Aerospace Science and Technology 165 (2025) 110466

Contents lists available at ScienceDirect

Aerospace Science and Technology

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

Original article

Predicting airfoil dynamic stall loads using neural networks

Emanuel A.R. Camacho [a,*], André R.R. Silva [a], Flávio D. Marques [b]

[a] Aeronautics and Astronautics Research Center (AEROG-LAETA), Covilhã, CB, Portugal
[b] Laboratory of Dynamics (LABDIN), São Carlos School of Engineering, São Carlos, São Paulo, Brazil

ARTICLE INFO

Communicated by Cummings Russell

Keywords:
Dynamic stall
Artificial intelligence
Pitching airfoil
Experimental data integration

ABSTRACT

Dynamic stall is an aerodynamic regime characterized by loss of airfoil lift, drag increment, and abrupt changes in the pitching moment. Such effects can couple with structural dynamics where perturbations can be easily amplified, making this a critical phenomenon that jeopardizes operational safety. Hence, there is always the need to constantly study the basics of dynamic stall and provide newer predictive models that can take advantage of the current interest peak in artificial intelligence. The present work builds upon that need, exploring the ability of a simple feed-forward network to predict the oscillation cycle of a pitching airfoil experiencing from light to deep stall of a NACA0012 airfoil close to a Reynolds number of approximately 1.4×10^6. The proposed neural network uses the angle of attack and its rate of change as inputs, then estimates the whole aerodynamic cycle at once, outputting an aggregated vector of drag, lift, and pitching moment coefficients. The training phase was conducted using a database containing several conditions obtained from experimental tests, with a strict convergence criterion of $R^2 = 0.99$ for both training and test datasets. Results show that the neural network, even in the least-performing conditions, can capture the aerodynamics and overall tendencies, even if some dynamics are underrepresented in the training dataset. The present work brings down the complexity of methodology while demonstrating that a simplistic architecture can still offer an accurate dynamic stall model.

1. Introduction

Unsteady aerodynamics, as a subject area, combines several essential study fields and applications that act as the foundation for many systems and technologies. One of these critical research fields concerns the dynamic stall regime, which is receiving enormous attention mainly due to its significant impact on the performance and safety of various systems.

At the earlier stages of dynamic stall exploration, it was suggested that if dynamic stall phenomena were better understood, there would be an easier way to improve airfoil design [1]. Dynamic stall is an unsteady aerodynamic phenomenon that occurs when an airfoil rapidly exceeds a critical stall angle, leading to flow separation and the formation of a dynamic stall vortex (DSV). This process usually involves four phases: a phase of flow attachment, onset of flow separation, followed by vortex formation and lift enhancement, vortex shedding with abrupt lift loss, and then, flow reattachment as the angle of attack decreases [2]. These phases result in significant oscillations in drag, lift, and moment, which, as aforementioned, have a significant influence on performance. However, while there is considerable research on dynamic stall and its adverse effects, currently there is no fast and reliable way to estimate unsteady aerodynamics given a specific periodic oscillation [3]. The current spike in interest in artificial intelligence (AI) concepts can accelerate the intersection between aerodynamics and neural networks-based models, as these have the potential to capture and generalize the underlying patterns and phenomena that govern dynamic stall behavior [4].

The capacity of AI modeling to predict unsteady flows was leveraged by Miotto and Wolf [5], who considered a convolutional neural network (CNN) to model the aerodynamic response from flowfield images. In fact, the results showed that the neural network could correlate important flow characteristics with the airfoil response, indicating that such a methodology could be an alternative to traditional high-cost modeling. Aside from load prediction, convolutional neural networks also have the potential to predict the evolution of coherent structures around an airfoil [6], which can further enhance the correlations between loads and flow features. Additional airfoil characteristics, such as the airfoil pressure distribution response regarding the variation of the angle of attack, can also take advantage of machine learning [7].

* Corresponding author.
E-mail addresses: earc96@icloud.com, earc.aerog@gmail.com (E.A.R. Camacho).

https://doi.org/10.1016/j.ast.2025.110466
Received 10 January 2025; Received in revised form 3 June 2025; Accepted 9 June 2025
Available online 16 June 2025

Also focused on AI modeling driven by flow field images and physical representations, Mi and Cheng [8] developed a straightforward method that avoids considering aerofoil geometry information while eliminating the necessity of discretely predicting the flow field. Others, for instance, Damiola et al. [9], used a state-space neural network to model the unsteady lift force of a pitching airfoil, capturing highly nonlinear flow features like stall vortex formation, with the authors reinforcing that the followed procedure is valuable for engineering applications involving design optimization and real-time control. Kasmaiee et al. [10] used neural networks to train the aerodynamic coefficients as a function of control parameters from a suction-based active flow control system, being these the suction location, velocity, opening length, and suction jet angle relative to the airfoil surface. With the use of artificial intelligence, the authors report a considerable drag reduction with a lift improvement, all achieved with the suppression of the dynamic stall vortex. Dynamic stall mitigation can also come from shape optimization, as shown by Liu et al. [11], who developed a surrogate model based on deep learning connected with a multi-island genetic algorithm. With this methodology, the rotor airfoil was successfully optimized, and a noticeable reduction in the drag and pitching moment peaks was obtained.

The dynamic stall phenomenon also plays a crucial role in horizontal-axis wind turbines. Therefore, accurate modeling and prediction of the dynamic stall is required. This motivated Shi et al. [12] to explore a data-knowledge fusion method that merges physical knowledge into a neural network. Others, as Baldan and Guardone [13], also use a physics-based loss function linked to a convolutional neural network, showing remarkable performance while predicting aerodynamic loads in a wide range of conditions.

In the wind energy field context, Küppers and Reinicke [14] presented a methodology employing a fully stochastic machine learning model, trying to model somehow the load peaks and their uncertainty since averaging can blur many aerodynamic features. Indeed, modeling imprecisions are noted by the authors as a significant limitation. For instance, a RANS approach will not truly capture cycle variations, and scale-solving models require considerable computational power. Nevertheless, Wang et al. [15] showed that the synergy between experimental and simulation data can reduce the prediction error up to five times when compared to conventional CFD methodologies.

It was also noted by Damiola et al. [16] who focused on the cycle-to-cycle variability, discussing the limitations of conventional phase-average, which can lead to inaccuracies in some conditions. At the root of their study, they proceed with the state-space neural network coupled with an extended Kalman filter, showing that it could dynamically embody experimental data while enhancing model accuracy. These methodologies can be helpful in developing a neural network-based reduced order model (ROM) for predicting the aerodynamic forces [17] or creating a nonlinear aeroelastic model based on artificial intelligence. Compared to traditional aeroelastic computational fluid dynamics simulations, AI-based reduced order models provide a noticeable decrease in computation resources without sacrificing accuracy, as Torregrosa et al. [18] explored.

As the dynamic stall phenomenon is inherently time-dependent, various research resort to conventional time-dependent neural networks. Zhang et al. [19] designed a recurrent neural network (RNN) to extract the sequential force data based on the prescribed kinematics and past airfoil dynamics. After training with a group of sinusoidal chirp signals with varying amplitude and frequency, the RNN showed its potential by accurately predicting the airfoil dynamics.

Model accuracy with limited data is also a desired outcome. Wang et al. [20] used a dynamic derivative model coupled with a fuzzy neural network model, showing that even with sparse experimental data, the model could achieve reasonable accuracy. The authors indicate that such methods are beneficial when experimental data is limited, reducing the dependence on high-fidelity data.

While a considerable amount of research applying AI methods has been conducted, significant challenges remain concerning applying neural networks to dynamic stall modeling. A critical aspect in this field is the lower use and lack of experimental data in the training process, with most research resorting to numerical results, which have several limitations [21]. The present work sheds some light on this aspect by using a database of experimental data containing cases affected by airfoil dynamic stall behavior. Furthermore, the paper shows how a conventional feedforward neural network (FNN) can offer a simple yet compelling solution to estimate the unsteady aerodynamics of a pitching airfoil, even with all the variability that comes with experimental measurements.

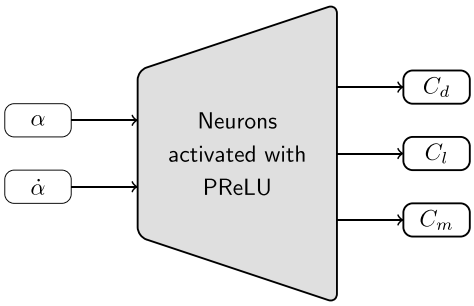

Fig. 1. FNN Architecture.

2. Methodology

The methodology used in this study involves developing a neural network capable of estimating two-dimensional aerodynamic coefficients for a diverse range of oscillating conditions. These conditions are characterized by sinusoidal waves, representing systems that experience periodic dynamic stall behavior.

2.1. Neural network architecture

A feedforward neural network (FNN) is considered to reach the main objective, where kinematics (α and $\dot{\alpha}$) are taken as inputs, and the three aerodynamic coefficients (C_d, C_l, and C_m) are the outputs. The selection of an FNN was motivated by its ability to model complex, nonlinear relationships between input features (such as angle of attack, and its rate of change) and aerodynamic outputs, thus being a great alternative to predict dynamic stall, known for its high nonlinearity. The FNN architecture is well-suited for regression tasks where each prediction is based solely on the current input, making it effective for mapping the aerodynamic response of an airfoil under unsteady conditions without requiring internal memory of previous states, which would otherwise require more intense training processes. In fact, the use of an FNN offers advantages in terms of computational simplicity and training efficiency, making it suitable for engineering applications where rapid evaluation and integration into larger simulation frameworks are required.

The neural network and adjacent blocks are all developed from the ground up in GNU Octave, allowing for greater control and upgradability in the future. A generic representation of the neural network architecture is shown in Fig. 1. Further details on its internal architecture are given in the results obtained during the training phase.

For the present work, the input layer is formally given by the input vector $\mathbf{x}$ that contains the periodic history of the angle of attack, α, and the pitching velocity, $\dot{\alpha}$, as

$$\mathbf{x} = \begin{bmatrix} \alpha & \dot{\alpha} \end{bmatrix}^{\mathrm{T}}. \tag{1}$$

Then, information from the inputs enters and keeps flowing through the neural network until reaching the output layer, recurring to

$$z_j^h = \sum_i w_{ij}^h a_i^{h-1} + b_j^h, \tag{2}$$

where z_j^h is the net input to neuron j in layer h, w_{ij}^h is the weight connecting neuron i in layer $h-1$ to neural j in layer h, a_i^{h-1} is the

activation of neuron i in layer $h-1$, and b_j^h is the bias of neuron j in layer h.

Concerning the activation phase, as one wants to create the ability to predict complex and nonlinear problems, nonlinearity is introduced into the neural network through a nonlinear function, where neurons are activated as

$$a_j = \sigma(z_j) \ , \tag{3}$$

with $\sigma(\cdot)$ being the activation function, assumed in the present work as the Parametric Rectified Linear Unit (PReLU) given by

$$\sigma(z) = \begin{cases} z & \text{if } z \geq 0 \\ cz & \text{if } z < 0 \end{cases} \ , \tag{4}$$

with $c = 0.1$.

At the first hidden layer, a_i^{h-1} is simply considered the input vector, $\mathbf{x}$. Concerning the output layer, it follows the same recurrence presented in Equation (2). However, no activation is performed, meaning that it uses a linear activation function, where the output is simply the weighted sum of the previous layer plus biases, without any modification. In the present work, the output layer should provide the drag, lift, and moment coefficients, expressed in vectorial form as

$$\mathbf{y} = \begin{bmatrix} C_d & C_l & C_m \end{bmatrix}^{\mathrm{T}}, \tag{5}$$

with $C_d = 2D/\rho U^2 S$, $C_l = 2L/\rho U^2 S$, $C_m = 2M/\rho U^2 Sc$, where D, L, M are the lift, drag and pitching moment, ρ is the fluid viscosity, U is the wind speed, S is the wing area and c is the chord length.

In its final architecture, the neural network has 200 inputs, a first hidden layer with 200, a second with 300, and at the output, 300 elements. Regarding the input layer, there are 100 points with the periodic oscillation of the angle of attack, followed by 100 elements corresponding to pitching velocity. On the other hand, the output layer has a sequence of elements representing the drag, lift, and moment coefficients, each with 100 elements, resulting in 300 outputs. With this architecture, one can simply input the whole cycle kinematics information and get the expected aerodynamic behavior.

2.2. Data preparation and training phase

The neural network is trained following a supervised learning strategy, assuming a known dataset. For the present work, the dynamic stall database reported by Green and Giuni [22] is used, obtained during experimental tests at the James Watt School of Engineering of the University of Glasgow. From the vast database, only the available data from the NACA0012 airfoil subjected to sinusoidal pitching motion while under bidimensional conditions was selected. The total number of corresponding cases is 223, all with a Reynolds number close to 1.4×10^6. For more details, refer to Ref. [22].

Before the training phase, the database is preprocessed to standardize the inputs, as the FNN is not input-flexible. Consequently, as the neural network requires, every condition goes through an interpolation subroutine before entering the neural network, creating an array with 100 elements. Additionally, to improve the convergence and learning performance, all variables, from inputs to outputs, are normalized by their respective maximum values. Furthermore, the standardization process also required phase-correcting the kinematics, which is a hard requirement during the data preparation. This procedure guarantees that each input neuron corresponds to a specific time frame in the period. Otherwise, estimates would be intrinsically dependent on the kinematics phase. As the FNN does not operate in time, this means that to get an accurate estimate of a specific condition, one needs to know the kinematics and the phase that would maximize accuracy, which is not known *a priori*. Accordingly, all conditions are time-shifted to the waveform,

$$\alpha = \overline{\alpha} + A_\alpha \cos(2\pi f t), \tag{6}$$

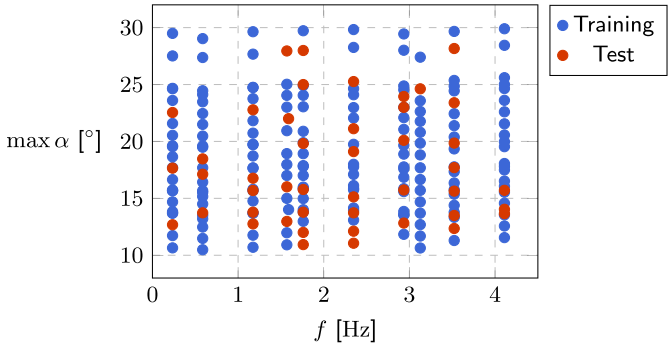

Fig. 2. Training and test datasets in the max α and f domain. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

where $\overline{\alpha}$ is the mean angle of attack, A_α is the pitching angle amplitude (degrees) and f is the pitching frequency (Hz).

For the specific training of the current neural network, one strives to minimize the FNN's outputs and the dataset targets, making this a typical regression problem. The mean squared error (MSE) is used as the loss function, $\mathcal{L}$, given by

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{7}$$

which should be minimized through optimization, where y_i is the true value, $\hat{y}_i$ is the FNN estimate and N is the sample size.

In the present work, training is performed with the gradient descent coupled with the Adaptive Moment Estimation (Adam) extension which incorporates adaptive momentum to improve the convergence speed and accuracy of training [23]. The complete FNN implementation is available at the repository neuroStall [24] under the MIT license.

3. Results and discussion

This section shows the training and testing phase results, focusing on the architecture selection, training quality, and overall performance of the proposed FNN regarding its potential to predict the temporal evolution of aerodynamic coefficients when the airfoil operates in dynamic stall regimes. The dynamic stall database reported by Green and Giuni [22] from the University of Glasgow was selected to proceed with the training and testing phases.

3.1. Training process

The training phase starts by randomly creating two subsets of data from the 223 cases in the database from the University of Glasgow. The first group, the training group, contains 80 % of the database and is used to properly train the neural network by minimizing equation (7). The second group, with 20 % of the database, is used to evaluate the generalization capabilities of the FNN and monitor the occurrence of an overfitting regime. These two data subsets are shown in Fig. 2, where the combination of motion frequency and maximum angle of attack reached during the oscillation can be seen.

Regarding the convergence and training completion, it was considered that the neural network should be trained under one thousand epochs with the R^2 of all test cases reaching at least 0.99. This procedure required a brief iterative study to identify a neural network architecture that could achieve such requirements. One aspect that should be mentioned is that the training vector is shuffled before each training epoch to remove any case sequence biases. Due to the random selection of conditions, such a condition can generate different convergence rates and curves from those presented in this Section if one wants to rerun the training phase.

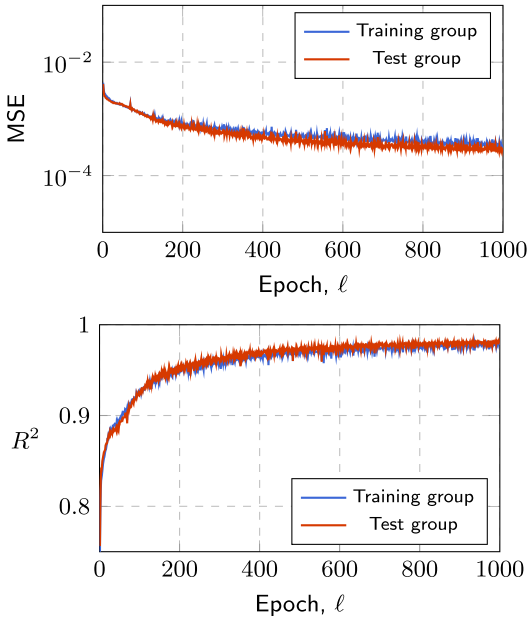

Fig. 3. MSE and R^2 during training considering one hidden layer with 200 neurons.

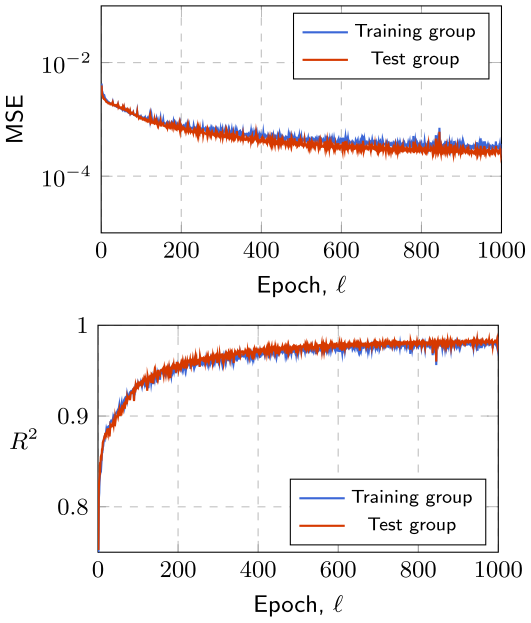

Fig. 4. MSE and R^2 during training considering one hidden layer with 300 neurons.

A sensitivity test regarding the number of internal neurons was carried out. An FNN possessing only one internal layer with 200 elements was considered first. Afterward, the number of neurons was increased to 300, showing the results in Figs. 3 and 4, respectively. As one can see, the FNN with one hidden layer, regardless of the number of elements, could not achieve the trained status, as it did not meet the established criteria under 1000 epochs.

An FNN with just one hidden layer did not favor the training process and must be upgraded to include an additional hidden layer. Now, with two hidden layers, the neural network is trained with a first layer containing 200 neurons and a second layer with 300 neurons that connect to the 300 outputs. The MSE and R^2 evolution over time are shown in Fig. 5. With the additional hidden layer, the FNN training convergence was achieved by verifying the established threshold, which was short

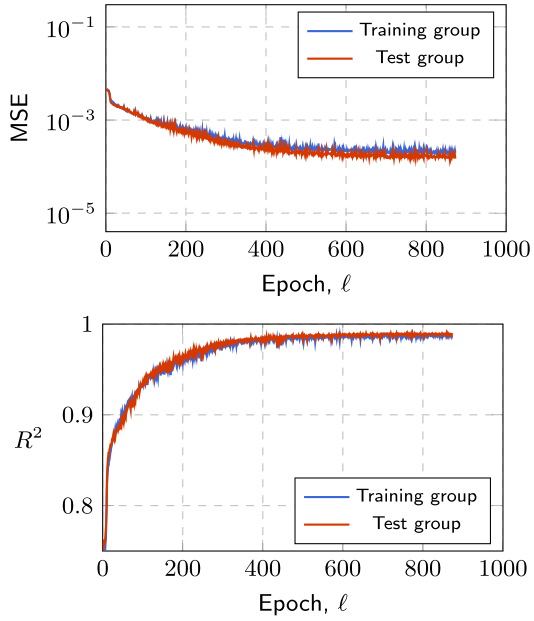

Fig. 5. MSE and R^2 during training considering two hidden layers with 200 and 300 neurons.

of 900 epochs, without reaching an overfitted state. While the R^2 is at least 0.99 for all cases put together, we should also note that some individual cases do not meet the threshold. These will be quantified further, viewing a broader analysis of the FNN performance.

Following the sensitivity analysis, the FNN training moves to the phase where we evaluate its capability to predict the aerodynamic behavior of oscillating cases. One quick and visual way to verify the predictive capabilities of the neural network is to plot the estimated and desired values (ground truth) for the two subsets of data (*cf.* Fig. 6), where the three aerodynamic coefficients are plotted separately.

Fig. 6 reveals that the training group exhibits a point cloud around the $y = x$ function, indicating that the neural network's output is near the actual values, regardless of the aerodynamic coefficient. The same can be seen in the test group, which indicates that the neural network is effectively generalizing from the training data. Nevertheless, one can still notice cloud dispersion, which is likely a consequence of the nature of experimental data, such as measurement noise, artifacts, or experimental inconsistencies.

For a more precise quantitative analysis, Fig. 7 shows the R^2 frequency distribution for each individual case in the training and test groups. As observed, both sets have a similar distribution, reinforcing that the neural network is generalizing well in the test group. Indeed, one can notice that some cases occurring in both groups did not reach the $R^2 = 0.99$ threshold, which might indicate that some of these examples belong to underrepresented regions or that some specific dynamics are not being learned well. Nevertheless, all cases have at least a R^2 of at least 0.90, indicating that the estimates given by the neural network are a good fit compared to measured data.

However, when the R^2 is used in this manner, where all cases and aerodynamic coefficients are aggregated, each with the same importance, an important reminder must be made. Could this consideration add a substantial bias during training, as C_m values are typically much smaller than those of C_d and C_l, consequently reducing the C_m importance in the R^2 value? While this was not a significant limitation in the present work, judging by the training group in Fig. 6, it could be considered for future research either segregating each coefficient with a smaller neural network or preparing a training that considers weighted error metrics to provide the same level of impact to the R^2 calculation.

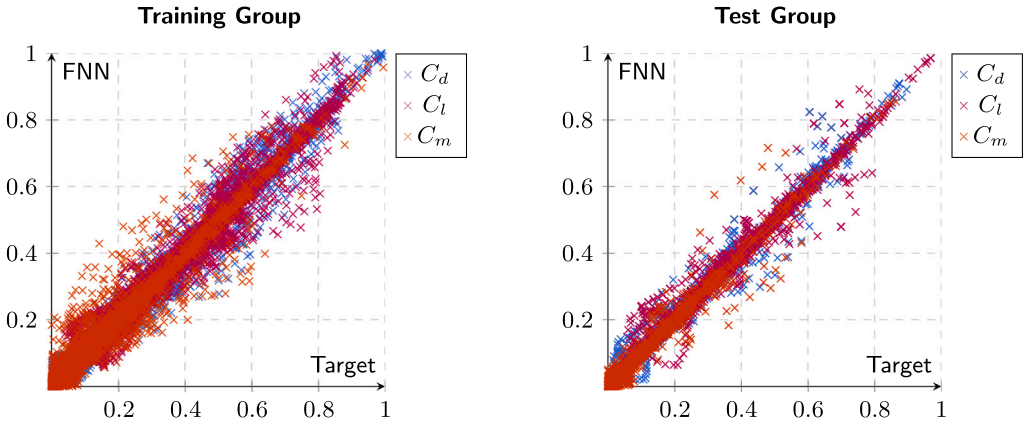

Fig. 6. FNN estimates *vs* targets of both training and test groups.

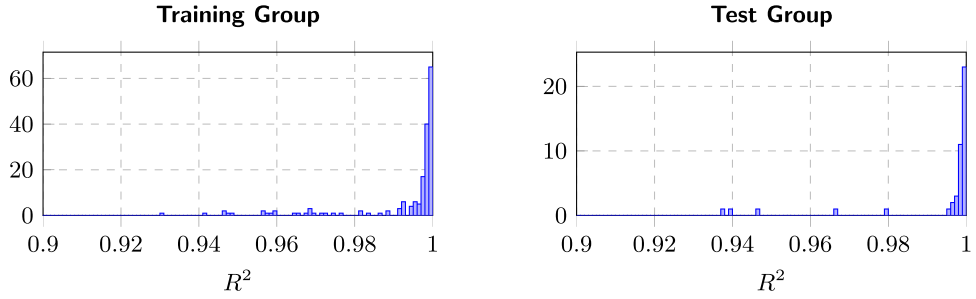

Fig. 7. FNN estimates *vs* targets of both training and test groups.

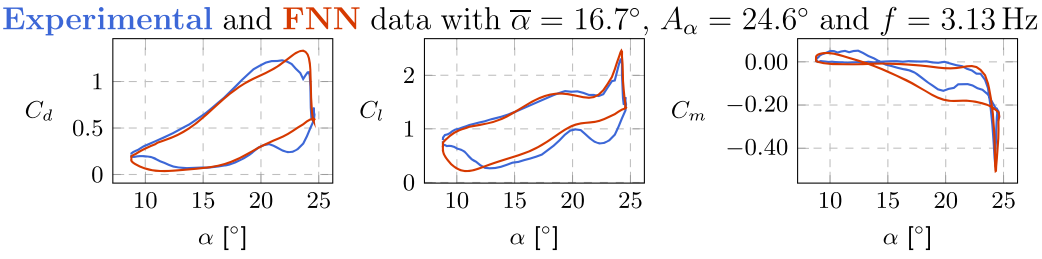

Fig. 8. Case 221 from the test pool with $R^2 = 0.9373$.

3.1.1. *Least-performing examples*

To properly assess the FNN's predictive performance, a group of graphs was selected to observe the aerodynamic coefficients as a function of the angle of attack. This section presents the top five least-performing cases (with the lowest R^2) to identify the FNN's most significant shortcomings.

Fig. 8 illustrates case 221, which has the lowest R^2 (= 0.9373) from the test group. Regarding drag, the FNN does provide a good approximation, but there is a visible overestimation of the maximum drag experienced at higher angles of attack. A similar effect is seen on the lift coefficient during the descending phase of the airfoil, which smooths out the oscillation seen between 20° and 25° and overestimates the lift generation during the whole descent. Nevertheless, the ascending phase is accurately represented with a great prediction of the peak lift. The pitching moment suffers from an identical problem, where it overestimates its magnitude for a vast portion of the period while interestingly capturing the intense nose-down peak moment. Nevertheless, while specific effects need to be addressed, the neural network provides an overall good approximation compared to the experimental measurements.

The second lowest R^2 (= 0.9392) is presented in Fig. 9 for case 196, exhibiting similar challenges as the case before. Drag is slightly overestimated, accompanied by a higher lift, mainly during the descending phase. While the moment coefficient indicated by the FNN resembles the infinity pattern, it fails to accurately predict the pitching moment magnitude.

In the following cases, 202 and 206, respectively, shown in Figs. 10 ($R^2 = 0.9463$) and 11 ($R^2 = 0.9668$), additional challenges in the aerodynamic prediction are seen. In contrast to the previous conditions, where, for instance, the lift peak was somewhat well predicted, here, some of the phases where abrupt changes occur are smoothed out, which indicates that, in some conditions, the FNN will be somewhat limited when predicting local or fast-occurring phenomena such as vortex shedding.

Fig. 12 ($R^2 = 0.9795$) depicts case 193, where the airfoil is at a pre-stall region. It is observed that the neural network performs quite well when compared to when it is subjected to higher angles of attack. It is essential to discuss these discrepancies from different regions of the angle of attack, which are likely to be influenced by the inherent variability and measurement noise that is typically present in experimental data. Indeed, and revisiting the previous cases, at lower angles of attack, the neural network performs very well, indicating that the FNN, as it is, is capable of predicting regions with smooth aerodynamic behavior, which suggests minimal data variability. However, when revisiting the post-stall regions, discrepancies quickly grow, implying that the network struggles in these areas either due to data sparsity, the complexity of the underlying physics, or, most likely, data variability due to the ran-

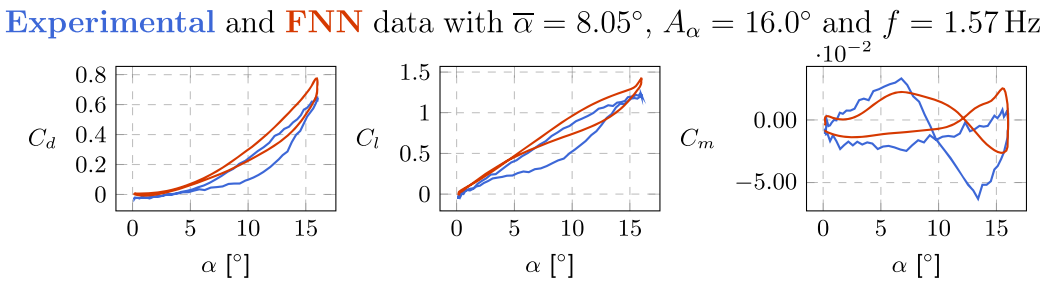

Fig. 9. Case 196 from the test pool with $R^2 = 0.9392$.

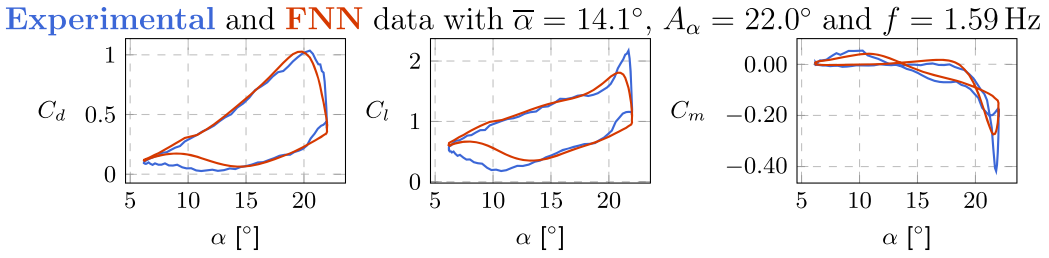

Fig. 10. Case 202 from the test pool with $R^2 = 0.9463$.

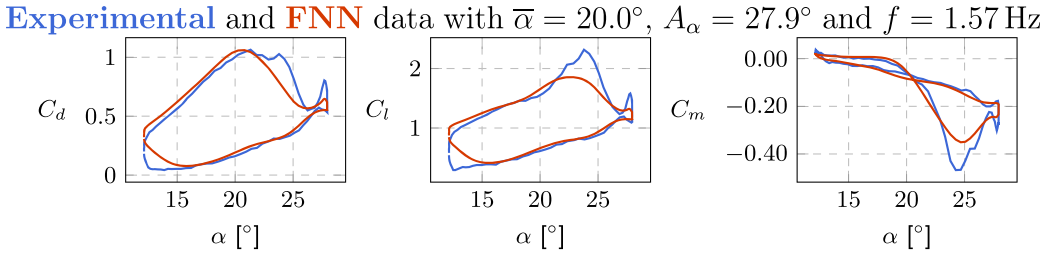

Fig. 11. Case 206 from the test pool with $R^2 = 0.9668$.

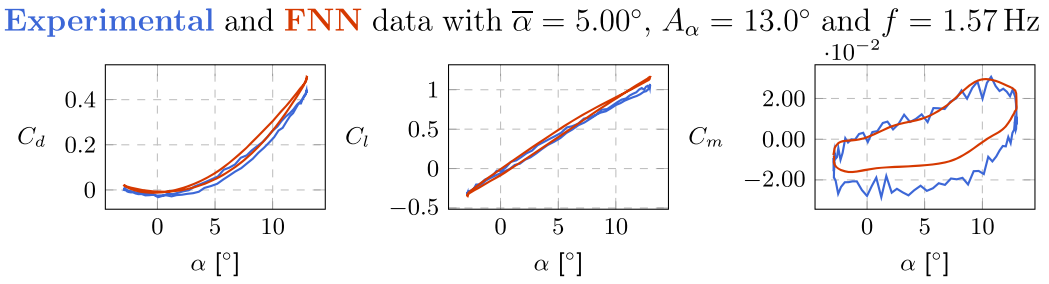

Fig. 12. Case 193 from the test pool with $R^2 = 0.9795$.

dom nature of dynamic stall coupled with turbulent flow. This condition comes as a consequence of employing a deterministic neural network, which struggles to truly capture variability in experimental data, as it will always tend to generalize by learning the mean trend rather than capturing measurement fluctuations.

While this subsection focused on the lowest-performance cases, one should not ignore the FNN's already good performance. The test group had all cases with an R^2 of at least 0.90.

3.1.2. *Best-performing and intermediate examples*

When looking at the five cases with the highest coefficient of determination, R^2, it is no surprise that, based on the observations made in the previous section, some of the best conditions are in the lower angle of attack region, where the attached flow is present.

In Figs. 13 through 15, one can observe that the predictions made by the neural network are very close to the measurements, indicating once again that this region is adequately modeled by the feedforward neural network, where data variability that may arise from highly nonlinear dynamics stall phenomena is not present.

Interestingly, the other two best-performing cases (namely, 16 and 145) touch high angles of attack regions with highly nonlinear effects (*cf.* Figs. 16 and 17). As observed, in such conditions, the neural network can anticipate all aerodynamic coefficients, capturing the drag's intricacies, the rapid increase in lift after 20° followed by the sharp decline typically associated with the vortex shedding process. One can also see the outstanding FNN capability in the moment coefficient of these particular cases, comprehending the complex aerodynamic pattern and accurately predicting the nose-down peak value.

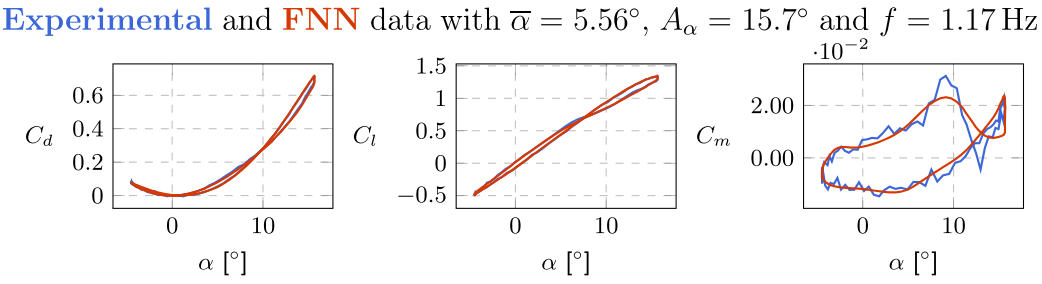

Fig. 13. Case 44 from the test pool with $R^2 = 0.9996$.

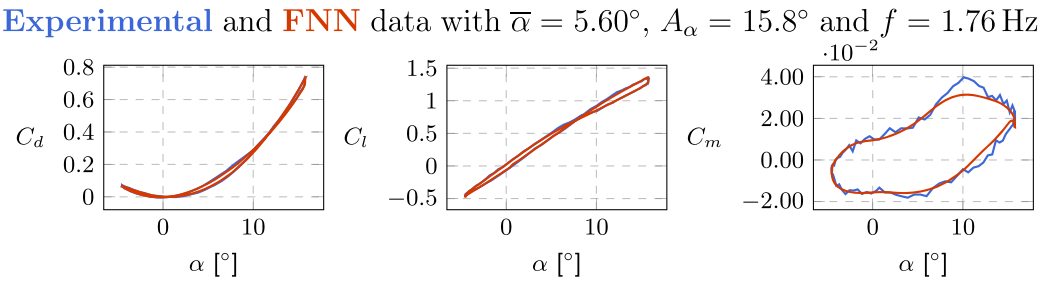

Fig. 14. Case 49 from the test pool with $R^2 = 0.9996$.

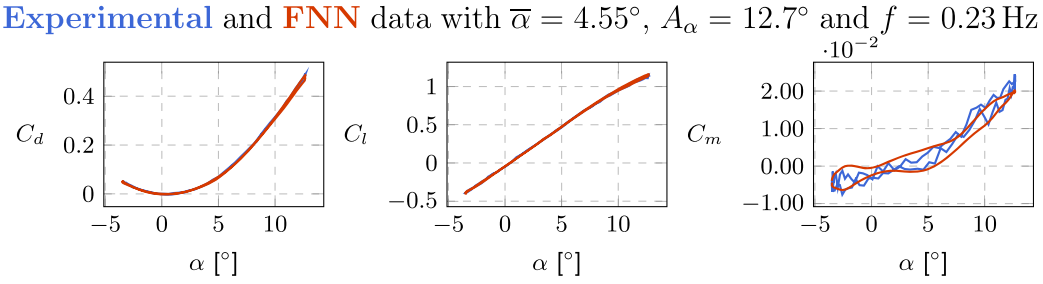

Fig. 15. Case 75 from the test pool with $R^2 = 0.9996$.

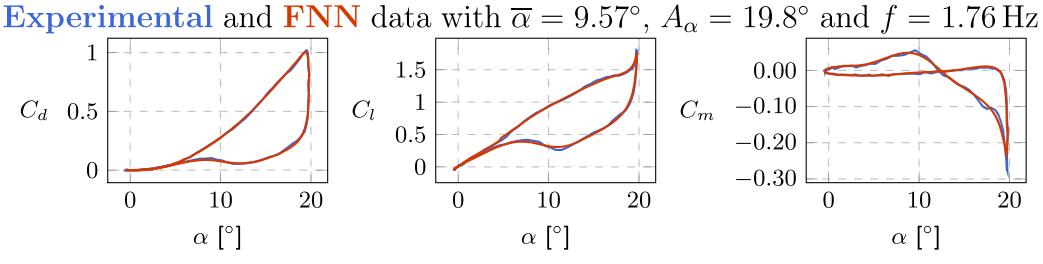

Fig. 16. Case 16 from the test pool with $R^2 = 0.9996$.

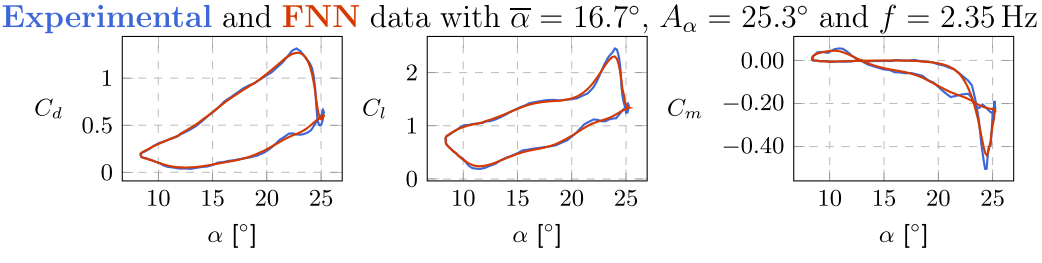

Fig. 17. Case 145 from the test pool with $R^2 = 0.9997$.

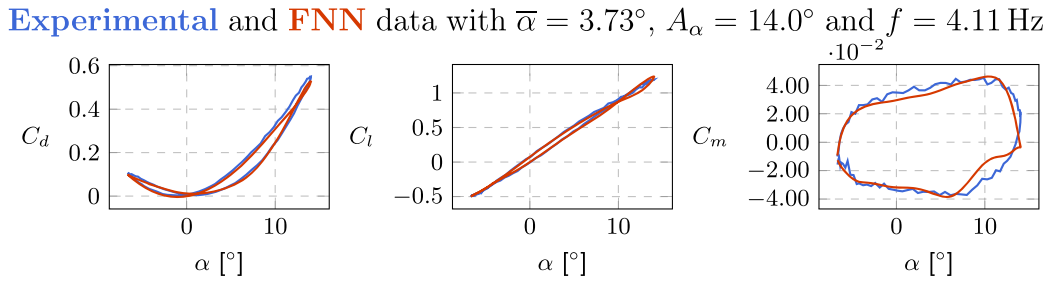

Fig. 18. Case 68 from the test pool with $R^2 = 0.9956$.

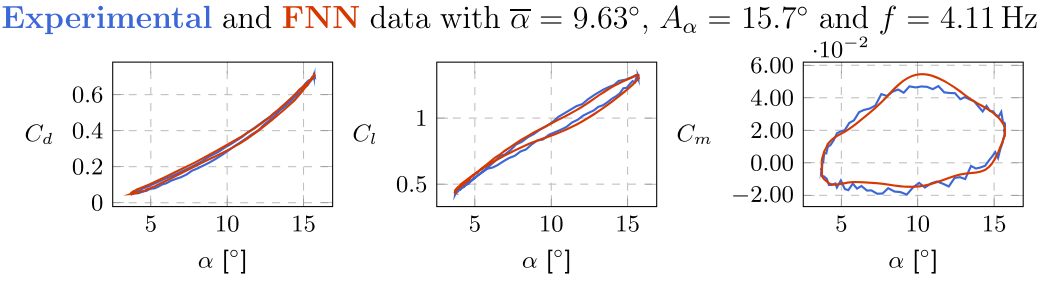

Fig. 19. Case 30 from the test pool with $R^2 = 0.9984$.

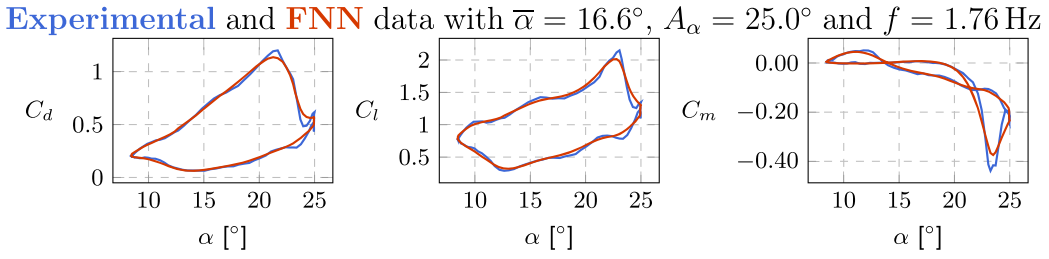

Fig. 20. Case 130 from the test pool with $R^2 = 0.9990$.

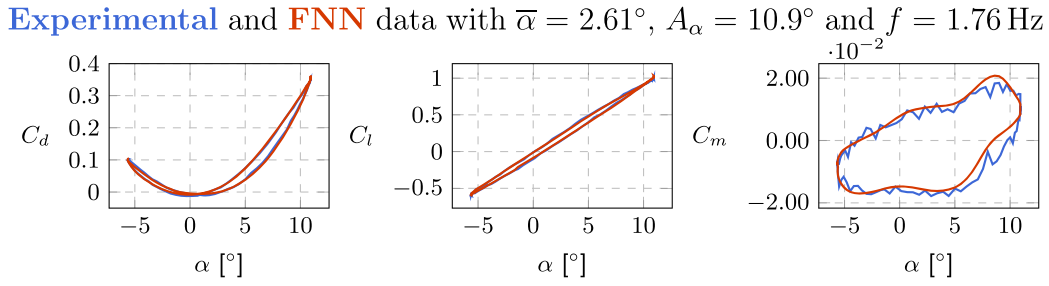

Fig. 21. Case 118 from the test pool with $R^2 = 0.9992$.

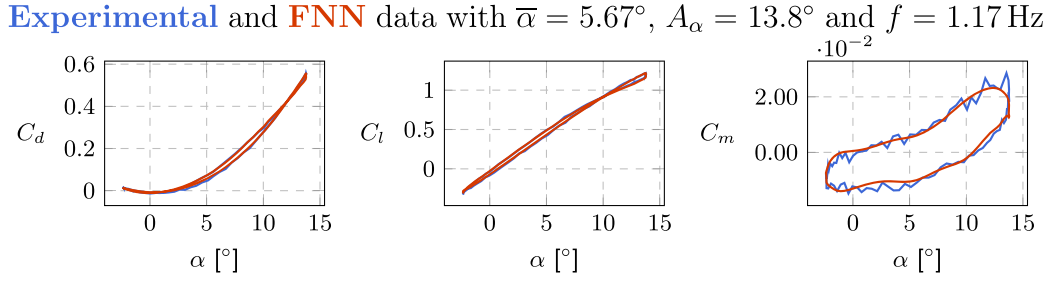

Fig. 22. Case 106 from the test pool with $R^2 = 0.9995$.

In addition to the best-performing prediction cases, five intermediate cases were evenly selected from the test group for a broader observation of the predictive capabilities of the FNN, shown in Figs. 18 to 22.

Results indicate that the FNN provides a straightforward way to quickly predict the aerodynamics of airfoils subjected to periodic waveforms. The neural network model captures the overall tendency of the

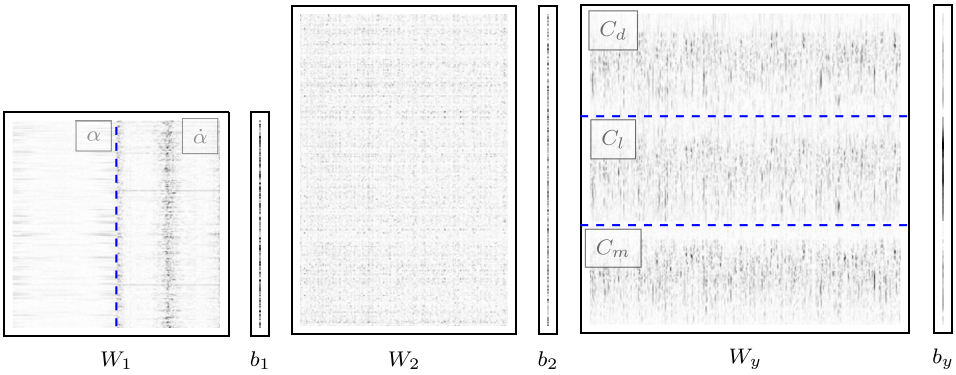

Fig. 23. FNN weights and biases.

aerodynamic coefficients during the oscillation cycle. Nonetheless, as seen before, in some instances, the aerodynamic coefficients can be slightly over- or under-estimated, significantly farther from the linear lift region.

3.2. The FNN model features

For the final phase of the analysis, an internal view of the neural network is provided to discuss its architecture. Fig. 23 shows the insides of the neural network, where the weights and biases arrays are normalized by their respective maximum absolute value. Blacker elements correspond to higher weight and bias values.

Looking at the first layer (W_1 and b_1), which is directly connected to the inputs, one can see that the weights are divided into two major vertical regions that correspond to the elements multiplied by the α and $\dot{\alpha}$ vectors. When comparing these two regions, the $\dot{\alpha}$ band is much more activated, with higher values located at the columns that correspond to the $t/T = 0.0$, 0.5, and 1.0, corresponding to when the airfoil changes the pitching direction and has maximum acceleration.

The slight over-dependence on the pitching velocity is no surprise since $\dot{\alpha}$ significantly influences unsteady aerodynamic effects such as vortex formation and shedding. These flow effects that introduce hysteresis in the aerodynamic coefficient are known to depend not just on the instantaneous angle of attack, α, but on how quickly it changes.

The second (hidden) layer, which acts as an intermediate feature extractor that combines the relationships between the inputs and the aerodynamic coefficients, appears to be highly entropic without clear patterns. This could be due to the complexity of the dynamic stall phenomenon, where many nonlinear effects and interactions exist, or the FNN's adaptation to capture subtle details in the data.

Identical to the first layer, the third layer is divided into regions, specifically three distinct horizontal bands, which correspond to the drag, lift, and pitching moment coefficients. This output segmentation indicates that the FNN is learning specialized features for each aerodynamic coefficient. Nevertheless, the distinct areas are not synonyms of complete independence, as the second layer can act as a mixer, mainly because the three aerodynamic coefficients are aerodynamically connected.

A closer look at the individual bands also indicates that the third layer is more activated closer to the middle of these regions, corresponding to the time frame around $t/T = 0.50$. Based on the training data and its structure, this is when the airfoil experiences high angles of attack, leading to the occurrence of the dynamic stall, which affects all coefficients. Accordingly, this last layer ends up highlighting highly unsteady separation effects which drastically increase drag, intensely affect C_l due to vortex-induced lift peaks and hysteresis, and cause considerable C_m shifts due to irregular and time-varying pressure distribution around the airfoil.

With this final analysis, one can get a simplistic yet powerful view of the FNN's insides, verify the relative importance between the inputs, and understand, from a superficial perspective, how the flow of information travels through the FNN. While the use of a feedforward neural network that predicts the whole aerodynamic cycle instantly offers a great way for preliminary design, especially when analyzing periodic regimes, for instance rotor aerodynamics, one should not neglect the limitations of such an approach. The current architecture has as its inputs α and $\dot{\alpha}$, meaning that no information regarding flow conditions is given. However, as is commonly known, the Reynolds number influences the evolution of dynamic stall phenomena. Hence, as a recommendation for a future update, one could incorporate a new input vector, for instance $\dot{\alpha}c/U$, that, through the flow speed, introduces the effect of the Reynolds number. An additional limitation is that the aerodynamic predictions are based solely on kinematics, which can be restrictive under certain conditions. Prediction accuracy could be enhanced by incorporating kinematic data and the historical aerodynamic behavior, which is a priority for future research. Moreover, the present architecture will provide phase-locked results without the possibility of real-time updates of aerodynamic coefficients, as explained in subsection 2.2. Nevertheless, as observed in the results, the FNN does capture temporal effects with its large network of connections from each time frame.

4. Conclusions

Studying dynamic stall phenomena is crucial to expediting airfoil design, identifying unwanted aerodynamic regimes, and, at a practical level, improving operational safety. While the root causes of the dynamic stall are well understood, there is still no reliable and inexpensive method to predict the aerodynamic loads. This implies that, for instance, the hidden patterns in experimental data are not being uncovered and that known limitations in numerical methodology are still being perpetuated.

Artificial intelligence techniques can help unlock the uncovered connections, offering a straightforward way to quickly understand the particular problem of oscillating airfoils without extensive and complex modeling. The present work has this objective: to show that the dynamic stall can be estimated even with a straightforward neural network architecture. A feed-forward network is selected to achieve such an objective, using the kinematics, including the angle of attack and pitch rate, as inputs. Since this is not a neural network acting in time, all cycle information is given at once, meaning that the complete cycle aerodynamics that includes drag, lift, and pitching moment are obtained simultaneously. The training phase was performed using experimental dynamic stall data from a known database from the University of Glasgow, containing many cases ranging from light to deep stalls with various pitching amplitudes and frequencies. With a supervised learning approach, using the gradient descent coupled with the Adam algorithm, the neural network achieved a $R^2 = 0.99$ for both testing and test datasets. Results are clear, showing that the neural network is a reliable tool to estimate the aerodynamic cycle even in the least-performing cases. The

good performance metrics indicate that the FNN understands the underlying dynamics of time-dependent stalls even with diverse conditions. Indeed, no case from either dataset experienced a R^2 less than 0.90. The final phase of the results shows a brief analysis of the weights where one can look at the insides of the neural network, which revealed a more considerable activation in the pitch rate compared to the angle of attack and a strong segmentation in the last hidden layer, indicating that specialized features for each aerodynamic coefficient are being learned.

Taking advantage of artificial intelligence to study unsteady aerodynamics should be fomented, where a simple neural network can capture the fundamentals of the dynamic stall with the ability to be trained with new data as obtained. Future research should be focused on the transient essence of the problem by exploring neural networks that act in time, such as recurrent or physics-informed neural networks, as well as exploring different kinematics for a more versatile use.

CRediT authorship contribution statement

Emanuel A.R. Camacho: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **André R.R. Silva:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization. **Flávio D. Marques:** Writing – review & editing, Validation, Supervision, Resources, Investigation, Funding acquisition, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors acknowledge the financial support from the following Brazilian Agencies: the Brazilian National Council for Scientific and Technological Development – CNPq (grant #306698/2023-4), the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through the project numbers UIDB/50022/2020, UIDP/50022/2020, LA/P/0079/2020.

Data availability

Repository will be available.

References

[1] W. McCroskey, L. Carr, K. McAlister, Dynamic stall experiments on oscillating airfoils, AIAA J. 14 (1) (1976) 57–63, https://doi.org/10.2514/3.61332.
[2] K. Mulleners, M. Raffel, Dynamic stall development, Exp. Fluids 54 (2) (Feb. 2013), https://doi.org/10.1007/s00348-013-1469-7.
[3] G. Bangga, T. Lutz, M. Arnold, An improved second-order dynamic stall model for wind turbine airfoils, Wind Energy Sci. 5 (3) (2020) 1037–1058, https://doi.org/10.5194/wes-5-1037-2020.
[4] G. Baldan, A. Guardone, A deep neural network reduced order model for unsteady aerodynamics of pitching airfoils, Aerosp. Sci. Technol. 152 (2024) 109345, https://doi.org/10.1016/j.ast.2024.109345.
[5] R.F. Miotto, W. Wolf, Prediction of airfoil dynamic stall response using convolutional neural networks, in: Proceedings of the AIAA AVIATION 2023 Forum, American Institute of Aeronautics and Astronautics, 2023.
[6] B. Zhang, Airfoil-based convolutional autoencoder and long short-term memory neural network for predicting coherent structures evolution around an airfoil, Comput. Fluids 258 (2023) 105883, https://doi.org/10.1016/j.compfluid.2023.105883.
[7] E. Saetta, R. Tognaccini, G. Iaccarino, Machine learning to predict aerodynamic stall, Int. J. Comput. Fluid Dyn. 36 (7) (2022) 641–654, https://doi.org/10.1080/10618562.2023.2171021.
[8] B. Mi, W. Cheng, Intelligent aerodynamic modelling method for steady/unsteady flow fields of airfoils driven by flow field images based on modified u-net neural network, Eng. Appl. Comput. Fluid Mech. 19 (1) (Dec. 2024), https://doi.org/10.1080/19942060.2024.2440075.
[9] L. Damiola, J. Decuyper, M. Runacres, T. De Troyer, Modeling airfoil dynamic stall using state-space neural networks, in: Proceedings of the AIAA SCITECH 2023 Forum, American Institute of Aeronautics and Astronautics, 2023.
[10] S. Kasmaiee, M. Tadjfar, S. Kasmaiee, Machine learning-based optimization of a pitching airfoil performance in dynamic stall conditions using a suction controller, Phys. Fluids 35 (9) (Sep. 2023), https://doi.org/10.1063/5.0164437.
[11] J. Liu, R. Chen, J. Lou, Y. Hu, Y. You, Deep-learning-based aerodynamic shape optimization of rotor airfoils to suppress dynamic stall, Aerosp. Sci. Technol. 133 (2023) 108089, https://doi.org/10.1016/j.ast.2022.108089.
[12] Z. Shi, C. Gao, Z. Dou, W. Zhang, Dynamic stall modeling of wind turbine blade sections based on a data-knowledge fusion method, Energy 305 (2024) 132234, https://doi.org/10.1016/j.energy.2024.132234.
[13] G. Baldan, A. Guardone, A deep neural network physics-based reduced order model for dynamic stall, https://doi.org/10.48550/ARXIV.2401.14728, 2024.
[14] J.-P. Küppers, T. Reinicke, A wavenet-based fully stochastic dynamic stall model, Wind Energy Sci. 7 (5) (2022) 1889–1903, https://doi.org/10.5194/wes-7-1889-2022.
[15] X. Wang, J. Kou, W. Zhang, A new dynamic stall prediction framework based on symbiosis of experimental and simulation data, Phys. Fluids 33 (12) (Dec. 2021), https://doi.org/10.1063/5.0075083.
[16] L. Damiola, M.C. Runacres, T. De Troyer, The challenge of cycle-to-cycle variability in dynamic stall modelling, J. Phys. Conf. Ser. 2767 (5) (2024) 052007, https://doi.org/10.1088/1742-6596/2767/5/052007.
[17] Y. Dai, H. Rong, Y. Wu, C. Yang, Y. Xu, Stall flutter prediction based on multi-layer GRU neural network, Chin. J. Aeronaut. 36 (1) (2023) 75–90, https://doi.org/10.1016/j.cja.2022.07.011.
[18] A. Torregrosa, L. García-Cuevas, P. Quintero, A. Cremades, On the application of artificial neural network for the development of a nonlinear aeroelastic model, Aerosp. Sci. Technol. 115 (2021) 106845, https://doi.org/10.1016/j.ast.2021.106845.
[19] B. Zhang, J. Han, T. Zhang, R. Ma, Unsteady aerodynamic identification based on recurrent neural networks, J. Vibroeng. 23 (2) (2020) 449–458, https://doi.org/10.21595/jve.2020.21612.
[20] X. Wang, J. Kou, W. Zhang, Z. Liu, Incorporating physical models for dynamic stall prediction based on machine learning, AIAA J. 60 (7) (2022) 4428–4439, https://doi.org/10.2514/1.j061210.
[21] A. Shelton, J. Abras, R. Jurenko, M. Smith, Improving the CFD predictions of airfoils in stall, in: Proceedings of the 43rd AIAA Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 2005.
[22] R.B. Green, M. Giuni, Dynamic stall database R and D 1570-AM-01: final report, https://doi.org/10.5525/GLA.RESEARCHDATA.464, 2017.
[23] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, https://doi.org/10.48550/ARXIV.1412.6980, 2014.
[24] E.A.R. Camacho, F.D. Marques, A.R.R. Silva, neuroStall, https://github.com/Emanuel96/neuroStall.git, Jan. 2025.