IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 24, 2021, accepted November 23, 2021, date of publication November 30, 2021,
date of current version December 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3131391

Proof of Concept: Calibration of an Overhead Line Conductors' Movements Simulation Model Using Ensemble-Based Machine Learning Model

HAMDI AMROUN[1], **MEHDI AMMI**[2], **AND FIKRI HAFID**[3,4]
[1]Université Sorbonne Paris Nord at Villetaneuse, 93430 Villetaneuse, France
[2]Université Paris 8, 93200 Saint-Denis, France
[3]Research and Development Team, RTE, 92800 Puteaux, France
[4]MSDA, Mohammed VI Polytechnic University (UM6P), Ben Guerir 43150, Morocco

Corresponding author: Hamdi Amroun (amroun@math.univ-paris13.fr)

This work was supported by the Réseau de Transport d'Electricité (RTE).

ABSTRACT In this paper, we present a new approach to use machine learning (ML) for the calibration of a physical model allowing the reproduction of the vibratory behavior of an overhead line conductor. This physical model known as Strip Theory (ST) has the advantage of being very precise but very complicated and cumbersome in its software operations and manipulations. A second model known as the Wake Oscillator (WO) has been implemented in order to meet the limitations of the ST model. In order to be able to use the WO model instead of the ST model, very heavy manual adjustments are required, which makes its use complicated. Precisely, the WO must be able to generate a time series similar to a time series generated by the ST model. In order to respond to this limitation, a machine learning model known as ENS has been proposed. The machine learning model will therefore take as input the data from the WO model and output the data from the ST model. A series of Machine learning models have been implemented and tested. The ENS algorithm was retained with a best Pearson's linear coefficient of determination (R2 Score) of almost 0.7 and a Root mean square deviation (RMSE) of 7.57e-09. This type of approach therefore makes it possible to calibrate the WO model so that simulations of the behavior of overhead line conductors are carried out only with the WO model.

INDEX TERMS Machine learning, numerical simulation, fluid mechanics, model calibration, overhead line conductor.

I. INTRODUCTION

Overhead line conductors are subject to environmental stresses, especially those resulting from interactions with the wind. The latter produce vibrations that can be problematic for the aging of these physical assets.

Based on this observation, RTE (Reseau de Transport d'Electricité) the French TSO (Transmission System Operator), has launched several research projects aimed at establishing physical models capable of reproducing the vibratory behavior of conductors under such conditions. This research has notably led to the implementation of two models, making it possible to have the movements of a conductor in time and space for a given wind speed [1].

The first model, called strip theory (ST), is based on Computation fluid dynamics (CFD) calculations, which are costly in terms of calculation time and cumbersome in their handling (business software, control of outputs, storage, etc.).

The second model, called the wake oscillator and implemented in an ad hoc tool, is based on a simplification of the coupling terms, avoiding all of the cumbersome associated with the first model.

However, the application of such a simplified model involves the selection of modeling parameters (''hyper-parameters''), which it is a question of calibrating via reference results i.e. the results provided by the ST model. The calibration of such a model therefore involves the comparison

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

of the output signals supplied by the two models, and for which the first model is considered as the reference to be reproduced. This step, classic for setting up a physical model, can prove to be costly in terms of the number of simulations if an optimized strategy is not put in place. This has so far been little covered in Rte's research on the subject, except through basic grid search methods, analogous to [2].

From an academic point of view, the emergence of Machine Learning (ML) methods has prompted physicists to question the improvement and optimization of models through the use of the latter. Indeed, numerical models provide a lot of data for a simulation, which it is regularly a question of being able to summarize in order to draw knowledge on the physical phenomenon studied. In this, and particularly in fluid mechanics, different strategies of combination of physical and ML approaches have been established very recently in the literature in order to make a simplified model correspond to a reference [3]. Among these approaches, two families can be distinguished: intrusive methods involving the introduction into models of a part of ML, non-intrusive methods aiming to improve the output signals of a model set by the ML. The problem addressed here falls into this second category, since a chaining of the two approaches is envisaged, the physical tool (WO) developed by Rte being seen initially as a black box, which means that we will use it as it is.

The objective of this paper is to propose a strategy for using ML to calibrate hyper-parameters of a WO model on simulations from a ST model. The idea is to allow the WO model to self-adapt thanks to the use of a ML model which will take the data of the WO as input and will produce the corresponding predictions as an output, i.e. ST model data. In the end, the model WO will be able to produce the expected outputs without manual calibration of the WO model.

To achieve this, we propose a processing pipeline consisting of three steps: The first step is to collect the necessary data to build a Machine learning model. This is done using the WO model by adjusting its hyperparameters so that the WO model is close to the ST model. At the end of this first step, we will have built a learning set. Then, we will propose a methodology allowing to build our ENS model. Finally, for comparison, a series of models will be tested in order to compare them to our model and choose the best. Obviously, good practice requires that we first build a baseline of learning models and if this baseline is not conclusive, we will therefore have to build a new model, namely our ENS model.

The following part discusses a state of the art on the machine learning approaches most used in fluid mechanics and introduces a benchmark of these approaches with particular emphasis on the calibration of models using machine learning technics.

II. RELATED WORK

The field of fluid mechanics is advancing rapidly, thanks to the availability of growing volumes of data from experiments, field measurements and large-scale simulations at multiple spatiotemporal scales [4].

Machine learning (ML) offers a multitude of techniques for extracting information from data that can be translated into knowledge about the underlying fluid mechanics. Additionally, ML algorithms can increase domain knowledge and automate tasks related to controlling and optimizing underlying flows [5].

Fluid mechanics (liquid or gas form) is a field widely studied in physics. In this area, the Navier-Stokes equation is the most important because it is used to describe the movement of fluids. Thus, a single equation can both simulate a flow of water from a tap and calculate the trajectory of the air. Other equations are also important (notably that of Euler) [6].

If the Navier-Stokes equation models the behavior of fluids, its solution is not the simplest. This comes from the fact that it is nonlinear, depending on the observation environment, turbulence (example: passage of an airplane during a calculation of the trajectory of an air current) make calculations almost impossible [7].

To overcome this difficulty, scientists have chosen to take approximations of the resolutions of this equation. There are several, depending on the constraints posed in terms of reliability or computing time. This does not make it possible to have a model that accurately reflects reality, but remains close enough to it for the solutions to be exploitable, in particular for carrying out simulations. The simulations therefore use the generation of millions of particles to which they apply Navier-Stokes approximations to calculate the motion of each of the particles. Statistical modeling and machine learning can be used in several applications, such as motion prediction, increasing the realism of simulations, etc., [8].

In recent years, many techniques have been developed to process this large volume generated by fluid mechanics, ranging from advanced algorithms for processing and compressing or merging data to fluid mechanics databases. However, the analysis of fluid mechanics data has relied, to a large extent, on domain expertise, statistical analysis and heuristic algorithms [9].

First principles, such as conservation laws, have been the main building blocks of flow modeling in recent years. However, for high Reynolds numbers, scale-resolution simulations using the most important model in fluid mechanics, the Navier - Stokes equations, are beyond our current computational resources. An alternative is to perform simulations based on approximations of these equations (as is often practiced in turbulence modeling) or laboratory experiments for a specific setup. However, simulations and experiments are expensive for iterative optimization, and simulations are often too slow for real-time control. As a result, considerable effort has gone into obtaining accurate and efficient reduced-order models that capture essential flow mechanisms at a fraction of the cost. ML offers new avenues for dimensionality reduction and reduced order modeling in fluid mechanics by providing a concise framework that complements and extends existing methodologies. We distinguish here two complementary contributions: i) reduction of dimensionality and ii) reduced order modeling [10]–[12].

Dimensionality reduction involves the extraction of key features and dominant models that can be used as reduced coordinates where the fluid is described in a compact and efficient manner [13], [14]. Reduced order modeling describes the spatio-temporal evolution of flow as a parameterized dynamic system, although it may also involve the development of a statistical grid of parameters at mean quantities, such as drag. Significant efforts have been made to identify coordinated transformations and reductions that simplify the dynamics and capture the physics of essential flows [15]–[17].

Model reduction, such as the Galerkin projection of the Navier - Stokes equations on an orthogonal basis of the POD (Proper orthogonal decomposition) modes, benefits from a close link with the governing equations; however, it is intrusive, requiring human expertise to develop models from a work simulation.

ML is a rapidly growing body of modular algorithms that can be used for the identification and modeling of data-driven systems. Unique aspects of data-driven modeling of fluid flows include the availability of partial prior knowledge of the equations, constraints, and symmetries that govern [18]–[20].

With the advancement of simulation capabilities and experimental techniques, fluid dynamics is becoming a data rich field, thus becoming accessible to ML algorithms [21]–[23].

Based on the universal approximation theorem, which states that a sufficiently large neural network can represent an arbitrarily complex input-output function, deep neural networks are increasingly used to obtain more nonlinear coordinates. efficient for complex flows. However, deep learning often involves the availability of large volumes of training data that far exceeds network parameters [24]–[26] The resulting models are generally good for interpolation, but may not be suitable for extrapolation when the new input data has different probability distributions than the training data. In many modern ML applications, such as image classification, the training data is so large that it is natural to expect that most future classification tasks will come down to interpolation of the data [27]–[29].

Data partitioning and classification are the other pillars of ML. There are dozens of mature algorithms to choose from, depending on the size of the data and the number of categories we want [15], [30], [31].

The k-means algorithm (as an example) has been used successfully by to develop a data-driven discretization of a high-dimensional phase space for the fluid mixing layer. This low-dimensional representation, in terms of a small number of clusters, allowed workable Markov transition models of how the flow evolves over time from one state to another. Because cluster centroids exist in the data space, it is possible to associate each cluster centroid with a physical flow field, allowing additional interpretation [32]–[34]. Further work has used k-means to partition the phase space into distinct regions, in which local reduced order bases have been constructed, which has improved stability and robustness to parameter variations [35]–[37].

Classification is also widely used in fluid dynamics to distinguish between various canonical behaviors and dynamic regimes. some work has used the k-nearest neighbor algorithm to detect exotic wakes. Likewise, neural networks have been combined with models of dynamic systems to detect flow disturbances and estimate their parameters [38]–[40]. Graphical and related network approaches in fluids have been used for community detection in wake flows [41]–[45].

Much of ML that could be used in fluid mechanical model calibration approaches focuses on the science of imaging (or Computer Vision,), providing robust approaches to improve resolution and remove noise based on statistical inference. These super-resolution and denoising algorithms have the potential to improve the quality of fluid simulations and experiments [46]–[49]. Super-resolution involves inferring a high-resolution image from low-resolution measurements, taking advantage of the statistical structure of the high-resolution training data [50]–[52].

Several approaches have been developed for super-resolution, for example, based on a library of examples, sparse representation in a library, and more recently CNNs (Convolutional neural network) [53]–[57]. Experimental PIV flow field measurements provide a compelling application where there is a tension between the local resolution of the flow and the size of the imaging domain. Super-resolution could exploit expensive, high-resolution data over smaller domains to improve resolution over a larger imaging domain [58]–[61].

Other work has developed a super-resolution algorithm based on CNNs (convolutional neural networks) and demonstrated its efficiency on the reconstruction of turbulent flows, showing that the energy spectrum is preserved with precision. A disadvantage of super-resolution is that it is often extremely computationally expensive, making it useful for applications where high-resolution imaging can be prohibitively expensive; however, improved approaches based on neural networks can significantly reduce costs [62]–[65].

Note also that some works have recently used generative adversarial networks (GANs) for super-resolution. The processing of experimental PIV and particle tracking was also one of the first applications of Machine Learning [49], [66], [67]. Neural networks have been used for fast PIV and PTV, with impressive demonstrations for tracking Lagrangian particles in three dimensions. CNNs have also been used to construct velocity fields from pairs of PIV images [68]–[72]. Related approaches have also been used to detect spurious vectors in PIV data to remove outliers and fill in corrupted pixels [73]–[75].

Machine learning algorithms are well suited to stream optimization and to control problems involving ''black box'' or multimodal cost functions. These algorithms are iterative and often require several evaluations of cost functions of several orders of magnitude more than algorithms based on gradients [76]–[78]. In addition, they do not offer guarantees of convergence. techniques such as Reinforcement

Learning (RL) have been shown to outperform even optimal flow control strategies. Indeed, there are several classes of flow control and optimization problems where learning algorithms can be the methods to choose and to be applied [79]–[81]. Unlike flow modeling, optimization and control learning algorithms interact with the data sampling process in several ways [82]–[85].

ML can be applied to develop explicit substitution models that relate the cost function and control/optimization parameters [86], [86]–[88]. Substitution models such as neural networks can then lend themselves to methods based on gradients, even if they often remain stuck in local minima [89]–[91].

Multi-fidelity algorithms can also be used to combine surrogates with the cost function of the complete problem. As learning progresses, new data is requested based on the results of the optimization. Alternatively, the optimization or control problem can be described in terms of learning the probability distributions of parameters that minimize the cost function [92], [92]. These probability distributions are constructed from cost function samples obtained during the optimization process. In addition, the high dimensional and non-convex optimization procedures that are currently employed to train nonlinear ML models are well suited to the high dimensional nonlinear optimization problems in flow control.

Note that the lines between optimization and control are blurred by the availability of powerful computers and infrastructure(Big data and Cloud Computing, etc.). However, the range of critical spatiotemporal scales and the nonlinearity of the underlying processes will likely make real-time flow control optimization a challenge for decades to come [87], [93], [94].

GA (Genetic algorithms) have also been deployed to solve several flow control problems. They require that the structure of the control law is predefined and contains only a few adjustable parameters. An example of using GAs for control design in fluids was used for the experimental optimization of the backward stage mixing. As with the control of neural networks, the learning time increases with the number of parameters, which makes it difficult or even prohibitive for controllers with non-linearities (for example, a constant-linear-quadratic law), l 'signal history (e.g., Kalman filter) or multiple sensors and actuators [95], [96].

Genetic programming has been used extensively in active control for engineering applications and in recent years in several flow control factories. This includes learning multi-frequency open-loop actuation, multi-input sensor feedback, and distributed control. Note that most of the control laws were obtained in 1000 test evaluations, each requiring only a few seconds in the wind tunnel [97], [98].

In recent years, reinforcement learning (another branch of artificial intelligence and machine learning) has moved beyond the domain of games, robotics, etc., and has become a fundamental mode of problem solving in an increasing number of fields from all over the world. calibration of existing models, in particular to reproduce the dynamics of hydrological systems, actively control the oscillatory laminar flow around bluff bodies, study the individual or collective movement of fish, maximize the range of simulated and robotic gliders, optimize kinematic movement drones and optimizes the movement of micro-swimmers [99], [100].

Knowledge of fluid mechanics is essential for RL applications, as success or failure depends on the appropriate selection of the politic states, actions, and rewards that reflect the governance mechanisms of the flow problem. Natural organisms and their sensors, such as the visual system in a bird or the lateral line in a fish, can guide the choice of states. As sensor technologies advance at a rapid pace, the algorithmic challenge may be that of optimal sensor placement. The actions reflect the flow actuator and may involve body strain or wing flapping. Rewards can include energetic factors, such as cost of transportation or being close to the center of a school of fish to avoid predation. The computational cost of RL algorithms remains challenging for its widespread adoption by the major fluid dynamics domain, but we believe that this deficiency can be compensated by the parallelism inherent in RL or even in the development of sophisticated big data infrastructures. There is growing interest in methods designed to be transferable from low precision (e.g., two-dimensional) simulations to high-precision (e.g., three-dimensional or more) simulations or from simulations to related real-world applications [101].

The applications of ML in modeling, optimization and flow control problems in experiments and simulations are increasingly attracting interest in the sciatica community. Certain works have highlighted certain successes of ML in critical tasks of fluid mechanics, such as dimensionality reduction, feature extraction, feature engineering, PIV processing, super-resolution, reduced order modeling, closure turbulence, shape optimization, flow control and transfer learning (TL) [102].

Machine learning models includes data-driven optimization, applied machine learning algorithms (regression techniques, classification, etc.) that are well suited to high dimensional nonlinear problems (multi-parameter problems), such as those encountered in fluid dynamics (model calibration, etc.); Fluid mechanics expertise will be needed to formulate these optimization, regression problems and supervised models in general. ML algorithms present a variety of tools, very little explored in fluid mechanics research, which can augment existing modes of research and development. Knowledge of fluid mechanics field and centuries-old conservation laws remain relevant in today's era of big data and computation. This knowledge can help formulate more precise questions and reduce the high computational cost often associated with the application of ML algorithms in flow control and optimization (e.g., Deep learning models, etc.). Exploration, exploitation and visualization of large search spaces could be simplified by Machine learning, and increasingly efficient high-performance computing resources further help to encourage its application [103].

It is clear that machine learning approaches are very promising in fluid mechanics to address several subjects. Nevertheless, we note that, despite the maturity of the field of machine learning, its use for the calibration of fluid mechanics models remains very limited especially for overhead line conductors' models. In addition, existing works only incorporates video and image type data and makes very little use of time series that are ubiquitous in most laboratory simulation experiments. Finally, to our knowledge, there have not yet been any approaches exploring the calibration of overhead line conductor models using non baseline ML models.

In this paper, we explore an approach based on the use of an ensemble learning approach (ENS) to calibrate the WO model so that it can be used to approximate the ST model. Our approach is subdivided into three stages:

The first step is to collect data that will be used to build our ENS model. The data collection requires the use of the WO model using the correct values of the hyper parameters in order to have data representative of the problem, that is, close to the ST model. The second step consists of modeling: the data thus generated will be used to build our ENS model. The third step is to evaluate the model built using data that did not participate in its design.

The rest of this paper is organized as follows: section 3 presents the methodology followed to calibrate this prediction model. Section 4 presents the results of this study. Section 5 presents a discussion of these results and finally we end with a conclusion and perspectives as well as some references.

III. METHODOLOGY

This section is divided into two sections. The first section tackles to the strip theory model and the wake oscillator model regarding the basis of the approximation of the Strip Theory model by the Wake Oscillator model. It is mainly a synthesis of the work previously carried out by the RTE and EuroBios R&D teams, of which we offer a summary for the good understanding of the second part which mainly deals with the use of machine learning for the calibration of the Wake Oscillator.

The second part returns to the incorporation of machine learning and the implementation of our ENS model, for the calibration of the Wake Oscillator model.

The WO model is by definition of the problem of this paper the model to be calibrated. Indeed, as specified previously, the model of WO has the characteristics necessary to approximate without loss of much information of the model of ST. In addition, the previous work of our colleagues has shown that this model retains the basic characteristics of the ST model with simplifications of a few terms of this model. It is for this reason that this model is a perfect choice for this approximation [104], [105]. Although there are several approaches and works that use the ST-type model, we mainly used the approach developed in this study referenced here [106].

The next section discusses the foundations and reasoning behind this WO approximation of ST and some studies around the adjustment of WO hyper parameters. This part is very important for understanding the rest which is the heart of this paper: the machine learning approach and where the data comes from.

A. STRIP THEORY AND WAKE OSCILLATOR MODELS: CONTEXT

This section returns to the application of strip theory for overhead line operators. Some results and foundations in the use of this modeling of fluid and structures interaction are established there. However, given the numerical implementation cost associated with such simulations, a simplification of this interaction is proposed via a wake oscillator model. This part reviews the work required to use the strip theory database to calibrate the wake oscillator and its theoretical basis.

A database of Fluid Structure Interaction (FSI) results is necessary to calibrate and validate simplified modelling. A FSI architecture has been developed previously [3], which includes a vibrating string model to represent the cable coupled to a series of Computational Fluid Dynamics (CFD) 'strips'. The simplified modelling consists of the same vibrating string model, but coupled to a Van Der Pol oscillator model with adjustable coefficients. Finding the correct values for these coefficients is the objective of this part.

In the next section, we will discuss some basic explanatory elements of the FSI approach.

1) FSI APPROACH

a: VIBRATING STRING CABLE MODEL

The cable is modelled as a vibrating string

$$m\frac{\partial^2 Y}{\partial T^2} - H\frac{\partial^2 Y}{\partial Z^2} = F_L(Z, T)$$

(1) where Y and F_L are, respectively, the cable displacement and lift force per unit length, both at span location Z and time T, m is the cable mass per unit length, H is the cable tension. In the FSI case, F_L is found from CFD simulations placed along the cable length.

b: NUMBER OF CFD STRIPS

A cable model that time-advances using the second-order implicit Crank-Nicolson method has been implemented. It is possible to have a greater number of cable elements than CFD strips, allowing better resolution of the cable while keeping the expensive fluid flow simulations to a reasonable number.

There are five CFD strips in Figure 1, where the lift force is taken as constant along each CFD strip of the cable. The number of cable elements (202) is sufficient to allow a smooth representation. In Figure 1 the lift force is approximately constant over the central three CFD strips, meaning that assuming constant force applied across each half wavelength could provide a good approximation. It is possible that the force on the outer two strips is less influential on the overall cable dynamic, and some qualitative or quantitative reasoning could be put forward to justify this.

A simple test is needed to find the optimum number of CFD strips for a single cable half-wavelength.

Figure 2 shows results from 5 simulations with different numbers of CFD strips for the same conditions [1]. The growth rate in each case seems to be roughly the same. CFD1 and CFD2 develop vortex shedding oscillations about half a second earlier, which means that their vibration amplitudes are about 10% larger at a given time. In the current context, simulations that demonstrate good qualitative behaviour are the priority. Hence, a single CFD strip could be used for each half wavelength, which will substantially reduce the simulation cost and will be important for realistic conductor cable spans where 20+ wavelengths may be needed.

Simulations for one and three CFD strips could be used to confirm that the same result is obtained for the stabilised state, which will probably arise after ≈ 100 s.

c: INITIALISATION OF HIGHER MODES

The initial motion of the cable is caused by the development of the vortex shedding, which will be triggered by perturbations in the fluid or any initial motion of the cable. If a single CFD strip is used for each half-wavelength then the vortex shedding at each consecutive half-wavelength should eventually have the opposite phase angle. The CFD strips should be initialised so as to have opposing phases from the start of the simulation, in order to avoid long-lived transients. The vortex shedding is tripped by a small vertical velocity $\pm 10^{-4}$ m s^{-1}, and alternating this between positive and negative creates an opposite phase for consecutive half-wavelengths. Figure 3 shows a simulation initialised in this way and demonstrates how two inverse fluid flow simulations are created.

2) VAN DER POL OSCILLATOR MODEL

a: FORMULATION

The cable is modelled using (1). For a fluid with density ρ and velocity U over a cylinder with diameter D and from the definition of the lift coefficient C_L, $F_L = \frac{1}{2}\rho U^2 D C_L$, which is related to the flow variable q via $q = 2C_L/C_{L_0}$, where C_{L_0} is the reference lift coefficient found from experiment or simulation. The cable displacement can be non-dimensionalised as $y = Y/D$, giving

$$\ddot{y} - \frac{H}{mL^2\omega^2} y'' = Mq$$

with a Van der Pol oscillator q to represent the vortex shedding

$$\ddot{q} - \left(q^2 - 1\right)\varepsilon \dot{q} + q = \frac{A}{\omega^2}\ddot{y}$$

where H is the cable tension, m is the cable mass per unit length, $\omega = 2\pi U\mathrm{St}/D$, $M = \rho D^2 C_{L0}/16\pi^2 \mathrm{St}^2 m$ and the coefficients ε and A are to be determined by comparison with the FSI results. The Crank-Nicolson method is used to integrate equations (2) and (3) in time. The initial cable displacement y_0 is zero, while the modes are initialised in a similar way to the FSI with $q_0 = 0.05\sin(n\pi s)$, where n is the expected mode number.

3) FIRST RESULTS

a: LAMINAR FLOW

Simulations without a turbulence model and an artificially large value of viscosity $\left(\mu = 1.81 \times 10^{-4}\ \mathrm{Pa\ s}\right)$ were used to run the preliminary range simulations shown in Figures 4, 5 and 6. Results from oscillator calculations are also shown in the Figures 7 and 8 for a range of A and ε values. Two peaks are observed, one for the mode 1 vibration (a single half-wavelength) and the mode 2 peak, which has a full wavelength. The mode 1 simulation was run with a single CFD strip while the mode 2 was run with two. The mode 2 peak is about twice as high and wide as the mode 1 peak. Both peaks demonstrate a sharp falloff at the lower velocities and a curved falloff at higher velocities, which is inline with observations in the literature [1]. The mode 2 peak is at a velocity that is slightly less than twice the velocity of the mode 1 peak, which indicates a change in Strouhal number between the two peaks. This is most likely caused by the increase in Reynolds number from Re = 209 for the mode 1 peak and Re = 282 for the mode 2 peak. In this range of Reynolds numbers the Strouhal number increases slightly [2], as observed in the change of cable oscillation frequency. Results at higher Reynolds number with a turbulence model are therefore necessary.

Also shown in Figure 4 are results from a range of [L. Zhang's code] oscillator calculations after 4o s. It can be seen how the Strouhal number used is not correct for the mode 2 peak. While the width of the peak is well represented by $A \approx 5$ and $\varepsilon \approx 0.1$, the amplitudes of the oscillator results are roughly half that found using the FSI simulation. Further optimisation or adjustment of the oscillator equation right-hand side terms may yield a closer match.

b: TURBULENT FLOW

At Reynolds numbers that will typically be encountered for conductor cables in weak winds, i.e. Re ≈ 2000 to 6000, the flow is expected to contain some turbulence. It is therefore important to include a turbulence model, both to allow the simulations to work and also to obtain realistic results. It is known that standard RANS turbulence models do not perform well for this type of flow. After studying various approaches [3] found that the $k - \omega$ -SST model produced the best results for this application, so this model will be used here.

In Figures 5 and 6 it is seen that the lift and drag coefficients are greater in the laminar flow case, which corresponds well with the known behaviour [2]. The result is larger amplitude oscillations in the laminar flow case. Comparing Figure 4 and Figure 5, it can be seen that increasing amplitude vibrations in time coincide with a reduction in the C_l oscillation amplitude,

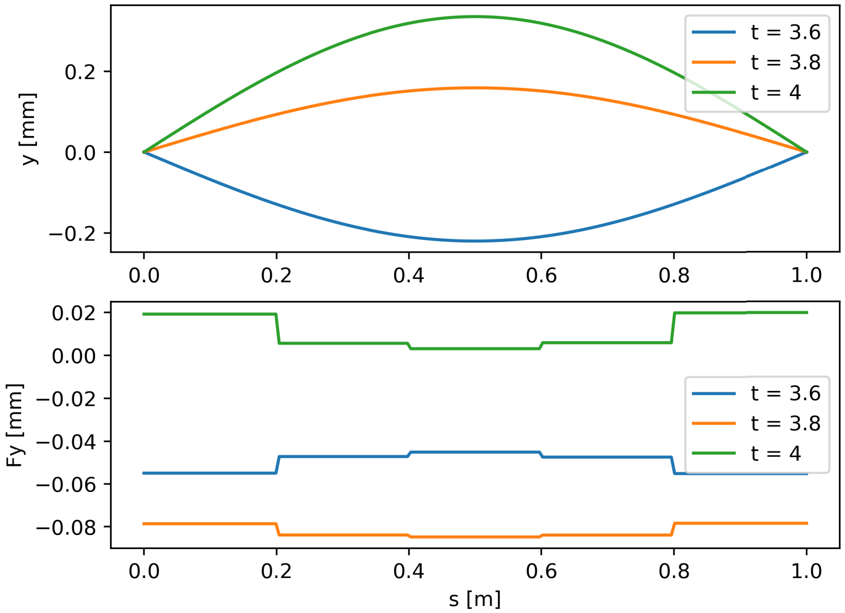


FIGURE 1. Cable with five CFD strips over half wavelength vibration mode with 202 cable elements for various time-steps.

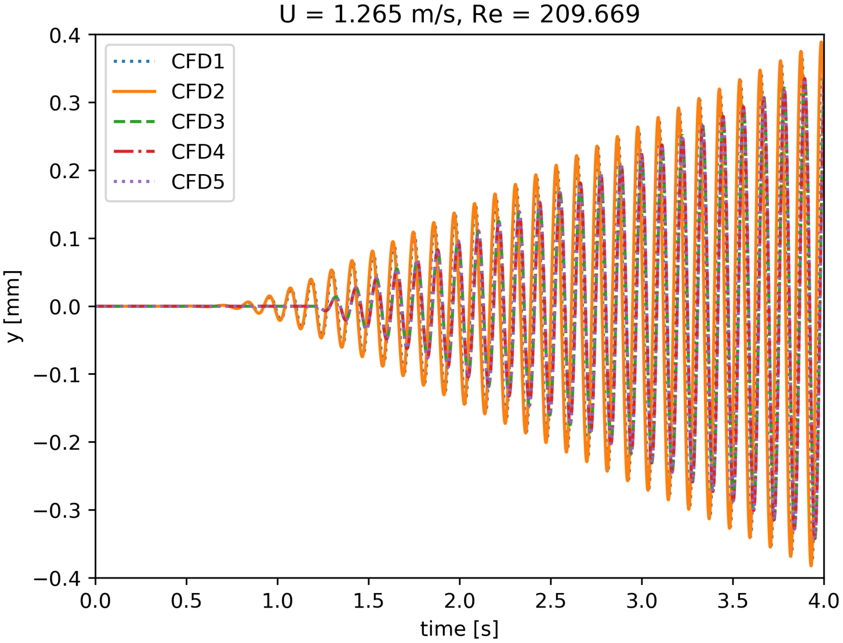


FIGURE 2. Displacement at centre location for simulations with varying number of CFD strips.

which is an effect that is not observed when the cable vibrations remain small over the duration of the simulation. In a similar approach to the laminar case, Figure 9 shows vibration amplitudes for velocities that correspond to the mode 1 and 2 peaks. The mode $1H = 500\,\mathrm{N}$ simulations show a large variation in Strouhal number, but this result can be discounted because it may be due to the low Reynolds numbers. Otherwise, the Strouhal number at both peaks is ≈ 0.24, while the Strouhal number away from the peaks is slightly lower at 0.23, which better corresponds with the expected value of 0.221[3, Table 1], with a total variation of about 5%. The mode 1 peak for $H = 2000\,\mathrm{N}$ has a sharp falloff at the lower velocities and a curved falloff at higher velocities, which is inline with observations in the literature [1]. The mode 2 peaks seem to have a similar form, but more points are necessary.

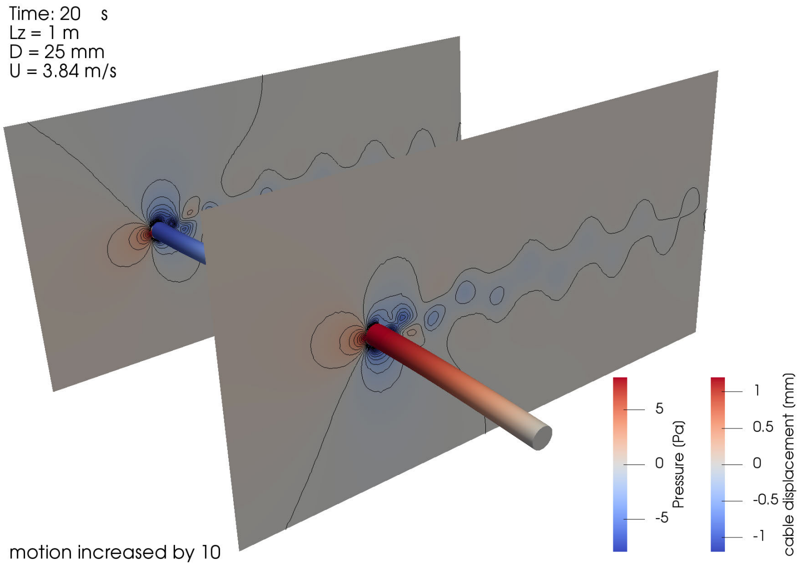


FIGURE 3. FSI simulation with two CFD strips (mode 2); initial $u_y = -10^{-4}$ on left and $+10^{-4}$ on right. Simulation $u_x = 3.84$ m s^{-1} and Re = 6400, with $H = 2000$ N. Contour spacing 1 Pa.

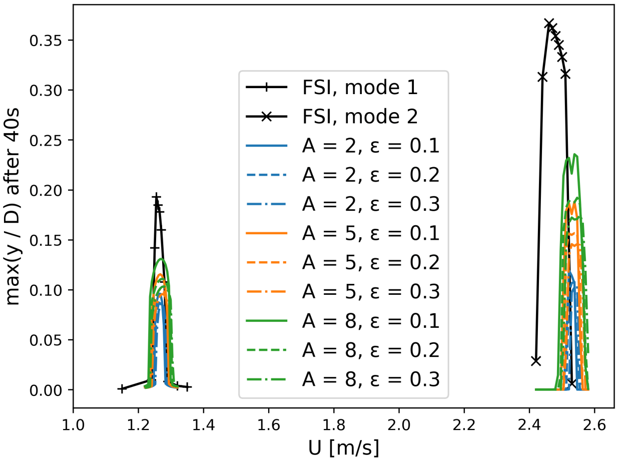


FIGURE 4. Vibration amplitudes after 4os for a 25 mm diameter and 1 m long cable ($H = 500$ N, $m = 1.57$ kg/m, cable has 82 elements for mode 1 and 162 elements for mode 2).

Values of lift coefficient oscillation amplitude and drag coefficient are $C_{L_0} \approx 0.55$ and $C_D = 0.74$ at the end of the simulation are shown in Figure 10. While C_{L_0} is larger at ≈ 0.57 early on (once the vortex shedding is established) these values are lower than expected $C_{L,rms} = 0.759$ which may be due to the low Reynolds number (Re $= 3150$). The low lift oscillation amplitude is most likely the reason why the cable vibrations are smaller than the laminar case. Higher Reynolds number $\left(O\left(10^4\right)\right)$ cases could be generated by altering the cable tension H and length L while maintaining the focus on modes 1 and 2 vibrations as demonstrated in Figure 11 and 12.

4) COMPARISON WITH VAN DER POL OSCILLATOR

Figures 13 and 14 shows a comparison between FSI results and the Van der Pol Oscillator model. Comparing Figure 13 and Figure 14 demonstrates a large difference between values used for the modes 1 and 2, along with a strong change in shape (rounded for mode 1 and seemingly sharp peaks for mode 2; more points needed). For mode 2 a factor 10 change in Figure 13 does not yield a significant change in vibration amplitude. Altering ε entails a significant shift, while not changing the width of the peak nor seeming to induce a significant change in amplitude. In the mode 2 case, increasing ε to 0.3 causes a better match with the $H = 2000N$ FSI result.

The sharp peak in the mode 2 oscillator result does not correspond well to the FSI result. The equivalent of Figure 6 and Figure 8 for the Van der Pol do not demonstrate the same interaction that causes a reduction in C_l or q amplitudes with time. Unlike in Figure 6, the increase in vibration amplitude for the Van der Pol oscillator is observed to be linear, making larger amplification possible…There is also no damping in the Van der Pol simulations.

B. FURTHER TESTING

1) STEADY STATE

The Van der Pol simulation shown in Figure 15 has been run up to 400 s. A fully stabilised state has not yet been obtained. It is possible that FSI simulations will demonstrate a greater tendency to find a stabilised state due to the reduction in lift coefficient oscillation amplitudes; this should be verified with a simulation. Finally, a last point to note concerning the time steps and the cable tension effects that: 1) The effect of

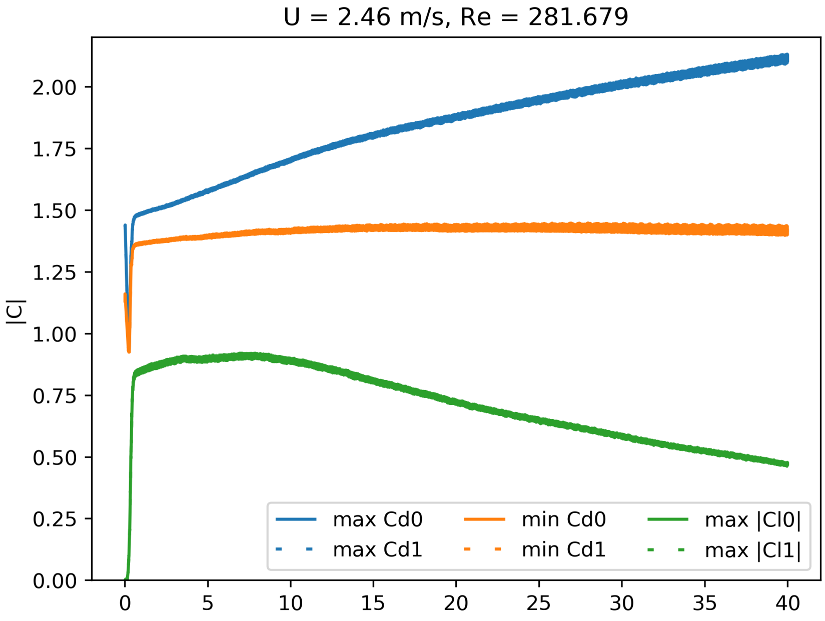


FIGURE 5. Lift and drag coefficients for mode 2 vibrations. (Laminar).

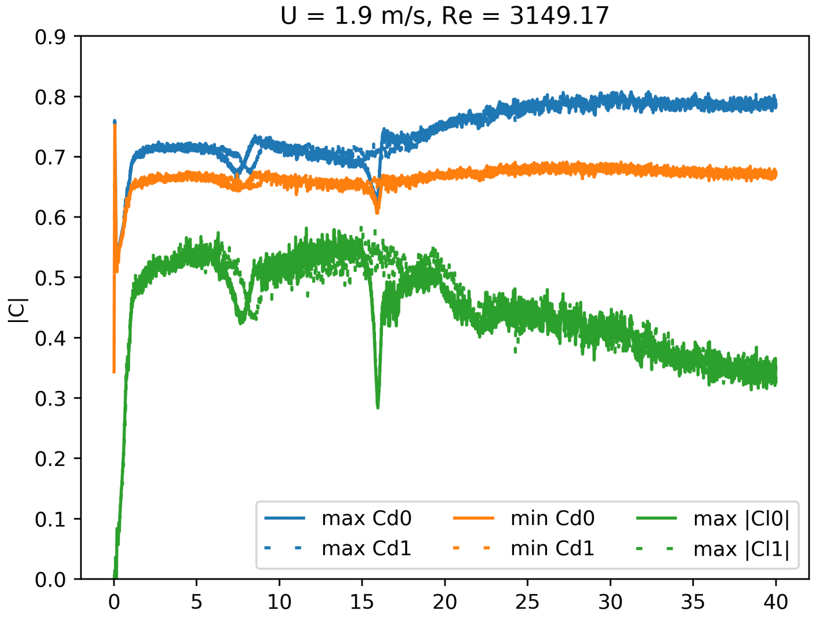


FIGURE 6. Lift and drag coefficients for mode.

time-step length should be investigated. Initial checks have not shown the same amplification for two different choices of Δt and 2) Shortened cables with lowered the tension values have been used to make simplified investigations. If cable stiffness is to be included, representative values of cable axial stiffness EA relative to the cable tension H need to be used. Otherwise, the effect of the EA term is too strong and causes unrealistic results to be produced.

C. ENS MODEL FOR THE CALIBRATION OF THE WAKE OSCILLATOR MODEL

In the previous section, we presented the foundations of this approximation between the two models: Strip theory and wake oscillator. This approximation is only possible with manual manipulations and adjustments of the wake oscillator model's hyper-parameters. What causes problem of slowness in these manipulations.

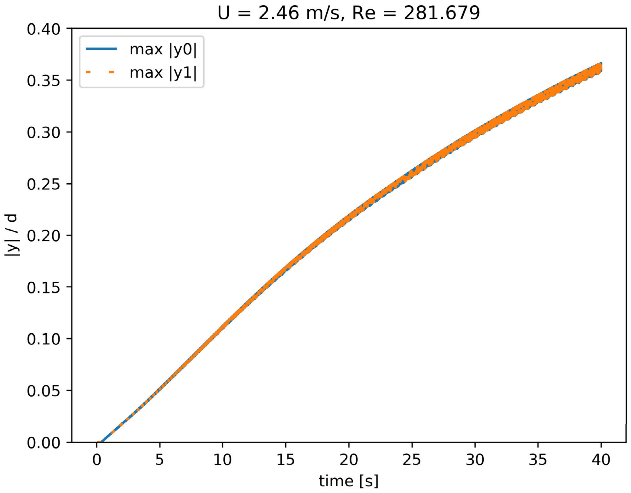


FIGURE 7. Cable oscillation amplitudes for mode.

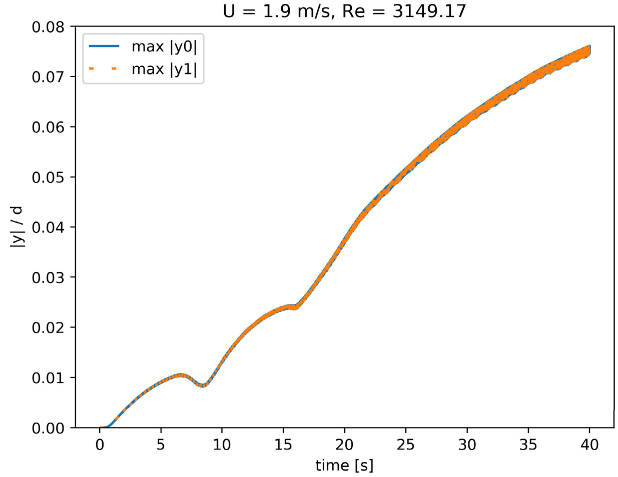


FIGURE 8. Cable oscillation amplitudes for mode 2 vibrations (Turbulent).

We propose a framework based on the use of machine learning to automatically calibrate the wake oscillator hyper parameters' model. In the following section, a methodology of the proposed ensemble model to predict the values of y/d is described.

The fact of proposing an overall model does not necessarily mean that it is a model that was chosen from the start. As a machine learning model, before embarking on the search for a good or accurate model, we should first establish a baseline. It is typically a series of machine learning models that should be tested as a priority to build on a basis for comparison. Then, if one of the models is satisfactory, we take it as the best model, otherwise we will have to build a new model, as is the case in this paper. It is therefore an iterative and progressive approach in which it is first necessary to build a basis of comparison before moving forward. The choice of the baseline is part of what is called "best practices" in the implementation of a model or a machine learning strategy.

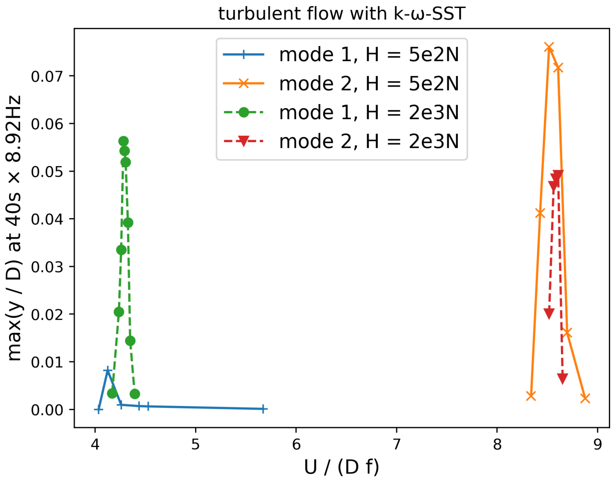


FIGURE 9. Vibration amplitudes before $t_{\max} L\sqrt{H/m}/2 = 357$ for a 25 mm diameter and 1 m long cable with $k-\omega$-SST turbulence model ($H = 500$ N, $m = 1.57$ kg/m, cable has 82 elements for mode 1 and 162 elements for mode 2).

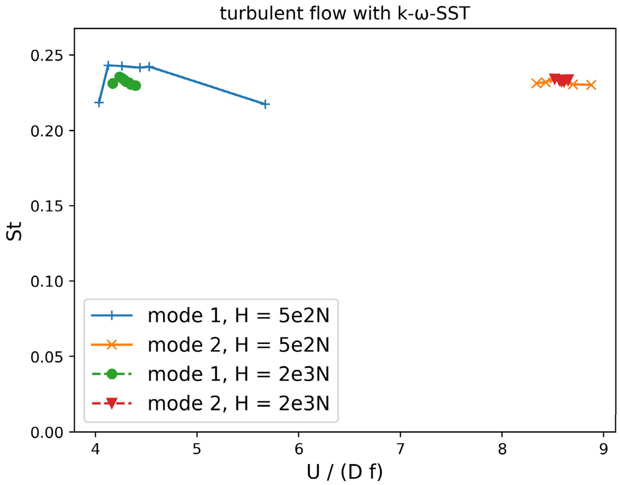


FIGURE 10. Frequencies before $t_{\max} L\sqrt{H/m}/2 = 357$ for a 25 mm diameter and 1 m long cable with $k-\omega$-SST turbulence model ($H = 500$ N, $m = 1.57$ kg/m, cable has 82 elements for mode 1 and 162 elements for mode 2).

The state of the art has clearly specified the models and algorithms to be tested for cleanliness before moving towards a more elaborate model, whether for regression or classification problems [107].

We have chosen the following metrics to measure the performance of our learning models [108], [109]:

- R2 Score: Pearson's linear coefficient of determination.
- RMSE: Root mean square deviation

Fig. 16 and 17 shows the outline of the proposed model. Figure 16 shows the logic followed for the calibration of the Wake Oscillator model. Figure 17 presents the methodology followed to train and build our machine learning model (ENS).

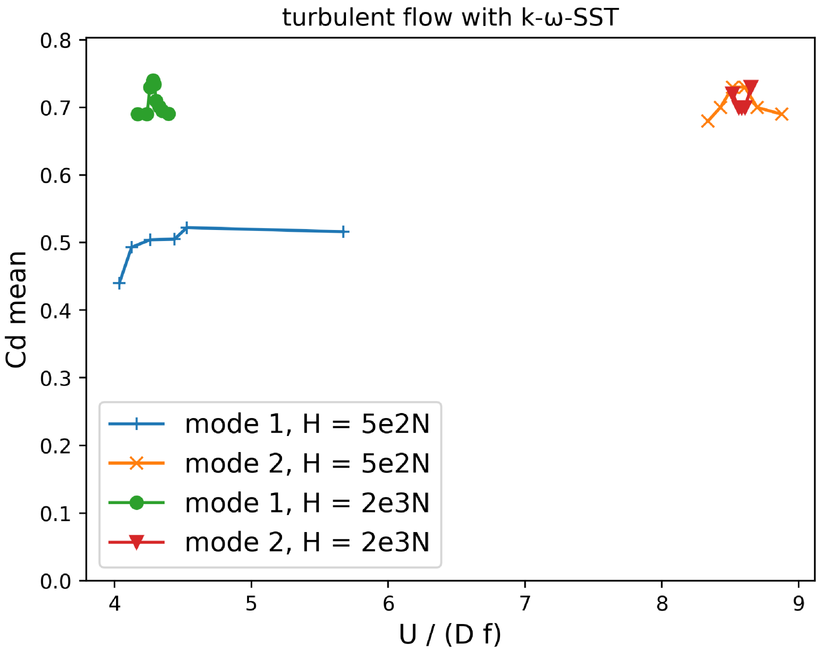


FIGURE 11. Mean drag coefficient for a 25 mm diameter and 1 m long cable with $k - \omega$-SST turbulence model ($H = 500$ N, $m = 1.57$ kg/m, cable has 8_2 elements for mode 1 and 162 elements for mode 2).

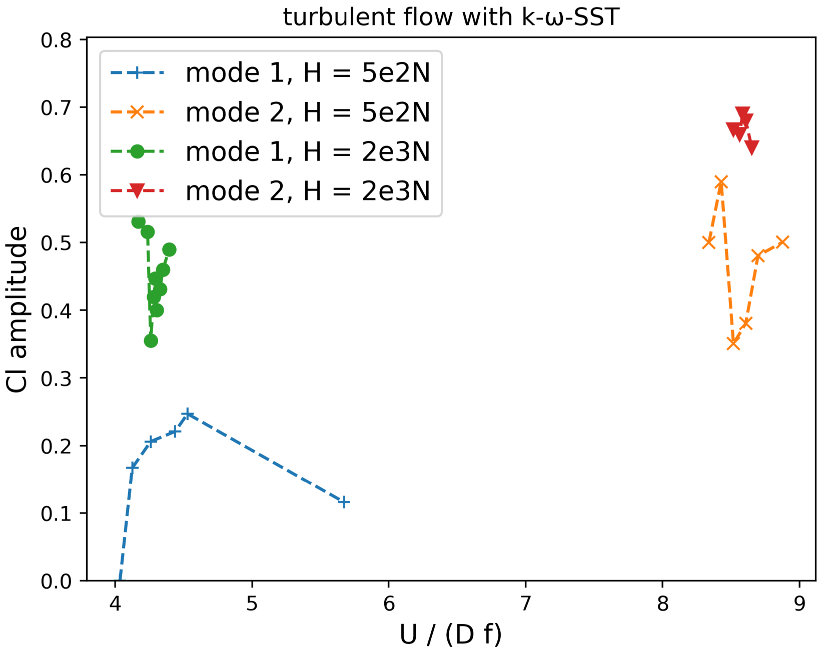


FIGURE 12. Lift coefficient vibration amplitude for a 25 mm diameter and 1 m long cable with $k - \omega$-SST turbulence model ($H = 500$ N, $m = 1.57$ kg/m, cable has 8_2 elements for mode 1 and 162 elements for mode 2).

First, the historical data is pre-processed. Next, the proposed model selects the most relevant input features from our dataset. Finally, an optimal structure of our ENS model was obtained with the combination of two types of models: a series of Xgboost models combined with a DNN model thanks to Ridge regression to obtain a final model called ENS.

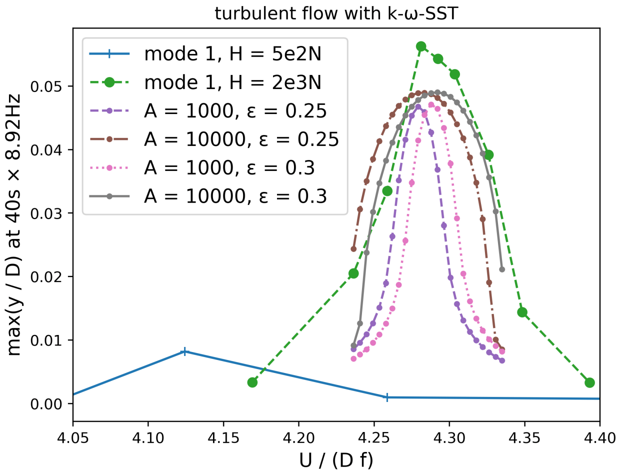


FIGURE 13. Same as figure 9 with Van der Pol Oscillator results.

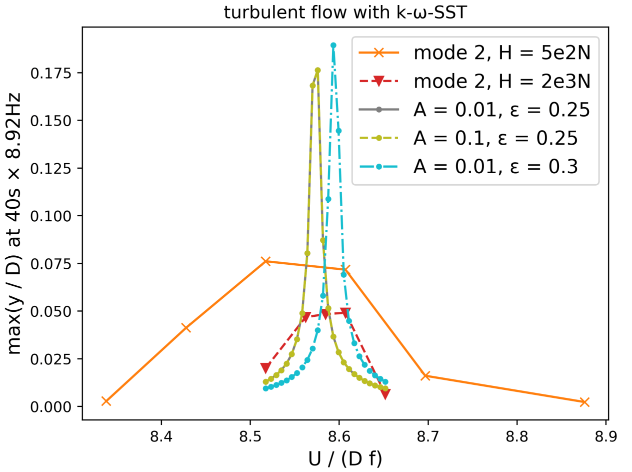


FIGURE 14. Same as figure 9 with Van der Pol Oscillator results.

1) DATA COLLECTION

In this study, our ENS model operates on time-series WO data in order to predict the values of y/d. WO data can be obtained from the executions of the WO model, following the different values of its hyper parameters. Our dataset contains 10 variables, including the hyper parameters values (md, U[m/s], d[m], m[kg/m], L[m], H[N], Nt, Dt[s], tf[s], ymax[m]) and the y/d values. Knowing that the value of y/d is the value to be calculated at each iteration of the model and according to the values of the hyper-parameters of the model.

Figures 19, 20 and 21 give an overview of the values of y/d of the two models (WO and ST) according to the values of the hyper-parameters. Adjusting these values makes it possible to improve this approximation, as we can see in Figures 18 and 19. Fine tuning of the hyper-parameters makes it possible to achieve an optimal configuration as in Figure 20. This is an example of a single time series. Our database consists of 60 time series. This represents a total recording of nearly 5620 seconds.

We need a suitable metric to compare 2 time series. One of the widely used metrics is the RMSE metric. This method requires having 2 time series of the same length, i.e., with the same number of points. In order to obtain time series with the same number, we apply a transformation on the reference model, basing the definition of the timestep (dt) and the output step (dr) on the final time of the WO model and on the number of total points.

$$tf\,(finaltime) = Dt/Dr \tag{1}$$

This formula makes it possible to obtain a time series of the ST model having the same number of points as the model of WO. This subsequently allows the use of methods to compare 2 time series including the RMSE. Now that the data is preprocessed, the next step discusses the modeling.

2) STRUCTURE OF ENS MODEL

the technical architecture of our ENS model is composed of 10 XGboost type models (to form an XGboost drill) and a neural network (DNN). All the XGboost and DNN models take as inputs the data of the WO, namely the values of y/d as well as the values of the hyper-parameters of the WO model which made it possible to obtain these values of y/d. These models individually produce the predictions that serve as metadata to feed the Regression Ridge model. The final output of the ENS model therefore takes the form of a weighted average of all the outputs of all the models (XGboost and DNN) taken individually. For the given input dataset $A = [X_1, X_2, \ldots .X_n]$, where X_i denotes the input vector consists of WO input time series data and hyper-parameters values $(x_i)\,, X_i =< x_1, x_2 \ldots x_n >$, we define a function to predict y/d of ST model at time $t+1$ on the basis of WO hyper parameters values.

It can be given as follows:

$$\hat{z}_{t+1} = f(X)_t = G\,(X_1, X_2, \ldots .X_n)_t$$

where $\hat{z}_{t+1}$ denotes the predicted y/d, $G(.)$ represents the function of proposed ENS model which is a combination of XGBoost and DNN models as given below:

$$G(X) = [G_1(X), G_2(X), \ldots ., G_m(X)]$$

where $G_1(X), G_2(X)\ldots, G_{m-1}(X)$ denotes the $m - 1$ XGBoost models (i.e. in XGBoost forest as explained in the previous section) and $G_m(X)$ denotes the DNN in the ENS model. These base models are trained to predict y/d as follows:

$$\begin{bmatrix} \hat{z}_1 = G_1(X) \\ \hat{z}_2 = G_2(X) \\ . \\ . \\ \hat{z}_m = G_m(X) \end{bmatrix}$$

where $\hat{z}_i$ is the y/d predicted by individual base models (i.e. XGboost and DNN). The outputs of these base models are given as input to the Ridge Regression, which assigns

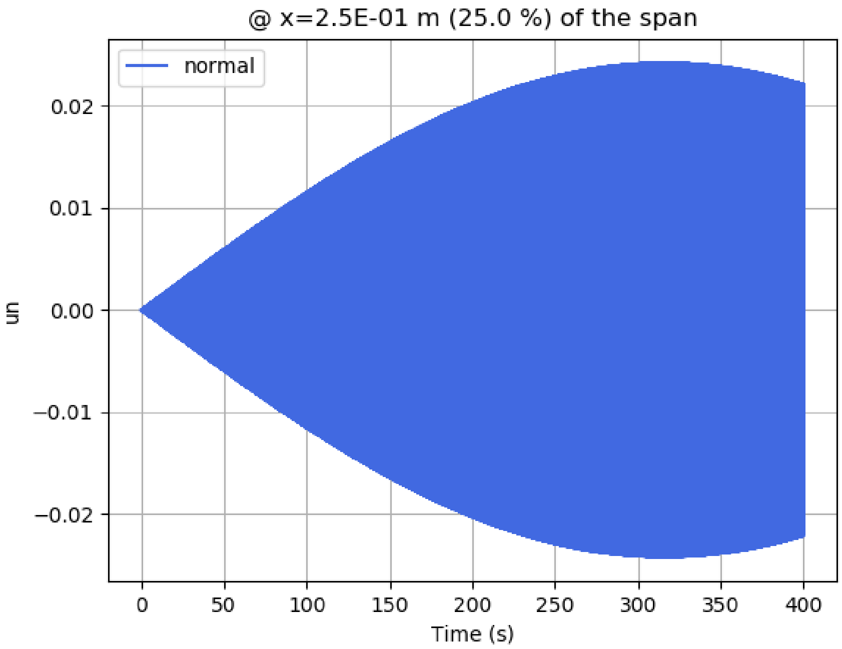


FIGURE 15. Extended mode 2 Van der Pol oscillator simulation.

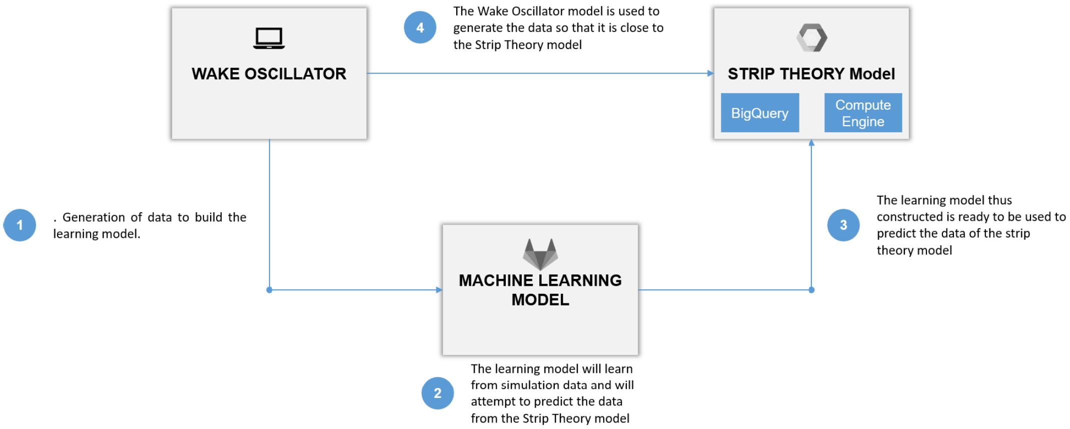

WAKE OSCILLATOR
4
The Wake Oscillator model is used to generate the data so that it is close to the Strip Theory model
STRIP THEORY Model
BigQuery
Compute Engine
1
. Generation of data to build the learning model.
MACHINE LEARNING MODEL
3
The learning model thus constructed is ready to be used to predict the data of the strip theory model
2
The learning model will learn from simulation data and will attempt to predict the data from the Strip Theory model

FIGURE 16. The context of the study: the WO model will generate data that must be calibrated to approach the data of the ST model. To do this automatically, a machine learning model has been developed.

weights to each base model to give the final y/d prediction as explained here:

$$\hat{z} = k_0 + k_1 * \hat{z}_1 + k_2 * \hat{z}_2 + \ldots\ldots + k_m * \hat{z}_m$$

where k_i denotes the weights assigned to each base model (XGboost and DNN). These weights are optimized using the square of the sum of the difference between the actual y/d and predicted y/d, as follows:

$$C(k) = \frac{1}{2n}\sum_{i=1}^{n}(z_i - \hat{z}_i)^2 + \lambda \sum_{j=1}^{m} k_j^2$$

where C(k) is the function which is used to optimize the weights of the Ridge Regression model and z_i is the actual y/d.

Each XGBoost sub-model is trained with 80% of randomly selected training data to ensure that the data and each model is trained on separate hyperparameters. The Regression Ridge model is added in order to integrate each prediction result of the y/d value of each submodel independently.

the literature suggests that to determine the generalization performance of models, our dataset is divided into three subsets: training, validation, and test set, having 80%, 10%, and 10% of total data, respectively. - Training set is used to train and build the sub models XGBoost and DNN. - Validation set is used to determine the performance of different sub-model architectures (number of regression trees in the XGBoost, number of hidden layers and units or neurons for the DNN architecture). - Test set is used to determine the error rate

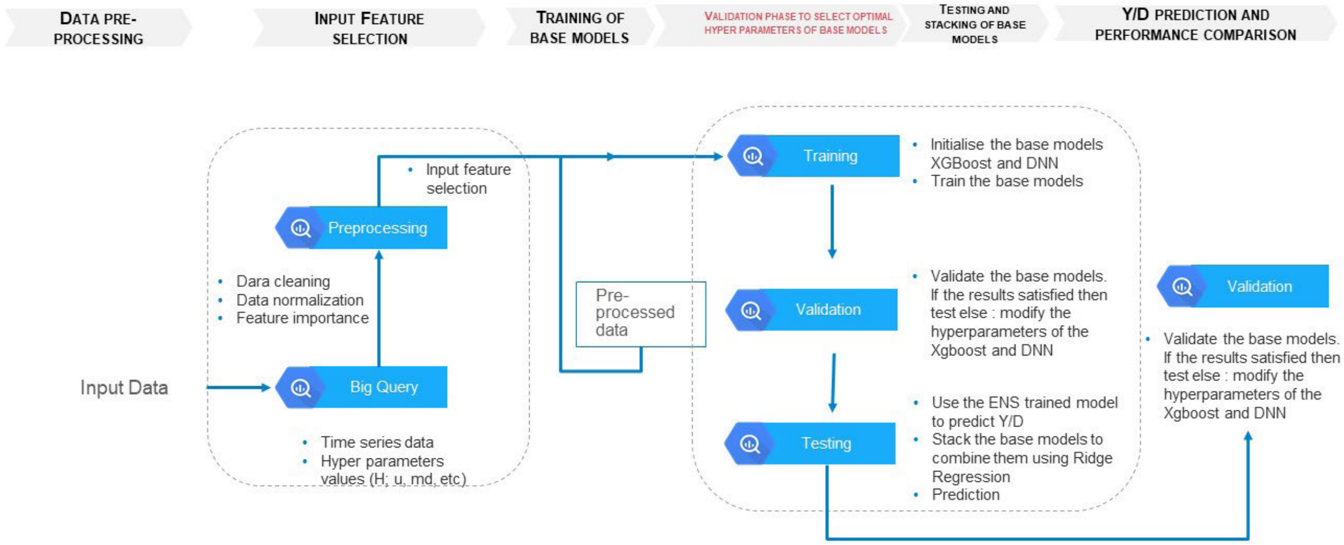

DATA PRE-PROCESSING
INPUT FEATURE SELECTION
TRAINING OF BASE MODELS
VALIDATION PHASE TO SELECT OPTIMAL HYPER PARAMETERS OF BASE MODELS
TESTING AND STACKING OF BASE MODELS
Y/D PREDICTION AND PERFORMANCE COMPARISON
Input Data
Big Query
Time series data
Hyper parameters values (H; u, md, etc)
Dara cleaning
Data normalization
Feature importance
Preprocessing
Input feature selection
Pre-processed data
Training
Initialise the base models XGBoost and DNN
Train the base models
Validation
Validate the base models. If the results satisfied then test else : modify the hyperparameters of the Xgboost and DNN
Testing
Use the ENS trained model to predict Y/D
Stack the base models to combine them using Ridge Regression
Prediction
Validation
Validate the base models. If the results satisfied then test else : modify the hyperparameters of the Xgboost and DNN

FIGURE 17. The framework of the implementation of the ENS model.

of the generalisation by the mean of the root mean square error (RMSE) and mean bias error (MBE).

3) IMPLEMENTATION

In this section, we propose to approach the fine tuning of our model. This process is the only guarantee of the quality of the model, its capacity for generalization and its robustness which, according to the literature, remain important elements to define especially when it comes to creating machine learning models for a field such as fluid mechanics.

a: XGBoost

We start by addressing and explaining the foundation of the best XGboost model structure. A single XGboost model is firstly evaluated on the validation set using a Grid search technique. this grid search technique must find the best combination between two parameters: (a) number of regression trees (XGBoost) ($n\_$ estimators, k) and (b) maximum depth of a regression tree (d). A set of XGBoost models are used to build an XGBoost forest. An optimal number was found by evaluating the model in the validation set.

b: DEEP NEURAL NETWORK

The structure of this model which was built using Tensorflow library is composed of a) input layer taking into account the different values of the hyper-parameters of the WO model, the values of y/d as a function of time, b) several separate layers with several neurons and having as an activation function the "Relu" function and finally c) an output layer with a single neuron representing the value of y/d having a linear activation function. The number of hidden layers was calculated as follows: we start with a single hidden layer with a number of epochs ranging from 50 to 350 in steps of 10. Then gradually we add more hidden layers and RMSE values are calculated in the training, test and validation sets respectively. We come to 8 hidden layers that produced the best result in terms of RMSE. Beyond this step, the model enters an overfitting phase. To this end, a dropout layer is added in the network as regularization method to prevents the over-fitting, improves the generalization performance as well.

All the implementations were carried out in an AWS environment under SageMaker under an instance of type "p4d.24xlarge"

In the next section (results), we also compared our results with the results of other competitive models (Random Forest, Linear Regression, Adaboost, SVR, XGboost, DNN). For the DNN model we tested, is a neural network type model with a single hidden layer, an input layer representing the data and an output layer with a single neuron representing the prediction value y/d.

IV. RESULTS

In this section, we will present the different results of the prediction of the values of y/d according to different machine learning models as well as our ENS model. Figure 24 also details the values of some hyper parameters of the models

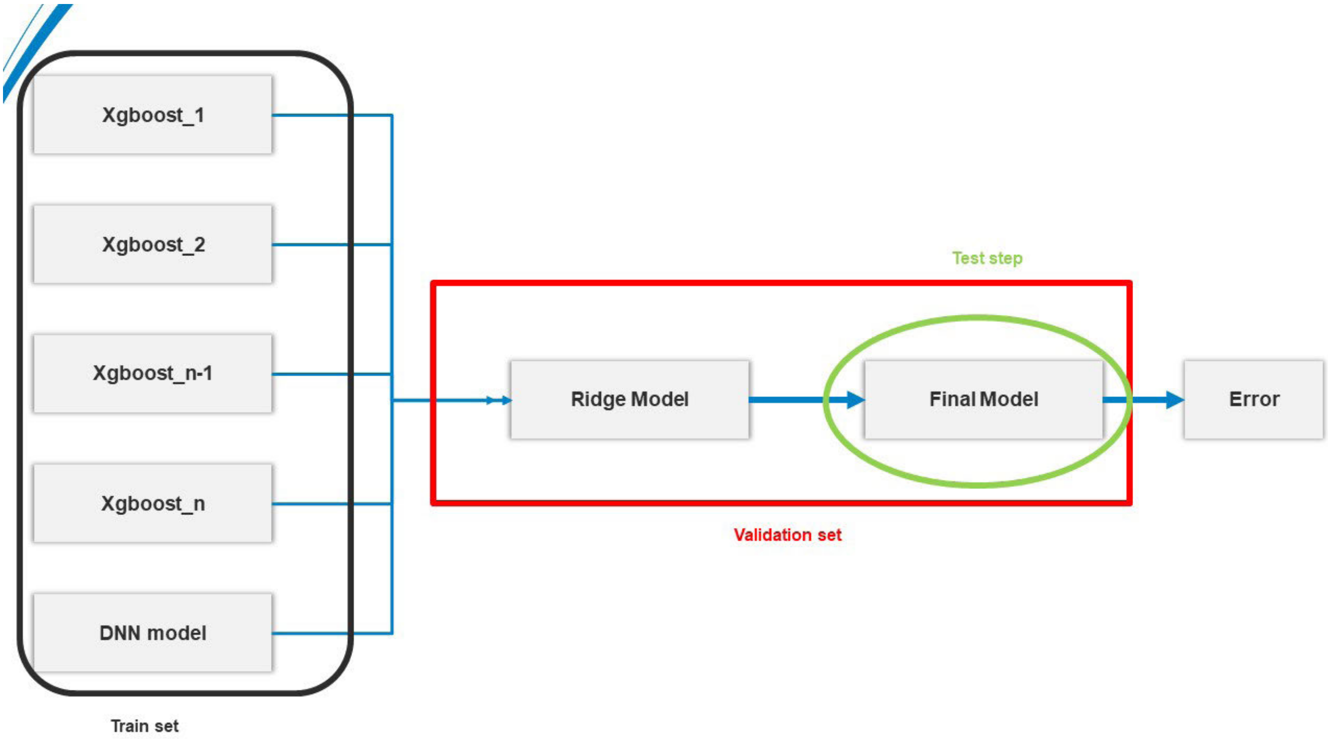


FIGURE 18. **The architecture of the proposed ENS model.**

taken into consideration in this study. This benchmark mainly concerns RMSE type errors and the R2 Score, because these two metrics are the most significant, often widely used in the state of the art, especially in the validation of machine learning models whether in regression or in classification [110].

This section is divided into three sections: i) the visualization of the data collected using the two models: WO and ST and the various adjustments made in order to produce an approximation of the two models, ii) Build our training dataset, the modeling and presentations of the first results with several learning models and finally iii) the prediction results of the value of y/d using our ENS model.

A. DATA VISUALIZATION

The creation of the training set required the use of the model WO in order to match the data of the model ST with the data of the model WO. The following Figures 19, 20 and 21 make it possible to visualize a sample of a single time series of simulation data of the data of the WO model compared to the ST model.

Notice that by changing the values of the hyper-parameters of the model WO, the two curves come closer until a better result is reached (Figure 21).

According to our previous work, we have indicated the most promising parameters in order to converge the WO simulation model to the ST model. Among these parameters the parameter u (Wind Speed), and the modifications made were tested for a range of values going from 0.1 to 7.0. Changing this parameter accentuates the general curvature of the time series, the points are closer together, The maximum amplitude taken by the points is greater. Next, we evaluated the impact of changing the "Tension" parameter on the time series. The range of values of the voltage parameter is 1 to 40,000. The range of values being large, it has been established that the voltage has a large impact and the step to explore the interval should be small enough not to miss interesting time series to exploit.

In the end, an exploration of the ranges of values with a step of 100 gives results which are more easily exploitable and transposable to other time series. Furthermore, a definition of the voltage parameter = 100 causes a significant increase in the maximum amplitude.

Following the study of the impact of the parameters u and h on the simulation time series, we extended the study to the parameters cl0 and eps defined in the configuration of the simulator. causes a linear increase in the maximum amplitude of the time series. Changing the value of eps has no impact on most time series, some time series shows an increase in amplitude for eps = 0.85.

B. MODELING

The training set created in the "Data Visualization" part was used to train a series of training models. For comparison, we compared the Y/D prediction results of our ENS model by a series of models. Table 1 summarizes the results obtained by understanding some Machine learning algorithms in terms

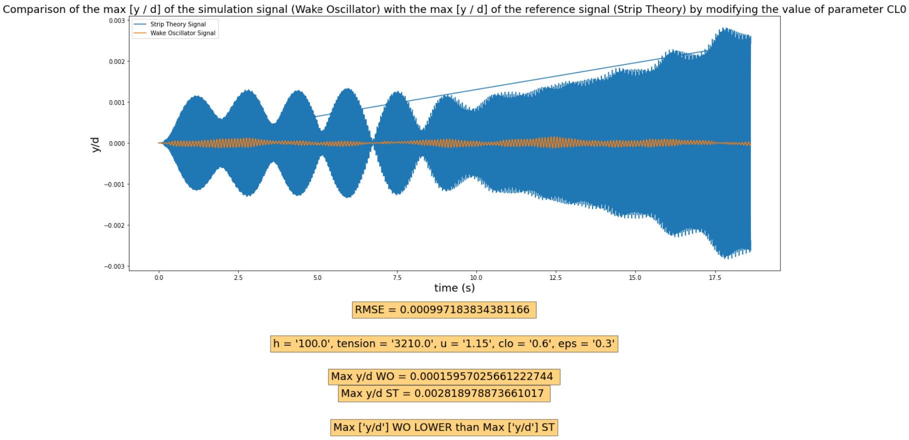


FIGURE 19. Visualization of simulation data WO vs. ST.

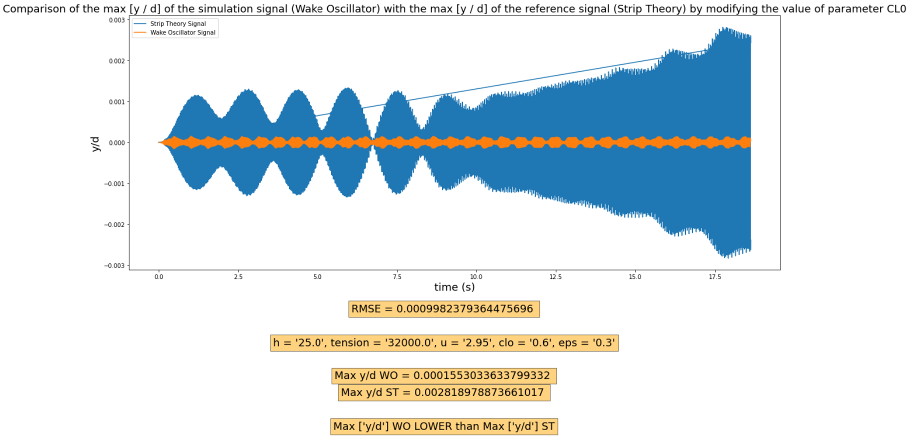


FIGURE 20. Visualization of simulation data WO vs. ST.

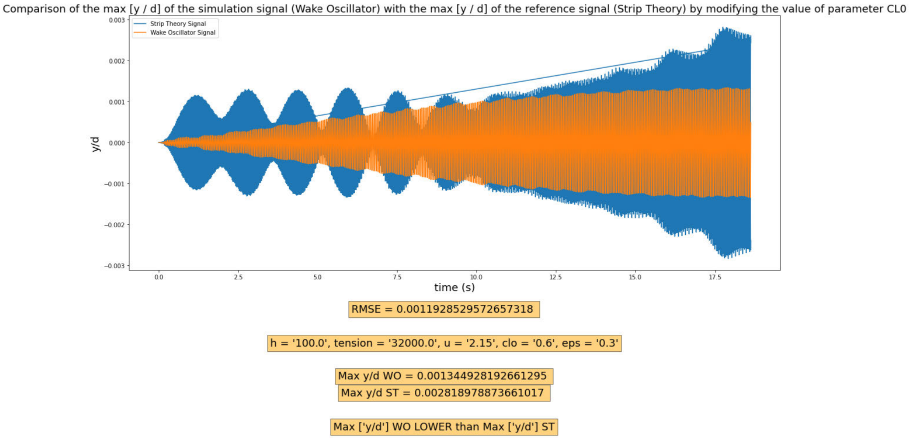


FIGURE 21. Visualization of simulation data WO vs. ST.

of R2 Score and RMSE. The R2 score gives an indication of how close the ST data is to the prediction data. The higher the R2 Score, the better the prediction quality. Since the R2 Score alone does not whistle, the RMSE was calculated (as an example) to show that the best model must also have the weakest RMSE. Figures 22 and 23 show an example of some prediction results of the Linear Regression model compared to our ENS model.

The Linear Regression model (as an example) has a very high RMSE compared to the ENS model. Same remark for

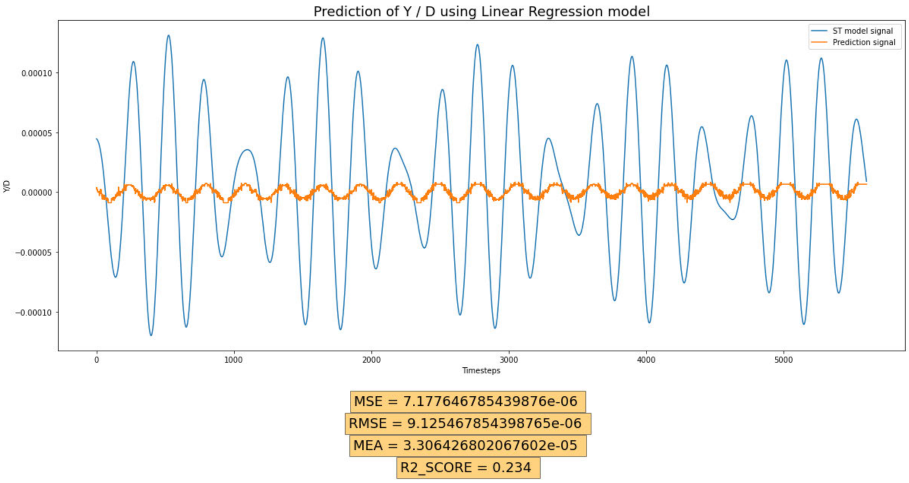


FIGURE 22. Predicting ST model data by training WO data with the Linear Regression model.

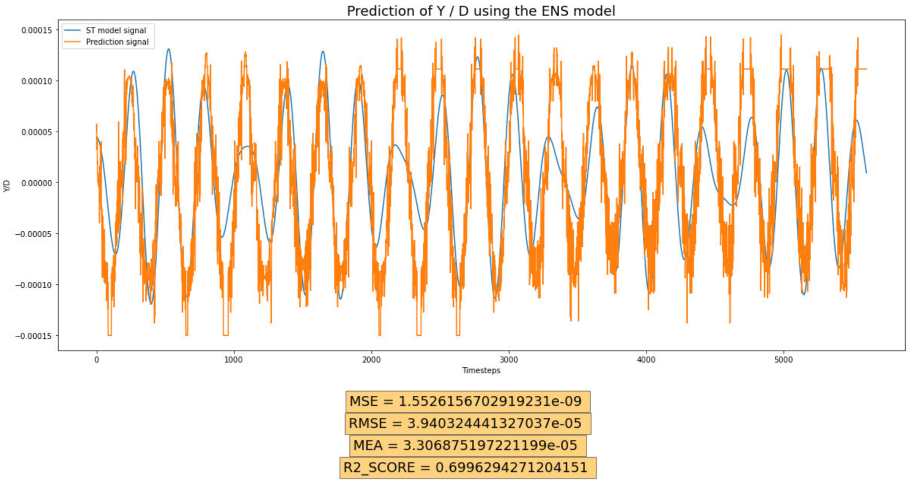


FIGURE 23. Predicting ST model data by training WO data with the ENS model.

TABLE 1. Results of our benchmark: Here, we took only two types of metrics: the RMSE to demonstrate the error made by the different models and the R2 Score to illustrate their predictive capacity (as long as R2 Score close to 1, this means that the model in question has 'better predictive ability).

Algorithm	R2Score	RMSE
Random Forest (RF)	0.322	1.1334678543987654e-05
LR	0.234	9.1254678543987654e-06
AdaBoost	0.299	6.3334678543987654e-06
SVR	0.294	7.8284678543987654e-06
XGBoost	0.390	5.6565898765678943e-06
DNN	0.400	1.7676789874657436e-08
ENS	0.699	7.5676543245673839e-09

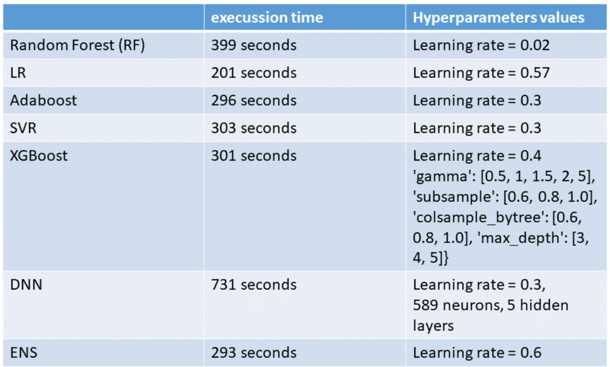
the R2 Score, the ENS model presents an R2 Score which is close to double that of the Linear Regression model. From the results of Figures 22 and 23, as well as the results of benchmarking in Table 1, we can see that the predictive capacity of our ensemble model confirms the results of the state of the art, namely that the set models produce better results than models used individually.

The predictive capacity of the ENS model is mainly due to the characteristics of XGBoost models of producing results on not too high quantities of data as well as the predictive power of DNN type models on time series especially when learning is done on data. almost raw time series.

All the details of the hyperparameter values of the different algorithms used here are summarized in the table in Figure 24.

V. DISCUSSION

The results of prediction of the values of y/d made it possible to show the predictive capacity of our ENS model compared to models well known in the state of the art. Basic models such as Linear Regression showed very low prediction as well as very high RMSE. A small improvement was observed using the Random Forest model but this remains insufficient in terms of prediction error which remains very high. The DNN-type models also showed an improvement in the predictive quality in terms of RMSE and R2 Score but remains below 0.5 in terms of R2 Score. The ENS model has



	execussion time	Hyperparameters values
Random Forest (RF)	399 seconds	Learning rate = 0.02
LR	201 seconds	Learning rate = 0.57
Adaboost	296 seconds	Learning rate = 0.3
SVR	303 seconds	Learning rate = 0.3
XGBoost	301 seconds	Learning rate = 0.4 'gamma': [0.5, 1, 1.5, 2, 5], 'subsample': [0.6, 0.8, 1.0], 'colsample_bytree': [0.6, 0.8, 1.0], 'max_depth': [3, 4, 5]}
DNN	731 seconds	Learning rate = 0.3, 589 neurons, 5 hidden layers
ENS	293 seconds	Learning rate = 0.6

FIGURE 24. Some hyperparameters values of the algorithms used in our study. In this table we have detailed the values of some hyperparameters of the algorithms used in our study. The execution time has been added in order to compensate for the scope of our approach, which despite everything remains low in execution time. For the other hyperparameter values, unless otherwise stated, we kept that the learning rate, the other values, if they are not mentioned means that they remain unchanged as defined in scikit learn.

surpassed all the models tested (table 1) and obtains better results in terms of R2 Score and a better error (RMSE). This explains the power and predictive capacity of the ENS model because coupled with the DNNs and a series of XGBoost, the DNNs participated in better managing the raw data coming from the WO model. In addition, the XGBoost models have brought a predictive capacity which has improved the prediction results. These results are in line with the state of the art in which several studies have shown the predictive power of global worlds. Our model in particular took its power from its structure which is composed of XGBoost models which are known in regression spots but also the power of models with a neural architecture (DNN) added more flexibility in the implementation of the model which positively influenced the quality of the prediction. The results show that the baseline models are less efficient than our model. For example, a linear regression or a decision tree could not obtain a good prediction and both lack capacity and amplitude, despite the tuning applied to all the hyperparameters of these models. was not enough. It could be explained by the lack of data which would give these types of models an interesting predictive capacity. [111], because not all machine learning models learn in the same way and do not require the same amount of data to predict correctly [112]. Second, and this point follows from the first, it is the quality of the training data which, despite the adjustment made by means of the WO model [113], this would probably not have been enough to bring more capacity to this type of model (baseline). And finally, it could be an intelligent use of certain features to be calculated which could increase the quality of the models [114]. But even with these three assertions, the state of the art has shown that on almost similar subjects, the different works very often require to create their own predictive models. [56].

Even if there are no studies to date that use the principle of calibration of overhead line conductor models by machine learning, the results of this paper are comparable to work that can be qualified as belonging to the field of fluid mechanics: whether for images / videos, energy and transport [77], [115]. These work and others employ baseline or near-baseline models with some modifications in either the type of training [116] which remains fairly supervised and obtains results with a very low RMSE or a modification in the heart of an algorithm such as convolutional kernels in CNNs to recognize images or objects in videos [117].

Although the work that we propose here in this paper does not only focus on how to create or propose an ensemble model (ENS in our case), we could have effectively done additional work in the representation of the data of learning to improve predictive quality [118]–[120]

VI. CONCLUSION

In this paper, we have presented and proposed an approach to calibrate a WO model. After presenting the fundamental principles and the foundations of the problem encountered, we proposed a methodology based on the use of machine learning allowing an effective and efficient calibration of the WO model so that it is automatically configurable. Our approach is based on the use of an ENS ensemble model composed of a series of XGBoost models and deep neural networks in order to increase the predictive capacity of the learning model. The results showed that the ENS model is more efficient in terms of precision, has fewer errors and manages to have a better coverage rate in terms of prediction data.

The approach we have proposed makes it possible to have a tool for calibrating the WO model in order to produce data from the ST model quite close to reality.

Despite the effectiveness of the proposed approach, which is a first of its kind, the model will require improvements and optimization in order to maintain constant performance. Several issues are to be raised and must be addressed in the context of future work, such as the capacity to generalize this type of model, given that the data used for its construction and validation are still limited. Other points must also be raised, in particular when it comes to putting this type of approach into production and asking the question of monitoring this type of models in production in order to avoid drift problems, for example. All these questions and others will be the subject of in-depth analysis.

ACKNOWLEDGMENT

Hamdi Amroun would like to thank the entire Réseau de Transport d'Electricité (RTE) team for placing their trust in them by entrusting them with this very ambitious and interesting project. He would also like to thank the entire EuroBios team in particular John Redford for his valuable explanations and his time, and Maxime Guéguin for his involvement. Without forgetting Emmanuel for the documentation on the Wake Oscillator and its explanations. He would also like to thank Youness El Marhraoui for his precise help in the use and generation of WO data. He also thank their two interns: Massil Ksouri and Nithushan for their involvement.

REFERENCES

[1] L. Zhang, J. Redford, F. Hafid, J.-M. Ghidaglia, and M. Gueguin, ''VIV modelled using simplified cable dynamics coupled to sub-critical cylinder flow simulations in a moving reference frame,'' *Eur. J. Mech.-B/Fluids*, vol. 85, pp. 214–231, Jan. 2021.

[2] R. Violette, E. de Langre, and J. Szydlowski, ''Computation of vortex-induced vibrations of long structures using a wake oscillator model: Comparison with DNS and experiments,'' *Comput. Struct.*, vol. 85, nos. 11–14, pp. 1134–1141, Jun. 2007.

[3] Z. Ge, Z. Song, S. X. Ding, and B. Huang, ''Data mining and analytics in the process industry: The role of machine learning,'' *IEEE Access*, vol. 5, pp. 20590–20616, 2017.

[4] J. C. Chang, S. Amershi, and E. Kamar, ''Revolt: Collaborative crowdsourcing for labeling machine learning datasets,'' in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2017, pp. 2334–2346.

[5] D. C. Mohr, M. Zhang, and S. M. Schueller, ''Personal sensing: Understanding mental health using ubiquitous sensors and machine learning,'' *Annu. Rev. Clin. Psychol.*, vol. 13, no. 1, pp. 23–47, 2017.

[6] D. Antons and C. F. Breidbach, ''Big data, big insights? Advancing service innovation and design with machine learning,'' *J. Service Res.*, vol. 21, no. 1, pp. 17–39, Feb. 2018.

[7] A. O. Oliynyk and J. M. Buriak, ''Virtual issue on machine-learning discoveries in materials science,'' *Chem. Mater.*, vol. 31, no. 20, pp. 8243–8247, Oct. 2019.

[8] Y. Fan, W. Li, N. Chen, J.-H. Ahn, Y.-J. Park, S. Kratzer, T. Schroeder, J. Ishizaka, R. Chang, and K. Stamnes, ''OC-SMART: A machine learning based data analysis platform for satellite ocean color sensors,'' *Remote Sens. Environ.*, vol. 253, Feb. 2021, Art. no. 112236.

[9] A. Romagnoni, S. Jégou, K. Van Steen, G. Wainrib, and J.-P. Hugot, ''Comparative performances of machine learning methods for classifying crohn disease patients using genome-wide genotyping data,'' *Sci. Rep.*, vol. 9, no. 1, pp. 1–18, Dec. 2019.

[10] A. D'Amour *et al.*, ''Underspecification presents challenges for credibility in modern machine learning,'' 2020, *arXiv:2011.03395*.

[11] J. Ludwig, S. Mullainathan, and J. Spiess, ''Augmenting pre-analysis plans with machine learning,'' in *Proc. AEA Papers*, vol. 109, May 2019, pp. 71–76.

[12] R. Fung *et al.*, ''Achieving accurate estimates of fetal gestational age and personalised predictions of fetal growth based on data from an international prospective cohort study: A population-based machine learning study,'' *Lancet Digit. Health*, vol. 2, no. 7, pp. e368–e375, Jul. 2020.

[13] J. Abrell, M. Kosch, and S. Rausch, ''How effective was the UK carbon tax?—A machine learning approach to policy evaluation,'' Mach. Learn. Approach Policy Eval., CER-ETH-Center Econ. Res., ETH Zürich, Zürich, Switzerland, Working Paper, Apr. 2019, p. 317, vol. 19.

[14] A. Tandia, M. C. Onbasli, and J. C. Mauro, ''Machine learning for glass modeling,'' in *Springer Handbook Glass*. Springer, 2019, pp. 1157–1192.

[15] G. L. Hughes, M. A. Lones, M. Bedder, P. D. Currie, S. L. Smith, and M. E. Pownall, ''Machine learning discriminates a movement disorder in a zebrafish model of Parkinson's disease,'' *Disease Models Mech.*, vol. 13, Jan. 2020, Art. no. dmm045815.

[16] M. M. Malik, ''A hierarchy of limitations in machine learning,'' 2020, *arXiv:2002.05193*.

[17] H. Zhao, Q. Hua, H.-B. Chen, Y. Ye, H. Wang, S. X.-D. Tan, and E. Tlelo-Cuautle, ''Thermal-sensor-based occupancy detection for smart buildings using machine-learning methods,'' *ACM Trans. Des. Automat. Electron. Syst.*, vol. 23, no. 4, pp. 1–21, Jun. 2018.

[18] D. M. Probst, M. Raju, P. K. Senecal, J. Kodavasal, P. Pal, S. Som, A. A. Moiz, and Y. Pei, ''Evaluating optimization strategies for engine simulations using machine learning emulators,'' *J. Eng. Gas Turbines Power*, vol. 141, no. 9, Sep. 2019, Art. no. 091011.

[19] N. L. Hepler, K. Scheffler, S. Weaver, B. Murrell, D. D. Richman, D. R. Burton, P. Poignard, D. M. Smith, and S. L. K. Pond, ''IDEPI: Rapid prediction of HIV-1 antibody epitopes and other phenotypic features from sequence data using a flexible machine learning platform,'' *PLoS Comput. Biol.*, vol. 10, no. 9, Sep. 2014, Art. no. e1003842.

[20] M. Yazdani, ''The ideal teaching machine,'' in *Computers and Modern Language Studies*. 1986, pp. 144–153.

[21] A. F. Klaib, N. O. Alsrehin, W. Y. Melhem, H. O. Bashtawi, and A. A. Magableh, ''Eye tracking algorithms, techniques, tools, and applications with an emphasis on machine learning and Internet of Things technologies,'' *Exp. Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 114037.

[22] D. Kinn, ''Reducing estimation risk in mean-variance portfolios with machine learning,'' 2018, *arXiv:1804.01764*.

[23] A. Srivastava, Y. V. Karpievitch, S. R. Eichten, J. O. Borevitz, and R. Lister, ''HOME: A histogram based machine learning approach for effective identification of differentially methylated regions,'' *BMC Bioinf.*, vol. 20, no. 1, pp. 1–15, Dec. 2019.

[24] A. S. G. Smith, ''Application of machine learning algorithms in adaptive web-based information systems,'' Ph.D. dissertation, Middlesex Univ., London, U.K., 1999.

[25] A. Zakutayev, N. Wunder, M. Schwarting, J. D. Perkins, R. White, K. Munch, W. Tumas, and C. Phillips, ''An open experimental database for exploring inorganic materials,'' *Sci. Data*, vol. 5, no. 1, pp. 1–12, Dec. 2018.

[26] A. Bhattacharjee, B. Soni, G. Verma, S. K. Borgohain, and X.-Z. Gao, *Machine Learning, Image Processing, Network Security and Data Sciences: Second International Conference, MIND 2020, Silchar, India, July 30–31, 2020, Proceedings, Part II*, vol. 1241. Springer, 2020.

[27] K. Cooper, C. Baddeley, B. French, K. Gibson, J. Golden, T. Lee, S. Pierre, B. Weiss, and J. Yang, ''Novel development of predictive feature fingerprints to identify chemistry-based features for the effective drug design of SARS-CoV-2 target antagonists and inhibitors using machine learning,'' *ACS Omega*, vol. 6, no. 7, pp. 4857–4877, Feb. 2021.

[28] S. Kim, D. Tasse, and A. K. Dey, ''Making machine-learning applications for time-series sensor data graphical and interactive,'' *ACM Trans. Interact. Intell. Syst.*, vol. 7, no. 2, pp. 1–30, Jul. 2017.

[29] M. S. Norouzzadeh, D. Morris, S. Beery, N. Joshi, N. Jojic, and J. Clune, ''A deep active learning system for species identification and counting in camera trap images,'' 2019, *arXiv:1910.09716*.

[30] S. Pillai, B. Good, D. Richman, and J. Corbeil, ''A new perspective on V3 phenotype prediction,'' *AIDS Res. Hum. Retroviruses*, vol. 19, no. 2, pp. 145–149, Feb. 2003.

[31] N. F. R. Vento, ''Hypothesis-based machine learning for deep-water channel systems,'' Ph.D. dissertation, Colorado State Univ., Fort Collins, CO, USA, 2020.

[32] M. Chandorkar, ''Machine learning in space weather,'' Ph.D. dissertation, Univ. Eindhoven, Eindhoven, The Netherlands, 2019.

[33] S. Agajanian, ''Development of integrated machine learning and data science approaches for the prediction of cancer mutation and autonomous drug discovery of anti-cancer therapeutic agents,'' Ph.D. dissertation, 2020.

[34] A. Yaganapu, ''Detection of SNPS associated with bone loss rate by using machine learning approaches,'' Ph.D. dissertation, 2020.

[35] S. Nam, ''Understanding and supporting vocabulary learners via machine learning on behavioral and linguistic data,'' Ph.D. dissertation, 2020.

[36] S. Bose, ''Towards explainability in machine learning for Malware detection,'' Ph.D. dissertation, Florida State Univ., Tallahassee, FL, USA, 2020.

[37] T. F. Hayes, ''A qualitative exploratory study of emergency medicine clinician perspectives on clinical decision support systems (CDSS) rooted in machine learning in England,'' Ph.D. dissertation, 2020.

[38] L. Shi, ''Leveraging big data and machine learning technologies for accurate and scalable genomic analysis,'' Ph.D. dissertation, Florida State Univ., Tallahassee, FL, USA, 2020.

[39] J. A. Caley, ''A survey of systems for predicting stock market movements, combining market indicators and machine learning classifiers,'' Ph.D. dissertation, 2013.

[40] D. Araci, ''FinBERT: Financial sentiment analysis with pre-trained language models,'' 2019, *arXiv:1908.10063*.

[41] N. Shah, D. Zhou, and Y. Peres, ''Approval voting and incentives in crowdsourcing,'' in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 10–19.

[42] T. Fletcher, ''Polarizable multipolar electrostatics driven by Kriging machine learning for a peptide force field: Assessment, improvement and up-scaling,'' Ph.D. dissertation, Univ. Manchester, Manchester, U.K., 2014.

[43] J. Allen, H. M. Davey, D. Broadhurst, J. J. Rowland, S. G. Oliver, and D. B. Kell, ''Discrimination of modes of action of antifungal substances by use of metabolic footprinting,'' *Appl. Environ. Microbiol.*, vol. 70, no. 10, pp. 6157–6165, Oct. 2004.

[44] V. Mohan, ''Computer vision and machine learning methods for the analysis of brain and cardiac imagery,'' Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, USA, 2010.

[45] I. Petousis, ''Density functional theory and machine learning methods for dielectric materials discovery,'' Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 2016.

[46] K. Swanson, ''Message passing neural networks for molecular property prediction,'' Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2019.

[47] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, ''Machine learning for fluid mechanics,'' *Annu. Rev. Fluid Mech.*, vol. 52, no. 1, pp. 477–508, Jan. 2020.

[48] M. P. Brenner, J. D. Eldredge, and J. B. Freund, ''Perspective on machine learning for advancing fluid mechanics,'' *Phys. Rev. Fluids*, vol. 4, no. 10, Oct. 2019, Art. no. 100501.

[49] M. Raissi, A. Yazdani, and G. Em Karniadakis, ''Hidden fluid mechanics: A Navier–Stokes informed deep learning framework for assimilating flow visualization data,'' 2018, *arXiv:1808.04327*.

[50] J. Ling, R. Jones, and J. Templeton, ''Machine learning strategies for systems with invariance properties,'' *J. Comput. Phys.*, vol. 318, pp. 22–35, Aug. 2016.

[51] M. Raissi, A. Yazdani, and G. E. Karniadakis, ''Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations,'' *Science*, vol. 367, no. 6481, pp. 1026–1030, Feb. 2020.

[52] A. P. Singh, S. Medida, and K. Duraisamy, ''Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils,'' *AIAA J.*, vol. 55, no. 7, pp. 2215–2227, Jul. 2017.

[53] K. Fukami, K. Fukagata, and K. Taira, ''Assessment of supervised machine learning methods for fluid flows,'' *Theor. Comput. Fluid Dyn.*, vol. 34, no. 4, pp. 497–519, Aug. 2020.

[54] S. Shamshirband, S. Hashemi, H. Salimi, S. Samadianfard, E. Asadi, S. Shadkani, K. Kargar, A. Mosavi, N. Nabipour, and K.-W. Chau, ''Predicting standardized streamflow index for hydrological drought using machine learning models,'' *Eng. Appl. Comput. Fluid Mech.*, vol. 14, no. 1, pp. 339–350, Jan. 2020.

[55] W. E. Faller and S. J. Schreck, ''Unsteady fluid mechanics applications of neural networks,'' *J. Aircr.*, vol. 34, no. 1, pp. 48–55, Jan. 1997.

[56] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, ''Machine learning accelerated computational fluid dynamics,'' 2021, *arXiv:2102.01010*.

[57] F. Cichos, K. Gustavsson, B. Mehlig, and G. Volpe, ''Machine learning for active matter,'' *Nature Mach. Intell.*, vol. 2, no. 2, pp. 94–103, Feb. 2020.

[58] L. Zhu, W. Zhang, J. Kou, and Y. Liu, ''Machine learning methods for turbulence modeling in subsonic flows around airfoils,'' *Phys. Fluids*, vol. 31, no. 1, Jan. 2019, Art. no. 015105.

[59] Z. M. Yaseen, A. M. Al-Juboori, U. Beyaztas, N. Al-Ansari, K.-W. Chau, C. Qi, M. Ali, S. Q. Salih, and S. Shahid, ''Prediction of evaporation in arid and semi-arid regions: A comparative study using different machine learning models,'' *Eng. Appl. Comput. Fluid Mech.*, vol. 14, no. 1, pp. 70–89, Jan. 2020.

[60] K. Fukami, K. Fukagata, and K. Taira, ''Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows,'' *J. Fluid Mech.*, vol. 909, pp. 1–21, Feb. 2021.

[61] Y. Bahri, J. Kadmon, J. Pennington, S. S. Schoenholz, J. Sohl-Dickstein, and S. Ganguli, ''Statistical mechanics of deep learning,'' *Annu. Rev. Condens. Matter Phys.*, vol. 11, pp. 501–528, Mar. 2020.

[62] Z. J. Zhang and K. Duraisamy, ''Machine learning methods for data-driven turbulence modeling,'' in *Proc. 22nd AIAA Comput. Fluid Dyn. Conf.*, Jun. 2015, p. 2460.

[63] N. Muttil and K.-W. Chau, ''Machine-learning paradigms for selecting ecologically significant input variables,'' *Eng. Appl. Artif. Intell.*, vol. 20, no. 6, pp. 735–744, Sep. 2007.

[64] P. Garnier, J. Viquerat, J. Rabault, A. Larcher, A. Kuhnle, and E. Hachem, ''A review on deep reinforcement learning for fluid mechanics,'' 2019, *arXiv:1908.04127*.

[65] K. Fukami, Y. Nabae, K. Kawai, and K. Fukagata, ''Synthetic turbulent inflow generator using machine learning,'' *Phys. Rev. Fluids*, vol. 4, no. 6, Jun. 2019, Art. no. 064603.

[66] N. Gautier, T. Duriez, J.-L. Aider, B. Noack, M. Segond, and M. Abel, ''Closed-loop separation control using machine learning,'' 2014, *arXiv:1405.0908*.

[67] S. Sankaran, L. Grady, and C. A. Taylor, ''Impact of geometric uncertainty on hemodynamic simulations using machine learning,'' *Comput. Methods Appl. Mech. Eng.*, vol. 297, pp. 167–190, Dec. 2015.

[68] T. Duriez, V. Parezanovic, J.-C. Laurentie, C. Fourment, J. Delville, J.-P. Bonnet, L. Cordier, B. R. Noack, M. Segond, M. W. Abel, N. Gautier, J.-L. Aider, C. Raibaudo, C. Cuvier, M. Stanislas, and S. L. Brunton, ''Closed-loop control of experimental shear flows using machine learning,'' in *Proc. 7th AIAA Flow Control Conf.*, Jun. 2014, p. 2219.

[69] M. Alber, A. B. Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold, and E. Kuhl, ''Integrating machine learning and multiscale modeling—Perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences,'' *npj Digit. Med.*, vol. 2, no. 1, pp. 1–11, Dec. 2019.

[70] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman, ''Universal differential equations for scientific machine learning,'' 2020, *arXiv:2001.04385*.

[71] J. Rabault, F. Ren, W. Zhang, H. Tang, and H. Xu, ''Deep reinforcement learning in fluid mechanics: A promising method for both active flow control and shape optimization,'' *J. Hydrodyn.*, vol. 32, no. 2, pp. 234–246, Apr. 2020.

[72] J. Weatheritt, R. Pichler, R. D. Sandberg, G. Laskowski, and V. Michelassi, ''Machine learning for turbulence model development using a high-fidelity hpt cascade simulation,'' in *Turbo Expo, Power for Land, Sea, and Air*, vol. 50794. New York, NY, USA: American Society Mechanical Engineers, 2017, pp. 1–13.

[73] Y. Zhang, W. J. Sung, and D. N. Mavris, ''Application of convolutional neural network to predict airfoil lift coefficient,'' in *Proc. AIAA/ASCE/AHS/ASC Struct., Struct. Dyn., Mater. Conf.*, Jan. 2018, p. 1903.

[74] K. Fukami, K. Fukagata, and K. Taira, ''Super-resolution reconstruction of turbulent flows with machine learning,'' 2018, *arXiv:1811.11328*.

[75] Z. Ti, X. W. Deng, and H. Yang, ''Wake modeling of wind turbines using machine learning,'' *Appl. Energy*, vol. 257, Jan. 2020, Art. no. 114025.

[76] J.-X. Wang, J. Wu, J. Ling, G. Iaccarino, and H. Xiao, ''A comprehensive physics-informed machine learning framework for predictive turbulence modeling,'' 2017, *arXiv:1701.07102*.

[77] C. Tesche, C. N. De Cecco, S. Baumann, M. Renker, T. W. McLaurin, T. M. Duguay, R. R. Bayer, D. H. Steinberg, K. L. Grant, C. Canstein, C. Schwemmer, M. Schoebinger, L. M. Itu, S. Rapaka, P. Sharma, and U. J. Schoepf, ''Coronary CT angiography–derived fractional flow reserve: Machine learning algorithm versus computational fluid dynamics modeling,'' *Radiology*, vol. 288, no. 1, pp. 64–72, Jul. 2018.

[78] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, ''Automatic differentiation in machine learning: A survey,'' *J. Mach. Learn. Res.*, vol. 18, pp. 1–43, Apr. 2018.

[79] J. Park and H. Choi, ''Machine-learning-based feedback control for drag reduction in a turbulent channel flow,'' *J. Fluid Mech.*, vol. 904, Dec. 2020.

[80] R. Matai and P. A. Durbin, ''Zonal eddy viscosity models based on machine learning,'' *Flow, Turbulence Combustion*, vol. 103, no. 1, pp. 93–109, Jun. 2019.

[81] M. Ghalandari, A. Ziamolki, A. Mosavi, S. Shamshirband, K.-W. Chau, and S. Bornassi, ''Aeromechanical optimization of first row compressor test stand blades using a hybrid machine learning model of genetic algorithm, artificial neural networks and design of experiments,'' *Eng. Appl. Comput. Fluid Mech.*, vol. 13, no. 1, pp. 892–904, Jan. 2019.

[82] M. Molina and F. Garip, ''Machine learning for sociology,'' *Annu. Rev. Sociol.*, vol. 45, no. 1, pp. 27–45, Jul. 2019.

[83] K. Hasegawa, K. Fukami, T. Murata, and K. Fukagata, ''Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes,'' *Theor. Comput. Fluid Dyn.*, vol. 34, no. 4, pp. 367–383, Aug. 2020.

[84] Z. Y. Wan and T. P. Sapsis, ''Machine learning the kinematics of spherical particles in fluid flows,'' *J. Fluid Mech.*, vol. 857, pp. 1–11, Dec. 2018.

[85] E. Jumin, N. Zaini, A. N. Ahmed, S. Abdullah, M. Ismail, M. Sherif, A. Sefelnasr, and A. El-Shafie, ''Machine learning versus linear regression modelling approach for accurate ozone concentrations prediction,'' *Eng. Appl. Comput. Fluid Mech.*, vol. 14, no. 1, pp. 713–725, Jan. 2020.

[86] A. Etemad-Shahidi, R. Yasa, and M. H. Kazeminezhad, ''Prediction of wave-induced scour depth under submarine pipelines using machine learning approach,'' *Appl. Ocean Res.*, vol. 33, no. 1, pp. 54–59, Feb. 2011.

[87] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, M. Walczak, J. Garcke, C. Bauckhage, and J. Schuecker, ''Informed machine learning—A taxonomy and survey of integrating knowledge into learning systems,'' 2019, *arXiv:1903.12394*.

[88] Y. Dabiri, A. Van Der Velden, K. L. Sack, J. S. Choy, G. S. Kassab, and J. M. Guccione, ''Prediction of left ventricular mechanics using machine learning,'' *Frontiers Phys.*, vol. 7, p. 117, Sep. 2019.

[89] S. J. Mooney and V. Pejaver, ''Big data in public health: Terminology, machine learning, and privacy,'' *Annu. Rev. Public Health*, vol. 39, no. 1, pp. 95–112, Apr. 2018.

[90] A. Chiuso and G. Pillonetto, ''System identification: A machine learning perspective,'' *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 2, no. 1, pp. 281–304, May 2019.

[91] R. Ranade, C. Hill, and J. Pathak, ''DiscretizationNet: A machine-learning based solver for Navier–Stokes equations using finite volume discretization,'' *Comput. Methods Appl. Mech. Eng.*, vol. 378, May 2021, Art. no. 113722.

[92] A. T. W. Min, R. Sagarna, A. Gupta, Y.-S. Ong, and C. K. Goh, ''Knowledge transfer through machine learning in aircraft design,'' *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 48–60, Nov. 2017.

[93] N. Savage, ''Better medicine through machine learning,'' *Commun. ACM*, vol. 55, no. 1, pp. 17–19, Jan. 2012.

[94] L. Joss and E. A. Müller, ''Machine learning for fluid property correlations: Classroom examples with MATLAB,'' *J. Chem. Educ.*, vol. 96, no. 4, pp. 697–703, Apr. 2019.

[95] T. I. Zohdi, ''A machine-learning framework for rapid adaptive digital-twin based fire-propagation simulation in complex environments,'' *Comput. Methods Appl. Mech. Eng.*, vol. 363, May 2020, Art. no. 112907.

[96] S. Kolouri, S. R. Park, M. Thorpe, D. Slepčev, and G. K. Rohde, ''Optimal mass transport: Signal processing and machine-learning applications,'' *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 43–59, Jul. 2017.

[97] S. Athey and G. W. Imbens, ''Machine learning methods that economists should know about,'' *Annu. Rev. Econ.*, vol. 11, no. 1, pp. 685–725, Aug. 2019.

[98] S. U. Gülek, ''Machine learning for fluid mechanics applications,'' Tech. Rep., 2020.

[99] S. Shamshirband, A. Mosavi, T. Rabczuk, N. Nabipour, and K.-W. Chau, ''Prediction of significant wave height; comparison between nested grid numerical model, and machine learning models of artificial neural networks, extreme learning and support vector machines,'' *Eng. Appl. Comput. Fluid Mech.*, vol. 14, no. 1, pp. 805–817, Jan. 2020.

[100] J. Ling and J. Templeton, ''Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty,'' *Phys. Fluids*, vol. 27, no. 8, Aug. 2015, Art. no. 085103.

[101] C. Ooi, Q. T. Le, M. H. Dao, V. B. Nguyen, H. H. Nguyen, and T. Ba, ''Modeling transient fluid simulations with proper orthogonal decomposition and machine learning,'' *Int. J. Numer. Methods Fluids*, vol. 93, no. 2, pp. 396–410, Feb. 2021.

[102] J. N. Kutz, ''Deep learning in fluid dynamics,'' *J. Fluid Mech.*, vol. 814, pp. 1–4, Jan. 2017.

[103] Q. Zhu, Z. Liu, and J. Yan, ''Machine learning for metal additive manufacturing: Predicting temperature and melt pool fluid dynamics using physics-informed neural networks,'' *Comput. Mech.*, vol. 67, pp. 1–17, Feb. 2020.

[104] K. Herfjord, T. Holmås, and K. Randa, ''A parallel approach for numerical solution of vortex induced vibrations of very long risers,'' in *Proc. Int. OMAE Symp., Buenos Aires*, 1998.

[105] Y. Duanmu, L. Zou, and D.-C. Wan, ''Numerical simulations of vortex-induced vibrations of a flexible riser with different aspect ratiosin uniform and shear currents,'' *J. Hydrodyn.*, vol. 29, no. 6, pp. 1010–1022, Dec. 2017.

[106] Y. Bao, R. Palacios, M. Graham, and S. Sherwin, ''Generalized thick strip modelling for vortex-induced vibration of long flexible cylinders,'' *J. Comput. Phys.*, vol. 321, pp. 1079–1097, Sep. 2016.

[107] M. Zinkevich. (2017). *Rules of Machine Learning: Best Practices for ML Engineering*. [Online]. Available: https://developers.google.com/machine-learning/guides/rules-of-ml

[108] J. Lupón, H. K. Gaggin, M. de Antonio, M. Domingo, A. Galán, E. Zamora, J. Vila, J. Peñafiel, A. Urrutia, E. Ferrer, N. Vallejo, J. L. Januzzi, and A. Bayes-Genis, ''Biomarker-assist score for reverse remodeling prediction in heart failure: The ST2-R2 score,'' *Int. J. Cardiol.*, vol. 184, pp. 337–343, Apr. 2015.

[109] T. Chai and R. R. Draxler, ''Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature,'' *Geosci. Model Develop.*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014.

[110] W. Wang and Y. Lu, ''Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model,'' *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 324, no. 1, p. 12049, 2018.

[111] A. N. Ahmed, F. B. Othman, H. A. Afan, R. K. Ibrahim, C. M. Fai, M. S. Hossain, M. Ehteram, and A. Elshafie, ''Machine learning methods for better water quality prediction,'' *J. Hydrol.*, vol. 578, Nov. 2019, Art. no. 124084.

[112] O. B. Olalusi and P. Spyridis, ''Machine learning-based models for the concrete breakout capacity prediction of single anchors in shear,'' *Adv. Eng. Softw.*, vol. 147, Sep. 2020, Art. no. 102832.

[113] L. Yao and Z. Ge, ''Big data quality prediction in the process industry: A distributed parallel modeling framework,'' *J. Process Control*, vol. 68, pp. 1–13, Aug. 2018.

[114] A. Ferencek and M. K. Borštnar, ''Data quality assessment in product failure prediction models,'' *J. Decis. Syst.*, pp. 1–8, Jun. 2020.

[115] Z. Wang, D. Xiao, F. Fang, R. Govindan, C. C. Pain, and Y. Guo, ''Model identification of reduced order fluid dynamics systems using deep learning,'' *Int. J. Numer. Methods Fluids*, vol. 86, no. 4, pp. 255–268, Feb. 2018.

[116] J. Zhang and W. Ma, ''Data-driven discovery of governing equations for fluid dynamics based on molecular simulation,'' *J. Fluid Mech.*, vol. 892, pp. 1–18, Jun. 2020.

[117] X. Yin, J. Wu, and H. Xiao, ''Recognition of permeability from images with convolutional neural networks,'' *Bull. Amer. Phys. Soc.*, vol. 63, Nov. 2018.

[118] A. O. Al-Sulttani, M. Al-Mukhtar, A. B. Roomi, A. A. Farooque, K. M. Khedher, and Z. M. Yaseen, ''Proposition of new ensemble data-intelligence models for surface water quality prediction,'' *IEEE Access*, vol. 9, pp. 108527–108541, 2021.

[119] S. Shamshirband, E. J. Nodoushan, J. E. Adolf, A. A. Manaf, A. Mosavi, and K.-W. Chau, ''Ensemble models with uncertainty analysis for multi-day ahead forecasting of chlorophyll a concentration in coastal waters,'' *Eng. Appl. Comput. Fluid Mech.*, vol. 13, no. 1, pp. 91–101, Jan. 2019.

[120] M. Ehteram, A. N. Ahmed, P. Kumar, M. Sherif, and A. El-Shafie, ''Predicting freshwater production and energy consumption in a seawater greenhouse based on ensemble frameworks using optimized multi-layer perceptron,'' *Energy Rep.*, vol. 7, pp. 6308–6326, Nov. 2021.

HAMDI AMROUN received the Ph.D. degree in machine learning jointly from the University of Paris-Saclay and IPAL, Singapore. He worked for years on several related subjects, including e-health, physical activity analysis, marketing, e-commerce, quantitative finance, and fluid mechanics. His main interest is in the topics of data analysis and deep learning models.

MEHDI AMMI received the degree in electronics engineering in 2000 and the Ph.D. degree in robotics in 2005. He is currently pursuing the Habilitation to Supervise Research (HDR) degree in computer science. He is also a Professor with the University of Paris 8. He was the Leader or Co-Leader of several international working groups, including IEEE TCH and EuroVR Haptic SIG, and was involved in more than 20 national and international projects and industrial collaborations. His research focuses on the study of pervasive environments and their societal and industrial applications, such as e-health, smart building, and factory of the future. He addresses this field with a multidisciplinary approach combining artificial intelligence (such as machine learning and neurosciences), mechatronic technologies (such as organic electronics and smart textiles), and human factors (such as psychology and behavioral analysis).

FIKRI HAFID received the M.Sc. and Ph.D. degrees from the Ecole Normale Supérieure Paris-Saclay and the degree in engineering from the Ecole Centrale de Marseille. He is currently a Research Engineer with the Réseaux de Transport d'Electricité (RTE). He is also a Research Associate with the Centre Borelli, ENS Paris-Saclay, and MSDA, UM6P. His research interests include CFD, AI, asset management, scientific computing, and HPC.