II
COMPARISON BETWEEN CDDMS PROPOSED IN [14] AND THE PROPOSED ADDM. TABLE REPORTS THE ERRORS MEASURED WITH THE MAPE WITH DIFFERENT FSS FOR THE DIFFERENT NSPS.

	NSP1		NSP2	
FS	CDDM	ADDM	CDDM	ADDM
1	8.0±0.3	8.0±0.3	7.9±0.3	7.9±0.3
1,2	13.0±0.3	6.9±0.3	12.2±0.3	7.0±0.3
1,3	12.9±0.3	7.1±0.3	12.2±0.3	7.0±0.3
1,4	16.1±0.3	7.2±0.3	14.8±0.3	7.0±0.3
1,5	10.9±0.3	6.9±0.3	10.5±0.3	7.0±0.3
All	19.0±0.3	**6.0±0.3**	17.5±0.4	**6.1±0.3**

TABLE III
COMPARISON BETWEEN PMS, DDMS, AND HMS. TABLE REPORTS THE ERRORS MEASURED WITH THE MAPE WITH DIFFERENT FSS FOR THE DIFFERENT NSPS. NOTE THAT THE PMS ARE ONLY ABLE TO FULLY PREDICT THE NSP1 AND PMS DO NOT CHANGE IF WE CHANGE THE FS SINCE THEY ALWAYS USE JUST A SUBSET OF THE FS1. INSTEAD, SINCE THE PMS ARE ABLE TO PREDICT PART OF THE NSP1 AND NSP2 WE CAN BUILD THE HMS FOR THOSE NSPS.

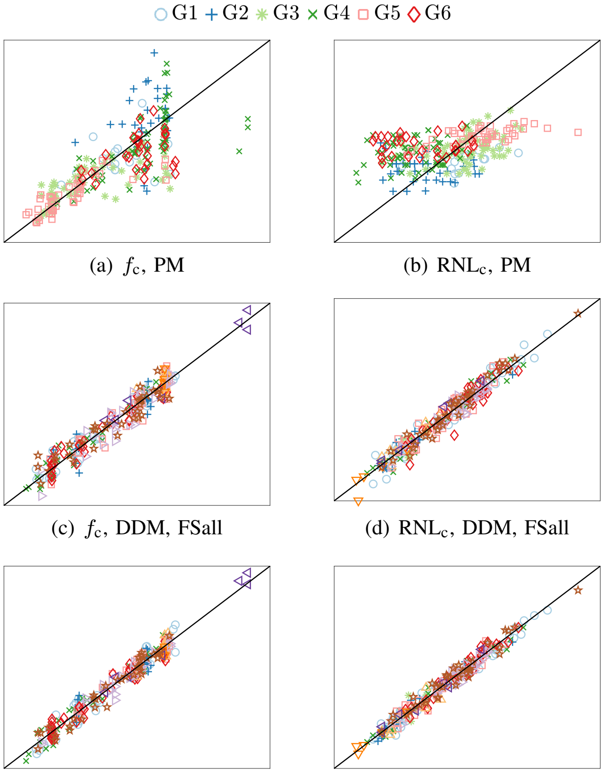
	NSP1			NSP2	
FS	PM	DDM	HM	DDM	HM
1	12.1±1.1	8.0±0.3	8.0±0.3	7.9±0.3	7.9±0.3
1,2		6.9±0.3	6.1±0.3	7.0±0.3	6.2±0.3
1,3		7.1±0.3	5.8±0.3	7.0±0.3	6.2±0.3
1,4		7.2±0.3	6.0±0.3	7.0±0.3	6.1±0.3
1,5		6.9±0.3	5.8±0.3	7.0±0.3	6.1±0.3
All		6.0±0.3	**4.3±0.3**	6.1±0.3	**4.5±0.3**

V. CONCLUSIONS

In this paper, an approach to estimate propeller cavitation noise spectrum by means of deep learning models has been presented. In order to verify a propeller to be compliant with noise requirements only considering the information available at the design stage, the proposed models have been developed exploiting a dataset collected through model scale measurements in a cavitation tunnel, combined with the detailed flow characterization obtained by Boundary Element Method calculations.

TABLE IV
COMPARISON BETWEEN THE BEST PM, DDM, AND HM IN PREDICTING THE DIFFERENT PARAMETERS OF NSP1. TABLE REPORTS THE ERRORS MEASURED WITH THE MAE, MAPE, AND PPMCC WITH BEST FS FOR THE DIFFERENT PARAMETERS OF NSP1. FOR THE PM THE BEST FS IS NOT INDICATED SINCE PM ALWAYS USES JUST A SUBSET OF THE FS1.

		FS*	f_c	$\mathrm{RNL_c}$
MAE	PM	-	206.2±21.4	4.9±0.4
	DDM	all	122.6±5.5	2.1±0.1
	HM	all	**84.7±5.4**	**1.5±0.1**
MAPE	PM	-	10.1±1.0	14.1±1.1
	DDM	all	6.0±0.3	5.9±0.3
	HM	all	**4.1±0.3**	**4.4±0.3**
PPMCC	PM	-	0.76±0.01	0.39±0.12
	DDM	all	**0.99±0.01**	**0.99±0.01**
	HM	all	**0.99±0.01**	**0.99±0.01**



(a) f_c, PM

(b) $\mathrm{RNL_c}$, PM

(c) f_c, DDM, FSall

(d) $\mathrm{RNL_c}$, DDM, FSall

(e) f_c, HM, FSall

(f) $\mathrm{RNL_c}$, HM, FSall

Fig. 8. Comparison between the best PM, DDM, and HM in predicting the different parameters of NSP1 according to Table III. Figure reports the scatter plot (measured values on the x axis and predicted ones on the y axis) with best FS according to Table III for the different parameters of NSP1.

According to the results, the proposed approach is definitely promising, however its potential needs to be further investigated. In particular, the inclusion of detailed flow quantity, although improving the performance of the method in a relative sense, has a somewhat limited effect from a practical point of view. This fact may be partially ascribed to the fact that also simpler modelization approaches allowed obtaining good results in the tested scenario. It must be noticed that, despite the effort spent in building the dataset, it still accounts for a rather limited number of propellers and configurations, preventing the opportunity to verify the performance of the method on fully unseen cases.

Future activities will be therefore dedicated to the enlargement of the dataset and to the investigation of more realistic

TABLE V

COMPARISON BETWEEN THE BEST PM, DDM, AND HM IN PREDICTING THE DIFFERENT PARAMETERS OF NSP2. TABLE REPORTS THE ERRORS MEASURED WITH THE MAE, MAPE, AND PPMCC WITH BEST FS FOR THE DIFFERENT PARAMETERS OF NSP2. FOR THE PM THE BEST FS IS NOT INDICATED SINCE PM ALWAYS USES JUST A SUBSET OF THE FS1 AND IS ONLY ABLE TO PREDICT A SUBSET OF THE PARAMETERS OF NSP2.

		FS	**RNL** $_{\frac{1}{3}}$ (1)	**RNL** $_{\frac{1}{3}}$ (2)	**RNL** $_{\frac{1}{3}}$ (3)	**RNL** $_{\frac{1}{3}}$ (4)	**RNL** $_{\frac{1}{3}}$ (5)	**RNL** $_{\frac{1}{3}}$ (6)	**RNL** $_{\frac{1}{3}}$ (7)	**RNL** $_{\frac{1}{3}}$ (8)	**RNL** $_{\frac{1}{3}}$ (9)	**RNL** $_{\frac{1}{3}}$ (10)	**RNL** $_{\frac{1}{3}}$ (11)	**RNL** $_{\frac{1}{3}}$ (12)
MAE	DDM	all	2.4±0.1	2.0±0.1	2.3±0.1	2.1±0.1	2.1±0.1	2.3±0.1	2.0±0.1	2.0±0.1	1.6±0.1	1.9±0.1	1.6±0.1	1.7±0.1
	HM	all	1.8±0.1	1.4±0.1	1.7±0.1	1.7±0.1	1.5±0.1	1.5±0.1	1.5±0.1	1.5±0.1	1.2±0.1	1.4±0.1	1.2±0.1	1.2±0.1
MAPE	DDM	all	6.9±0.3	6.9±0.3	7.2±0.3	7.0±0.3	6.9±0.3	7.4±0.3	6.6±0.3	7.0±0.3	6.8±0.3	7.0±0.3	6.9±0.3	7.3±0.3
	HM	all	5.1±0.3	4.9±0.3	5.3±0.3	5.4±0.3	5.0±0.3	4.9±0.3	5.0±0.3	5.1±0.3	5.2±0.3	5.1±0.3	5.3±0.3	5.3±0.3
PPMCC	DDM	all	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.98±0.01	0.98±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01
	HM	all	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.98±0.01	0.99±0.01	0.98±0.01	0.98±0.01	0.99±0.01	0.99±0.01

		FS	**RNL** $_{\frac{1}{3}}$ (13)	**RNL** $_{\frac{1}{3}}$ (14)	**RNL** $_{\frac{1}{3}}$ (15)	**RNL** $_{\frac{1}{3}}$ (16)	**RNL** $_{\frac{1}{3}}$ (17)	**RNL** $_{\frac{1}{3}}$ (18)	**RNL** $_{\frac{1}{3}}$ (19)	**RNL** $_{\frac{1}{3}}$ (20)	**RNL** $_{\frac{1}{3}}$ (21)	**RNL** $_{\frac{1}{3}}$ (22)	**RNL** $_{\frac{1}{3}}$ (23)	**RNL** $_{\frac{1}{3}}$ (24)
MAE	PM	-	-	-	14.2±1.1	14.1±1.2	13.8±1.3	14.1±1.2	14.2±1.3	14.1±1.3	14.0±1.3	14.1±1.3	14.3±1.4	14.3±1.4
	DDM	all	1.8±0.1	1.4±0.1	1.6±0.1	2.0±0.1	2.5±0.1	2.2±0.1	2.4±0.1	2.2±0.1	2.3±0.1	2.3±0.1	2.5±0.1	2.4±0.1
	HM	all	1.3±0.1	1.0±0.1	1.2±0.1	1.5±0.1	1.8±0.1	1.8±0.1	1.7±0.1	1.8±0.1	1.7±0.1	1.7±0.1	1.8±0.1	1.8±0.1
MAPE	PM	-	-	-	62.4±4.8	50.7±4.3	39.4±3.6	40.6±3.6	42.7±3.8	43.3±4.0	42.7±4.1	41.6±3.9	40.4±3.9	41.2±4.1
	DDM	all	6.8±0.3	6.7±0.3	6.8±0.3	7.2±0.3	7.0±0.3	6.4±0.3	7.1±0.3	6.8±0.3	6.9±0.3	6.8±0.3	7.0±0.3	6.9±0.3
	HM	all	5.1±0.3	5.0±0.3	5.3±0.3	5.3±0.3	5.2±0.3	5.2±0.3	5.0±0.3	5.6±0.3	5.0±0.3	4.9±0.3	5.1±0.3	5.1±0.3
PPMCC	PM	-	-	-	0.14±0.01	0.15±0.01	0.08±0.04	0.10±0.01	0.08±0.02	0.09±0.11	0.11±0.02	0.10±0.02	0.11±0.07	0.13±0.07
	DDM	all	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01
	HM	all	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01

scenarios, as the prediction of noise for a completely unseen propellers and configuration.

ACKNOWLEDGMENTS

The authors wish to express their gratitude to Fincantieri MM-ARC office, which kindly provided the propeller models used in the present work.

REFERENCES

[1] D. Ross, *Mechanics of Underwater Noise*. Peninsula Publishing, 1976.

[2] IMO, *Resolution MSC. 337(91), Code on Noise Levels on Board Ships*, London, 2012.

[3] Council of European Union, "Directive 2008/56/EC," 2008, https://eur-lex. europa. eu/legal-content/EN/TXT/?uri=CELEX:32008L0056.

[4] J. Bosschers, "Investigation of hull pressure fluctuations generated by cavitating vortices," in *International symposium on marine propulsors*, 2009.

[5] A. E. Raestad, "Tip vortex index-an engineering approach to propeller noise prediction," *The Naval Architect*, pp. 11–15, 1996.

[6] N. A. Brown, "Thruster noise," in *Dynamic Positioning Conference*, 1999.

[7] J. P. Petersen, O. Winther, and D. J. Jacobsen, "A machine-learning approach to predict main energy consumption under realistic operational conditions," *Ship Technology Research*, vol. 59, no. 1, pp. 64–72, 2012.

[8] T. W. P. Smith, E. O'Keeffe, L. Aldous, and P. Agnolucci, "Assessment of shipping's efficiency using satellite ais data," in *International Council on Clean Transportation*, 2013.

[9] F. Cipollini, L. Oneto, A. Coraddu, A. J. Murphy, and D. Anguita, "Condition-based maintenance of naval propulsion systems: Data analysis with minimal feedback," *Reliability Engineering & System Safety*, vol. 177, pp. 12–23, 2018.

[10] D. Calcagni, G. Bernardini, and F. Salvatore, "Automated marine propeller optimal design combining hydrodynamics models and neural networks," in *International Conference on Computer Applications and Information Technology in the Maritime Industries*, 2012.

[11] H. Yang, S. Shen, X. Yao, M. Sheng, and C. Wang, "Competitive deep-belief networks for underwater acoustic target recognition," *Sensors*, vol. 18, no. 4, p. 952, 2018.

[12] W. Soares-Filho, J. M. de Seixas, and L. P. Caloba, "Principal component analysis for classifying passive sonar signals," in *IEEE International Symposium on Circuits and Systems*, 2001.

[13] B. Aktas, "A systematic experimental approach to cavitation noise prediction of marine propellers," Ph.D. dissertation, Newcastle University, 2017.

[14] L. Miglianti, F. Cipollini, L. Oneto, G. Tani, and M. Viviani, "Model scale cavitation noise spectra prediction: Combining physical knowledge with data science," *Ocean Engineering*, vol. 178, pp. 185–203, 2019.

[15] S. Gaggero, D. Villa, and S. Brizzolara, "Rans and panel method for unsteady flow propeller analysis," *Journal of Hydrodynamics, Ser. B*, vol. 22, no. 5, pp. 564–569, 2010.

[16] S. Gaggero and D. Villa, "Cavitating propeller performance in inclined shaft conditions with openfoam: Pptc 2015 test case," *Journal of Marine Science and Application*, vol. 17, no. 1, pp. 1–20, 2018.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[18] ITTC Specialist Committee on Hydrodynamic Noise, "Model-scale propeller cavitation noise measurements," International Towing Tank Conference, Recommended Procedures and Guidelines 7. 5-02-01-05, 2017.

[19] ITTC Propulsion Committee, "Model manufacture, propeller models terminology and nomenclature for propeller geometry," International Towing Tank Conference, Recommended Procedures and Guidelines 7.5-01-02-01, 2008.

[20] Y. L. Young and S. A. Kinnas, "A bem for the prediction of unsteady midchord face and/or back propeller cavitation," *J. Fluids Eng.*, vol. 123, no. 2, pp. 311–319, 2001.

[21] B. Maines and R. E. A. Arndt, "The case of the singing vortex," *Journal of fluids engineering*, vol. 119, no. 2, pp. 271–276, 1997.

[22] F. Proctor, N. Ahmad, G. Switzer, and F. Limon Duparcmeur, "Three-phased wake vortex decay," in *AIAA Atmospheric and Space Environments Conference*, 2010.

[23] S. D. Jessup, "An experimental investigation of viscous aspects of propeller blade flow," Ph.D. dissertation, Catholic University of America, 1989.

[24] Y. T. Shen, S. Gowing, and S. D. Jessup, "Tip vortex cavitation inception scaling for high reynolds number applications," *Journal of Fluids Engineering*, vol. 131, no. 7, 2009.

[25] T. Hommes, J. Bosschers, and H. W. M. Hoeijmakers, "Evaluation of the radial pressure distribution of vortex models and comparison with experimental data," *Journal of Physics: Conference Series*, vol. 656, p. 012182, 2015.

[26] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[27] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[28] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[29] L. Oneto, *Model Selection and Error Estimation in a Nutshell*. Springer, 2020.

[30] L. L. C. Kasun, H. Zhou, G. B. Huang, and C. M. Vong, "Representational learning with extreme learning machine for big data," *IEEE intelligent systems*, vol. 28, no. 6, pp. 31–34, 2013.

[31] J. Tang, C. Deng, and G. B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 4, pp. 809–821, 2016.

[32] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *AAAI Conference on Artificial Intelligence*, 2017.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE international conference on computer vision*, 2015.

[34] A. Coraddu, M. Kalikatzarakis, L. Oneto, G. J. Meijn, M. Godjevac, and R. D. Geertsmad, "Ship diesel engine performance modelling with combined physical and machine learning approach," in *International Naval Engineering Conference and Exhibition*, 2018.

[35] A. Coraddu, L. Oneto, F. Baldi, and D. Anguita, "Vessels fuel consumption forecast and trim optimisation: a data analytics perspective," *Ocean Engineering*, vol. 130, pp. 351–370, 2017.