IMPERIAL COLLEGE LONDON

Department of Mechanical Engineering
Advanced Mechanical Engineering

Machine Learning Modelling of Stall on Airfoils

By
Berat Kaan Fırat

Bachelor of Science, Mechanical Engineering, Middle East Technical University

A thesis submitted in partial fulfilment of the requirements for the Master of Science Degree of Imperial College London and the Diploma of Membership of Imperial College.

September 2022

Word Limit: 18000

Word Count: 10990

Abstract

Research and developments have been growing for wind power industry in the form of onshore and offshore wind turbines as they provide cost-effective, clean, and sustainable energy. The increase in the need of clean and sustainable energy source and competition in renewable energy has created a need of a more efficient design for wind turbines that is capable of withstanding high loads while avoiding stability problems of the blades like stall.

CFD analysis require many simulations and are mostly time consuming when the design space of possible airfoil geometries is considered. On the other hand, the recent growth of data driven methods and applications has led to advances in many scientific fields. Therefore, machine learning applications can be used together with a data set from a high-fidelity data from CFD to build a surrogate model. However, more conventional methods require large data set to make the machine learning model learn the pattern and usually struggles to foresee the results for the unseen data.

This project aims to combine the methods on CFD simulations with machine learning applications to create a predictive model that determines the stall of an airfoil by using novel machine learning techniques. In this technique, physically lower-fidelity solutions are being injected into the machine learning model to make the most efficient output from the model predictions with a given amount of data.

To provide the data set of different airfoil geometries covering corresponding stall angles, a CFD solver has been utilised. Artificial neural networks were trained, and physics guidance is provided by a vortex panel method.

It has been shown that when the neural networks are constrained by physics input, data is used more efficiently, and predictions are more accurate to the physical situations in addition to the fact that the models identify the pattern using data, making the concept of finding the stall quicker which would otherwise take quite long times when solely CFD simulations were performed. Therefore, it is more reliable to depend on the predictions from physics guided neural networks which provided stall predictions up to 5% error whereas pure neural networks model provides 15% error when both models have the same structure.

Acknowledgements

I would like to express my gratitude to my supervisor Prof Mehdi Vahdati and my co-supervisor Dr Fanzhou Zhao and Marco Rauseo for the invaluable support and guidance throughout this project.

I would like to thank the energy company Equinor which financed this project and my studies at Imperial College.

I would like to thank Dr Ulrich Hansen and Prof Alex Taylor for their effort on running AME course and being supportive and guiding for all AME students this year.

Finally, I would like to thank my family and friends who kept their endless support and kept me motivated throughout my time doing this Master's.

Contents

Nomenclature

CFD	:	Computational Fluid Dynamics
DNS	:	Direct Numerical Solution
LES	:	Large Eddy Simulations
RANS	:	Reynolds Averaged Navier Stokes
Re	:	Reynolds number
Ma	:	Mach number
C_P	:	Pressure coefficient
δ_{ij}	:	Kronecker delta $\delta_{ij} = 1$ for $i = j$, else 0
SU2	:	CFD code used in this project, Stanford University Unstructured [1]
SA	:	Spalart-Allmaras turbulence model
SST	:	Menter Shear Stress Transport turbulence model
ANN	:	Artificial neural networks
CNN	:	Convolutional neural networks
ML	:	corresponds to pure machine learning model without physics guidance
PGML	:	corresponds to ğhysics guided machine learning

1. Introduction

1.1. Motivation

Determination of airflow characteristics of airfoils is a significant problem in aerospace industry and energy sector, specifically wind turbine applications. As being the most important part of the energy generation system in wind turbines, blades require significant consideration starting from initial design of blades [2]. Also, since turbine blades are required to withstand excessive forces, which in turn generates moment around the hub and generates electricity, it is the ultimate reason why the design of blades is crucial and so is selection of the shape of the blades to make sure that the blades of turbines stay away from the stall region and withstand the wind force. Therefore, there has always been a need to investigate designed shape of airfoils either experimentally or computationally.

Experimental methods require a physical set-up like wind tunnel, there is a need of manufacturing of models, and the experiments take a long time for many airfoils. On the other hand, with the computational methods, larger number of design options can be covered quickly.

The design and analysis of the blades are based on understanding the airfoil characteristics. The most basic characteristics for the airfoil are lift and drag coefficients, and to respond the need of withstanding high forces, it is required to be known in how large interval an airfoil can produce lift and where the stall occurs. This makes the lift coefficient and stall of airfoils important design consideration.

Therefore, in the industry, the computational methods are needed to propose a design before seeking its testing and experiments. The motivation of this project is to propose a computational model for the mentioned need of finding the stall of airfoils.

1.2. Physical considerations of airfoil flows

When an airfoil is under uniform flow, the flow is perturbated by the existence of airfoil and the flow stagnates at the leading edge where a high pressure is observed. After the stagnation point, flow accelerates and follows the shape of the airfoil because of continuity.

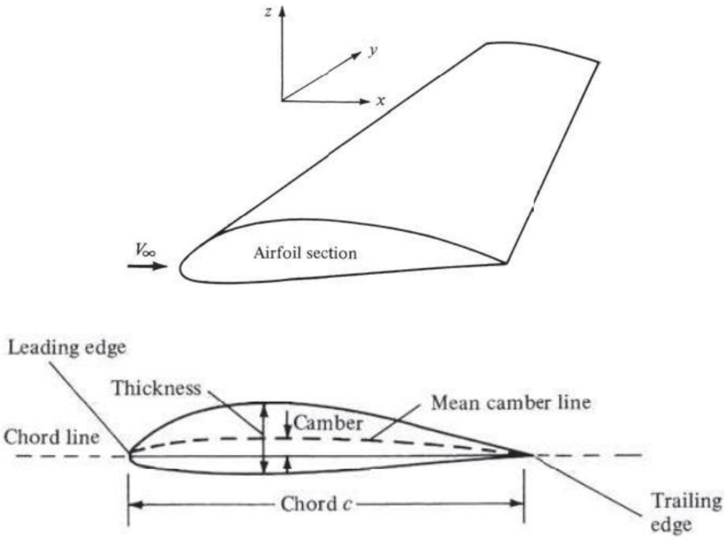

Figure 1: Airfoil section and airfoil nomenclature [3]

As the angle of inclination of airfoil increases or in the cambered airfoils with zero inclination (angle of attack), the flow travels across the top of the airfoil faster than the air moving along the bottom surface. Therefore, lower pressure exists on the top surface and higher pressure on the bottom surface which creates lift. This pressure difference can be explained by the pressure coefficient over the airfoil body.

$$C_P = \frac{P - P_\infty}{\frac{1}{2}\rho_\infty V_\infty^2} \tag{1}$$

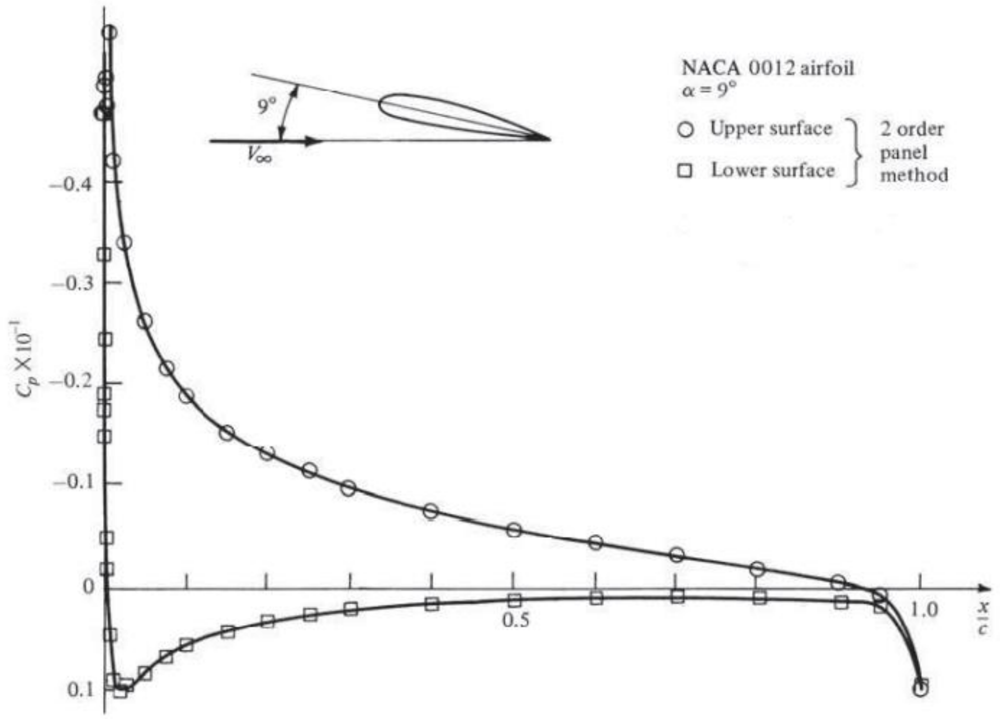

Figure 2: Pressure distribution over NACA0012 airfoil [3]

One can see the pressure coefficient distribution around the airfoil in Figure 3. The difference between two curves of each plot is the pressure that creates the lift. Therefore, the area between the two curves represents the lift generated due to the flow.

Airflow follows the airfoil for up to certain inclination (angle of attack) of airfoil after which separation occurs. The separated flow depends on the airfoil geometry as well as the airfoil angle of attack.

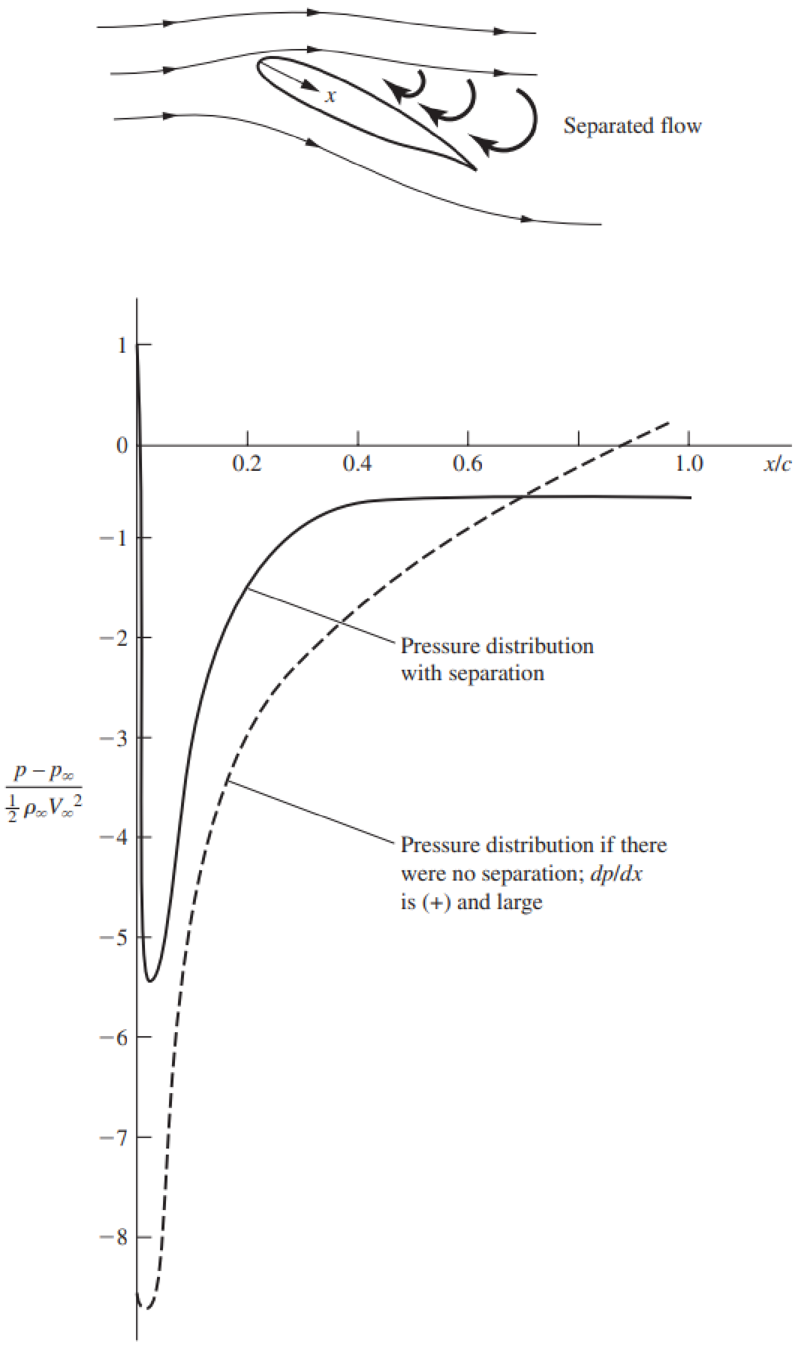

Figure 3: Pressure distribution over the top surface of an airfoil with and without separation [3]

When airfoil stalls, the pressure coefficient on the top surface of the airfoil increases as can be seen in Figure 3 because of the separation caused by the adverse pressure gradient.

Another way of investigating an airfoil is by checking lift and drag curves. Both left and right figures in Figure 4 show the lift against angle of attack curve for airfoils. Stall is the angle for an airfoil when the maximum lift coefficient is obtained at a certain angle of attack beyond which the lift, hence the lift coefficient, decreases and drag increases significantly. As it can be seen from Figure 4, as the angle of attack increases (for relatively low angles), flow follows the airfoil profile and lift continues to increase until where the maximum lift occurs and after which a decrease in lift and an increase in drag becomes realised [3]. This is a general variation of the coefficient of lift with angle of attack, and the determination of stall point, that is just after the angle where maximum lift coefficient occurs, is the quantity of interest in this project.

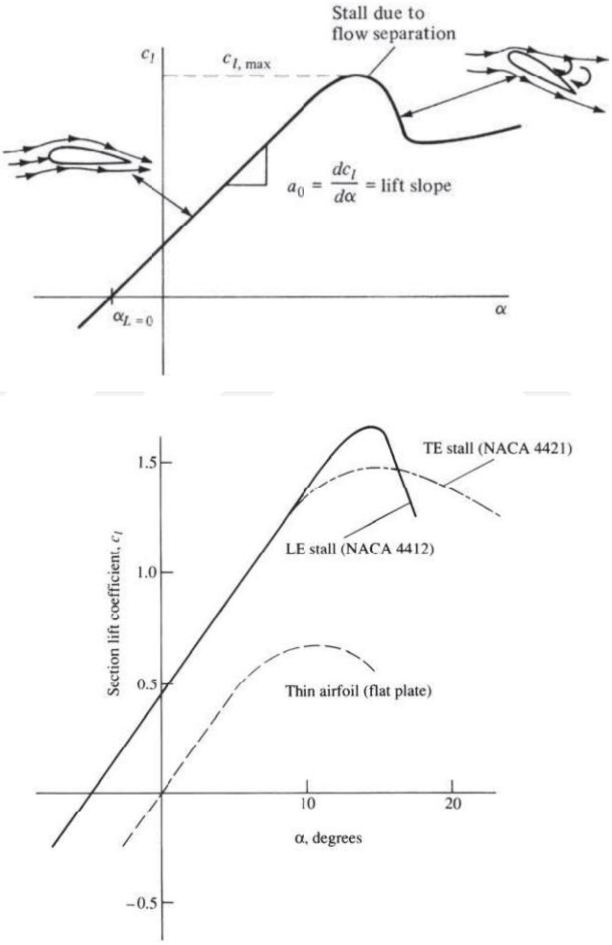

Figure 4: (Left) Lift coefficient variation with changing angle of attack for an airfoil. (Right) Lift coefficient curves for three different airfoils with different aerodynamic behaviour [3]

The plot on the right of Figure 4 shows the three main behaviours in variation of lift as well as type of the stall on airfoils. Very thin airfoils experience a lift curve like flat plates where the stall angle is earlier than other airfoils. Considering the flat plate is symmetric shaped object, its lift curve passes through zero lift at zero angle of attack. Moreover, flow separation and the turbulence start at the leading edge of the airfoil which makes the stalling more sudden and closer to leading edge stall. Leading edge stall is the type of stall occurring for thin airfoils. As the angle of attack increases, because the airfoil is thin, the geometry blocks the airfoil in a way that the blunt object does. Therefore, the separation of the flow starts at leading edge. On the other hand, trailing edge stall occurs when the airfoil is thick and at higher angles of attack flow can still follow the upper surface of the airfoil and as a result, flow starts to separate from the trailing edge resulting in slower decay in the lift coefficient in terms of angle of attack.

1.2.1. Potential Flows

In potential flow theory, velocity fields are tried to be explained with gradient of a scalar function called circulation. Potential flows are first developed to understand flows around the cylinders, but they are not good at explaining flows around the airfoil geometries unless a mathematical transformation.

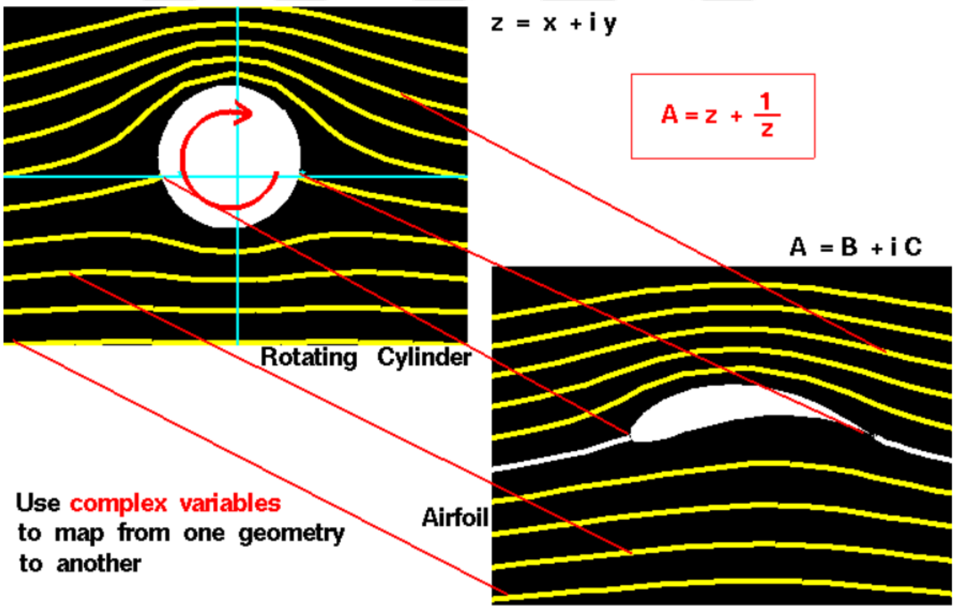

Figure 5: Juokowski Transformation Map [4]

With the mathematical transformation using mapping function given in Figure 5 by Juokowski, cylinder is converted into set of aerofoil shapes. Conversion is made from one complex plane to another. Creation of lift is explained by the following lift equation.

$$L = \rho_\infty V_\infty \Gamma \quad (2)$$

From the equation there could be infinite number of lifts for the airfoil, unique lift value is set by imposing Kutta condition. Kutta condition states that there should be sufficient circulation at specific angle of attack to ensure that the flow stagnation at the leading and trailing edge of the airfoil occurs as in the realistic flows. According to that, flow must leave the airfoil at trailing edge smoothly.

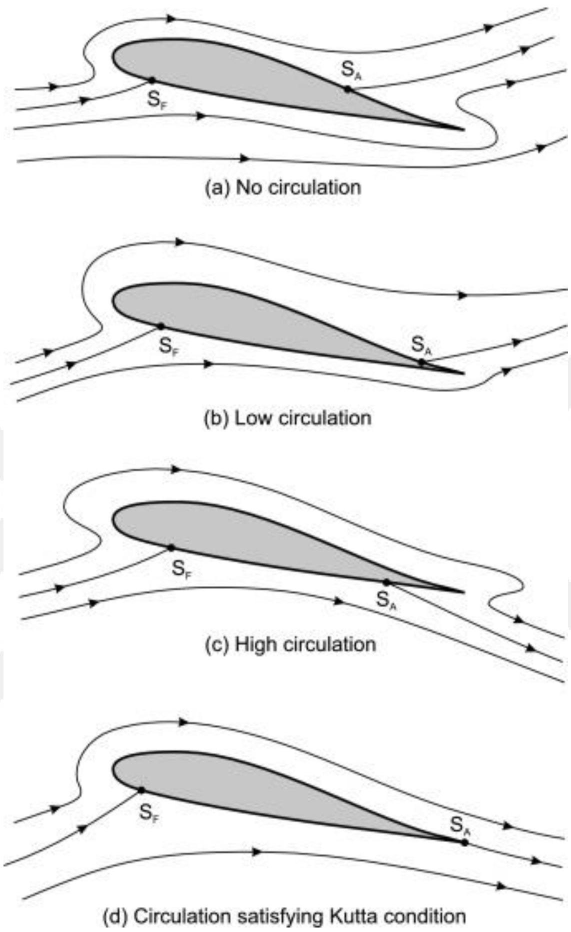

Figure 6: Kutta condition on airfoil [5]

Figure 6 shows the illustration of different circulations where the Kutta condition only satisfies in (d). For (a), there is no circulation. For (b), there is circulation, but not enough to make the flow leave the airfoil at trailing edge making the rear stagnation point at top surface. For (c), there is too much circulation which makes the stagnation point at bottom surface. All three flows are unrealistic and create three different lift values which makes it non-unique to determine the actual lift [5].

1.2.2. Vortex Panel Method

Vortex panel method is a numerical method that approximates the airfoil by set of panels which enclose the region of a 2D airfoil as closely as possible to its real area. While the number of panels to capture the shape can be modified, each panel works as vortex source with unknown circulations that ultimately must agree the Kutta condition and represent the flow around the airfoil. Since the vortex strengths are unknown, the main purpose of this solution is to solve the panel strengths such that in the end, Kutta condition is satisfied. [3]

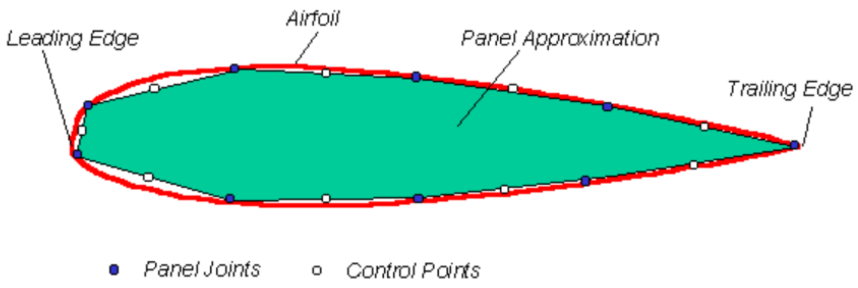

Figure 7: Vortex panel method approximation of an airfoil [6]

Figure 7 shows the representation of vortex panels attached to each other and complete the geometry of airfoil. In the panel method, vortex strength at each panel is tried to be solved by iterations. After that, summation of all the vortices yield the total circulation of airfoil which can reveal the lift using lift equation [7].

1.2.3. CFD Simulations

Continuity and Navier Stokes equations that when solved give the flow properties are given as follows.

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{3}$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] \tag{4}$$

The Navier-Stokes equations given by continuity and momentum equations are simplified in a statistical approach by Reynolds such that the instantaneous velocity and pressure are decomposed into mean and fluctuating parts. This way the averaging is done and resulting equations are called Reynolds Averaged Navier-Stokes equations.

$$u_i = U_i + u_i' \tag{5}$$

$$\frac{\partial U_i}{\partial x_j} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\nu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \overline{u_i' u_j'}\right] \tag{6}$$

This decomposition creates velocity covariance term $\overline{u_i'u_j'}$ called Reynolds stresses on which closure models are developed. Closure models include the modelling of Reynolds stresses where turbulence and flow properties are introduced in addition to fluid properties. Different modelling approaches have been developed after following approximation from Boussinesq with which the Reynolds stress term from the Navier-Stokes is eliminated.

$$-\overline{u_i'u_j'} = \nu_t\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}k \tag{7}$$

where ν_t is turbulent viscosity and $k = \overline{u_i'u_i'}/2$ is turbulent kinetic energy.

Closure models have been built by using some empirical relations on turbulent properties and are still being modified today to resolve the flows better. Therefore, there are many turbulence models available for the use in a particular problem.

In the literature, there are CFD studies on standard airfoils like NACA0012 and S809. Below is an example of a study where both airfoils were simulated with different turbulence models and compared with each other.

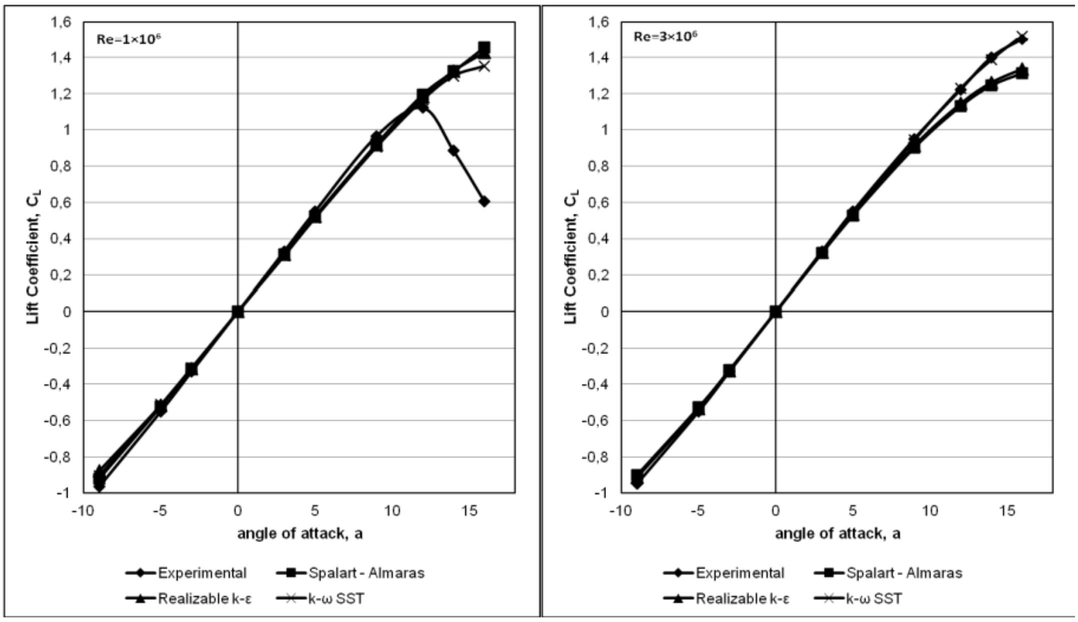

Figure 8: NACA0012 airfoil CFD results with different models and comparison with experimental data [8]

Duovi & Margaris completed different turbulence models with NACA0012 airfoil and made a comparison with the experimental data [8]. It has been decided that SST turbulence modelling is closer to experimental data and comparison of two airfoils (NACA0012 and S809) have been made with this modelling.

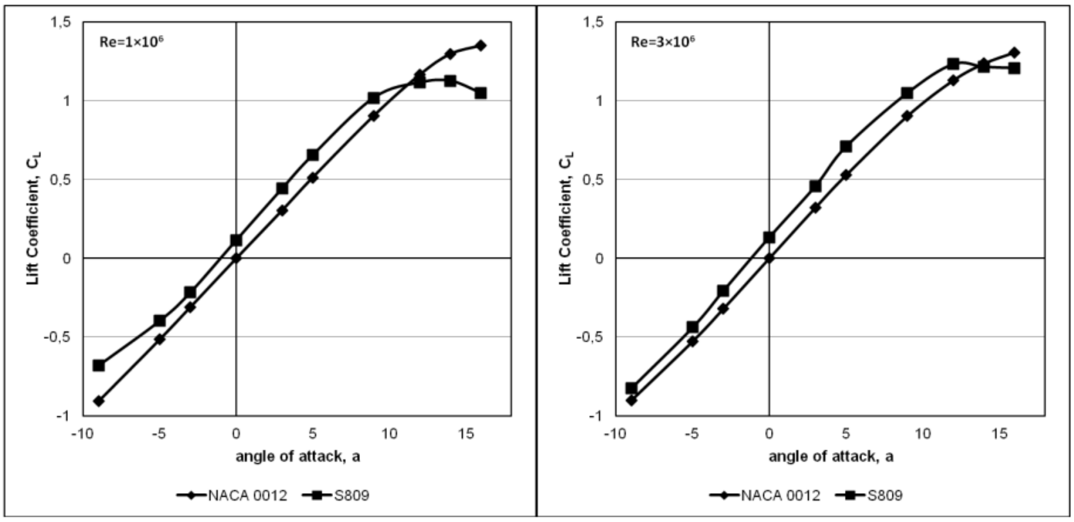

Figure 9: NACA0012 and S809 airfoils comparison with SST turbulence model [8]

After the decision of SST turbulence modeling, comparisons of two airfoils with different Reynolds numbers have been made and it has been concluded that S809 airfoil has greater lift coefficient values for the operating angles of attack lower than stall although closer to stall NACA0012 has greater lift coefficient [8].

1.3. Machine learning modelling considerations

Machine learning applications have become very popular in many areas of research and with the results being valuable and promising, data driven models are gaining more importance in scientific fields.

There are three main types of machine learning applications in general: supervised, semi-supervised and unsupervised. There are different models, for the type of problem that is dealt with. When a supervised model is considered, all the data set components have their labels and if there are many input variables, neural network models can work well with this type of problems.

Good thing about the neural network is that all the introduced parameters are nondimensionalised, so the maths to estimate network parameters (weightings of neurons and biases) can easily be performed using scaled parameters.

1.3.1. Artificial Neural Networks

An example of neural networks structure is shown in Figure 10. There are activation functions at each hidden layers which results in a value coming from weighted sum of inputs from previous layers' neurons which simply is the result from the hypothesis function, $h(x)$.

Therefore, this function performs the weighted sum in a way that at all the connections, values are multiplied with the corresponding weights.

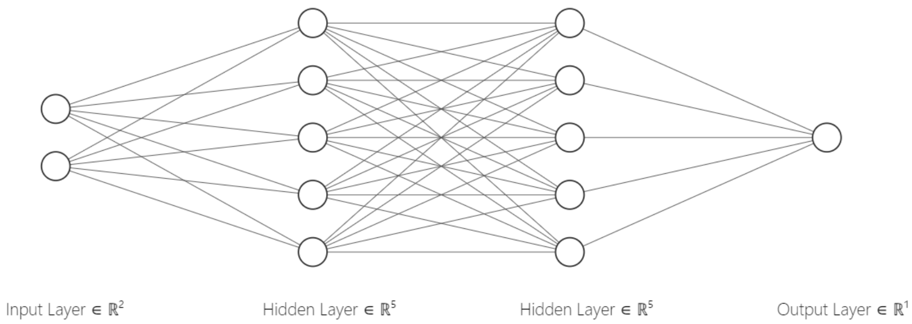

Figure 10: A simple neural network structure

Before the network training and learning, the data is trained by subtracting the mean and dividing by the standard deviation for each parameter which is one of the most common methods.

$$d_{sc} = \frac{d - \bar{d}}{\sigma_d} \tag{8}$$

Linking arrows in the network structure represent the weights. Inputs are multiplied by these arrows and summed at every hidden node resulting the weighted sum (or activation value).

Weighted summation is calculated at every connection [9].

$$net_j = \sum_{i=0}^{n} w_{ji}^{1} x_i + b_j \tag{9}$$

Arriving at each node, the summation is activated using the activation function of the node. Activation functions can be considered like transfer functions. This process results in following value at each node of one layer to be seeded into next layer.

$$y_j = f\left(net_j\right) \tag{10}$$

In practice, there are many mathematical activation functions, and they can be changed at each layer of the structure, so the resulting values are propagated in the forward direction to get the final value of the network, z (z if the final activation function is just linear). In other words, this is the nested summation and activation of every layer and every node.

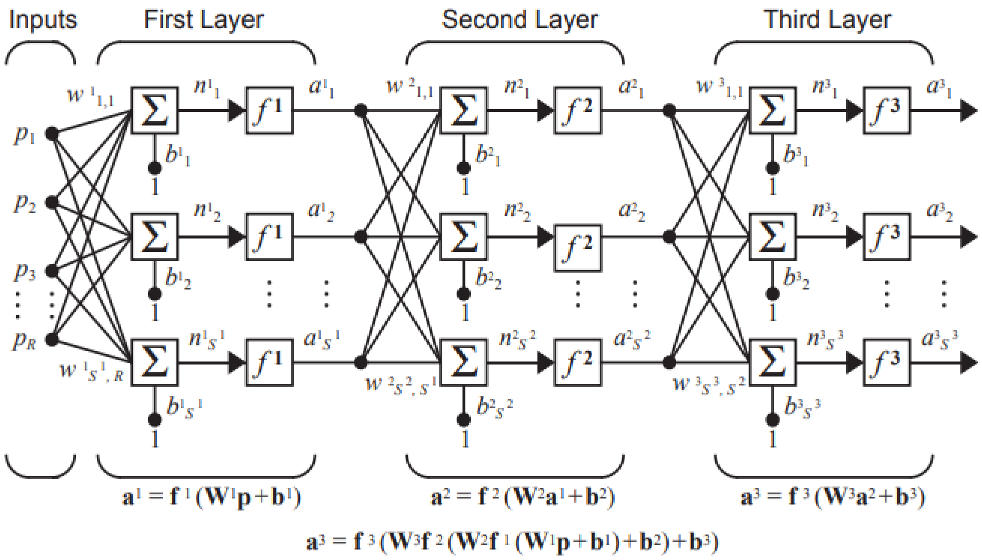

Figure 11: Three hidden layer network with S neurons [10]

Figure 11 explains the summary of the process mentioned above. Although the notations are different in this representation, it can be understood that f represents the activation function and b is the bias term at each node. Also, connection weights can also be represented by a weight matrix, $\boldsymbol{W}$.

In supervised learning model, a general loss function is defined such that it represents the error between the actual results and model results from feedforward operation. The performance index, which is simply a loss function can be given by the following mean squared error equation.

$$F(\theta) = E[(y - z)^2] \tag{11}$$

However, the loss function (or cost function) can also be written in the terminology of neural networks as introduced above. There are different types of cost functions that can be selected.

Quadratic cost (or simply mean squared error) is defined as follows. This is one of the most basic, but very useful function for majority of regression problems.

$$J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(\theta_{j,i}^{(l)}\right)^2 \tag{12}$$

On the other hand, for probability problems, categorical cross-entropy function can be used which is stated as follows.

$$\begin{aligned} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}&\left[y_k^{(i)} \log\left(\left(h_\theta\left(x^{(i)}\right)\right)_k\right)\right. \\ &\left. + \left(1 - y_k^{(i)}\right) \log(1 - \left(h_\theta\left(x^{(i)}\right)\right)_k)\right] \\ &+ \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(\theta_{j,i}^{(l)}\right)^2 \end{aligned} \tag{13}$$

where L is the total number of layers in network

s_l is number of units excluding the bias unit in layer l

K is the output unit

λ is the regularisation term which adds extra term in cost function to improve the generalisation of the network while decreasing overfitting

and θ represents the weight and bias parameters.

The most significant step of training the model is back-propagation in which the gradient of loss function is calculated at each iteration (epoch) to optimise the weights at each neuron and biases that will yield the minimum value of this function, $\min\left(J(\theta)\right)$. Therefore, there is a need to calculate both $J(\theta)$ and $\frac{\partial J}{\partial \theta_{ij}^{(l)}}$.

In backpropagation, error at each output unit is calculated starting from the last layer. This is followed by gradient computation. The essential purpose is to calculate the derivative of the cost function and ultimately understand the effect of each weight on the cost function. This effect is expressed by the chain rule of derivatives.

$$\frac{\partial J}{\partial w_{jk}^{(L)}} = \frac{\partial z_j^{(L)}}{\partial w_{jk}^{(L)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial J}{\partial a_j^{(L)}} \tag{14}$$

Similarly, the effect of biases on the cost function is investigated separately. In the end, the chain rule expressions which give the derivatives contribute the determination of gradient of the cost ∇J. [11]

Optimisation is done based on gradient descent which is one of the most popular algorithms to optimise the neural network. Minimum value of the cost function is tried to be gathered by this algorithm using the derivative of the cost function and predetermined values of learning rate. First the gradient (or slope) is calculated, then a step opposite to the gradient is being moved. The amount of movement, or the step size, is determined by the learning rate, α. Then the weight parameters are updated.

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \tag{15}$$

This update of weights is repeated until the convergence.

There are variants of gradient descent algorithms. In batch gradient descent, gradient of cost function is calculated with respect to all parameters in the neural network to update the weights, whereas in stochastic gradient descent (SGD), it is calculated for each sample (training example, $x^{(i)}, y^{(i)}$) in the dataset and the weights are updated right away making the computation more efficient and faster [12].

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J\left(\theta_0, \theta_1, x^{(i)}, y^{(i)}\right) \tag{16}$$

In SGD, in addition to learning rate, there is also momentum term which helps the algorithm dampen oscillatory behaviour of objective function around steeply changing curves in finding the minima. Momentum, γ, adds fraction to the learning rate affecting the update of weights at the current step. [12]

$$v_t = \gamma v_{t-1} + \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \tag{17}$$

$$\theta = \theta - v_t \tag{18}$$

Model parameters (weights and biases) are optimised as explained above. However, there are certain predetermined parameters that cannot be optimised using standard algorithms that make use of cost function. These parameters are number of layers in the network, number of neurons in each layer, learning rate and momentum of the gradient descent algorithm, weighing term of whichever regularisation is used. These parameters cannot be optimised or estimated from the data set or analytically. Therefore, for a given problem best values for those parameters are not known. They can be manually assigned prior to training of the network.

To select the best set of these parameters, hyperparameter tuning need to be performed using search techniques such as grid search or random search.

Grid search is an algorithm that precisely seeks through all the combinations of parameters and finds the minimum/maximum required by the problem. Therefore, it never misses the best combination, but it is computationally costly. Random search, on the other hand, is performed on set of combination of parameters for which the number can be specified for the design space of parameters. None of these algorithms use the knowledge of which direction the cost function gets minimum and where the best set of combinations lie in the design space like Bayesian search does.

These searching algorithms result in different designs of neural networks and typically, for each design, many the same-networked models are trained and averaged to make use of diversity in weights and biases of the same network stemming from randomness in training.

There are some examples of studies that combine physical problems with machine learning applications.

Figure 12 shows an example of neural networks where airfoil coordinates are input parameters to machine learning model and lift coefficient is the output of the model with concatenation of other parameters and results after the third hidden layer of the model. This artificial neural network structure has 4 hidden layers with 20 neurons at each hidden layer. [13]

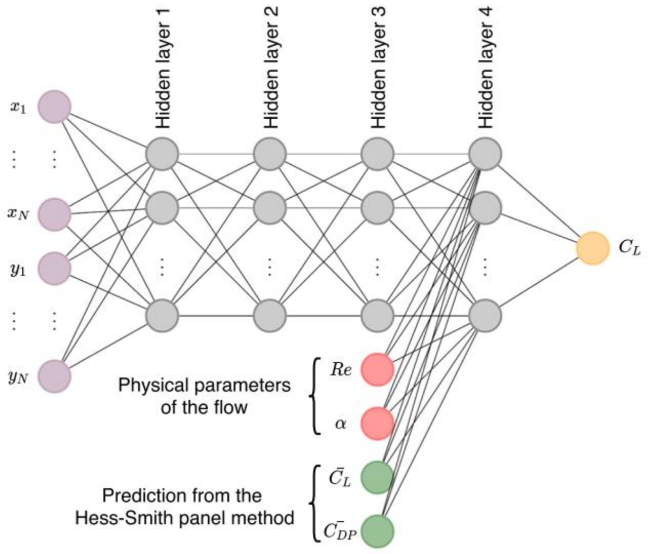

Figure 12: Schematic of inputs and outputs of physics guided neural network model [13]

The resulting lift coefficient predictions from pure machine learning model and physics guided machine learning model are shown in Figure 13 for two different testing airfoils.

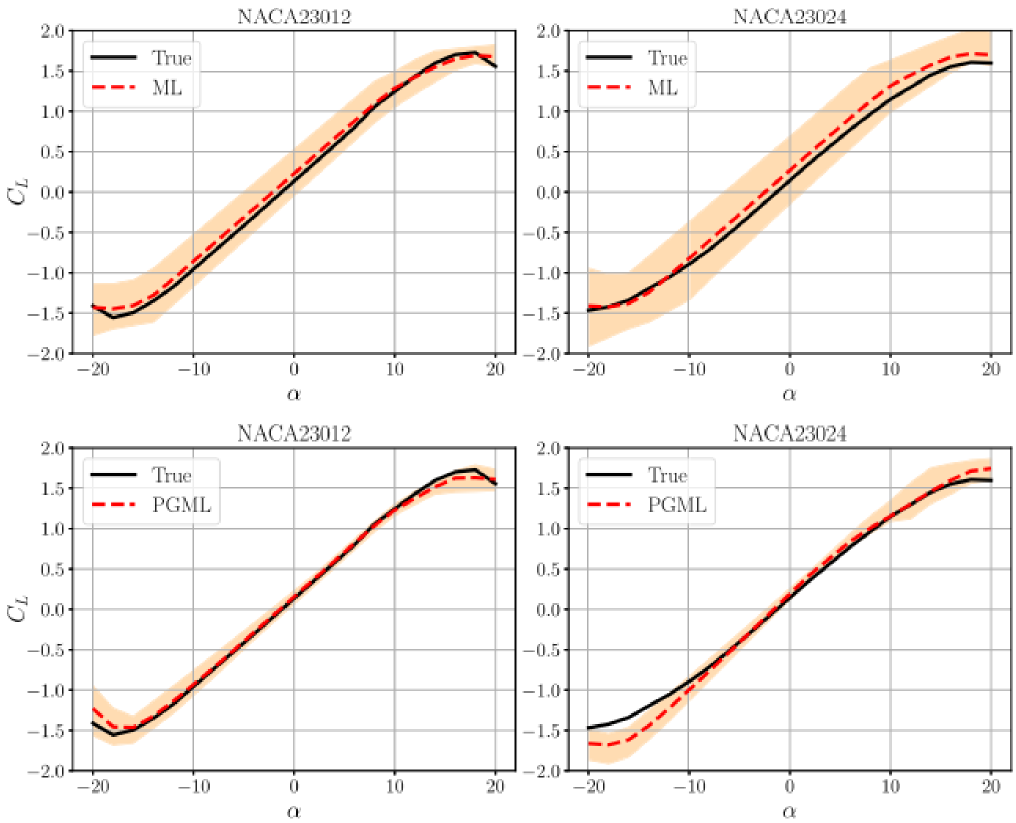

Figure 13: Actual and predicted lift coefficient using PGML [13]

It is shown that, inclusion of physics input makes these types of studies promising and can provide results with less uncertainty. However, this study relies completely on panel methods XFoil results being the main input instead of CFD simulations which otherwise could have been more comprehensive work and Hess-Smith panel method being the physics input.

Another example of neural networks is shown below where a wake flow due to cylinder is investigated.

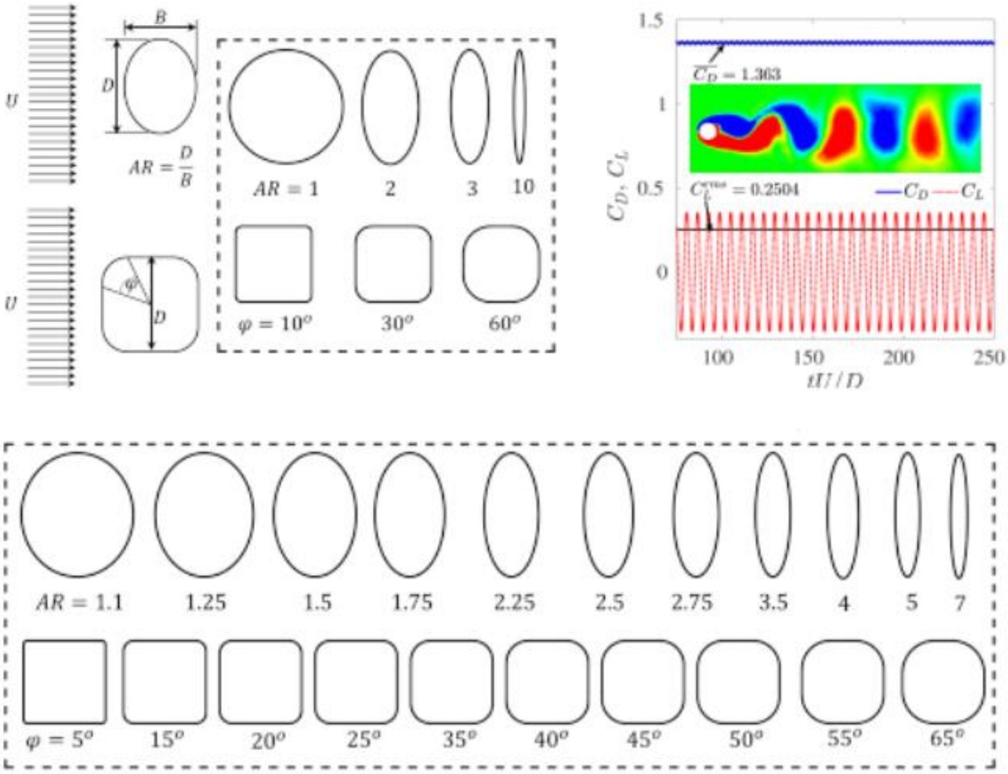

Figure 14: Elliptical cylinder problem definition [14]

Figure 14 shows the definition of the problem where an elliptical object is tried to be modelled using machine learning algorithms. In this problem, RMS values of C_L and C_D are being investigated with respect to aspect ratio of the object. The network in this problem is convolutional neural networks (CNN) which is a type of ANN with convolutional layers and is used mostly for image processing applications.

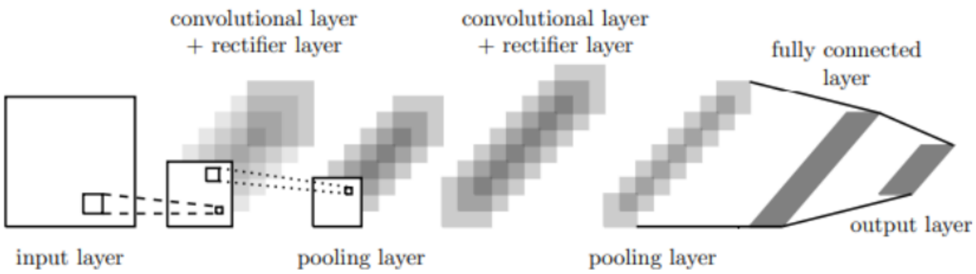

Figure 15: Convolutional neural network schematic [14]

Outcome of this study was shown with RMS values of lift and drag coefficients. The results of RMS values of C_L and C_D are shown in Figure 16.

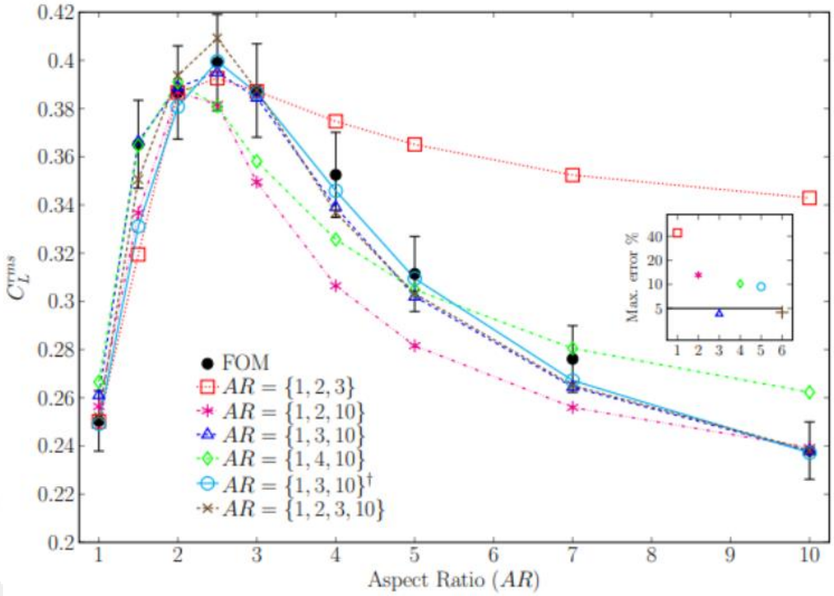

Figure 16: Lift prediction with changing training sets [14]

It was stated that each model represented by each curve on the graph was trained using available full order model (FOM) data with corresponding aspect ratios (AR) only. It is seen that, for example, when the model was trained with AR of 1, 2 and 3, the model cannot predict the wake region in higher aspect ratios.

This shows that when a surrogate model is expected to predict the quantity of interest, there are limitations on the accuracy of the model predictions on intervals where the model was not trained. This is one of the limitations of pure machine learning model with performance of unseen data. For good accuracy of results, one needs to select and train the model with a data set where the predictions are going to take place or combine with a physics guidance applications.

1.4. Project Objectives

The main objective of this project is to create a predictive machine learning model to determine the stall of an airfoil.

To aid this objective, there some steps need to be taken.

1. Simulation and validation of flow around a fixed airfoil to determine and set up the best CFD configuration.

2. Sampling of CFD data set of different thicknesses and angles of attack using determined CFD configurations
3. Start building a machine learning model of a smaller size of inputs which is only trained with thickness of airfoil geometry and angle of attack.
4. Sampling of CFD data set of different camber, position of camber, thicknesses, and angles of attack
5. Building a larger size machine learning model to be trained with camber, thickness, and angle of attack.

2. Methods

2.1. Methods on CFD simulations of airfoil

To comply with the purpose of prediction of lift coefficient and stall region of an airfoil, numerical simulations need to be performed. To simulate the flow around the airfoils or any physical flow, there are basically three options: DNS, LES, and RANS. DNS is the direct solution of Navier Stokes equations that the turbulence is explicitly resolved making it extremely expensive. On the other hand, LES and RANS are computationally less expensive while LES can resolve turbulence in large scales and RANS gives approximate time-averaged solutions. RANS is the CFD simulation that is the quickest among others while providing sufficiently good results. Therefore, the CFD simulations for this project are RANS simulations.

In this project, open-source CFD code SU2 was used to simulate the flow around airfoils which took place at high performance computers (HPC) of Imperial College London. The software uses Reynolds Averaged Navier-Stokes (RANS) equations to solve the flow field around airfoils.

To aid the purpose of completing the CFD simulations, one airfoil was chosen to validate the simulations and make the choice of CFD configurations. By fixing the geometry of airfoil, flow conditions are changed and effect on lift coefficient and therefore, the stall is aimed to investigate. Therefore, the validation section includes the determination of CFD configuration settings using the reference airfoil geometry, NACA0012. It is ultimately aimed that after fixing the CFD configurations, for other simulations the same set up is used to get the required data sets.

2.1.1. Basic setup of CFD simulations

The simulations take place viscous, steady, compressible flow around 2D airfoils using a turbulence model. Flow properties are $Re = 10^6$ and $Ma = 0.15$ which represents well the conditions for wind turbine applications.

Computational domain with a 50 m farfield radius and a 1 m chord length of airfoil inside is shown in Figure 17.

Simulations are run using 1 node and 16 CPUs of HPC at Imperial College.

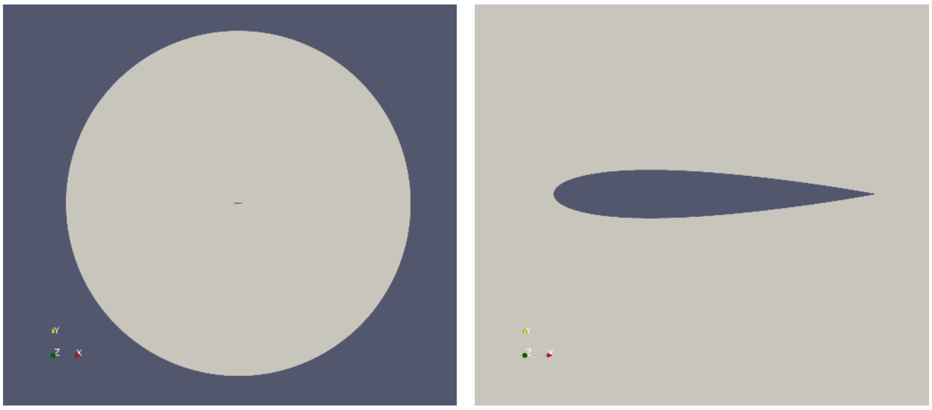

Figure 17: Computational domain and airfoil

As the boundary conditions, flow at farfield circle is identified by the code SU2 as farfield which allows freestream conditions with Re and Ma to be applied, and no-slip at the airfoil surface.

Using a code (python script from Marco Rauseo [15]) that runs the relevant library of the open-source finite element mesh generator, Gmsh, structured and O-type mesh grid for the solution domain was generated. The refinement on the mesh is controlled by the choice of number of points on airfoil and progression of cell sizes radially also to match the reference mesh grid of NACA0012 [16] shown in Figure 18. Therefore, the refinement is first done on the solution domain of NACA0012 airfoil.

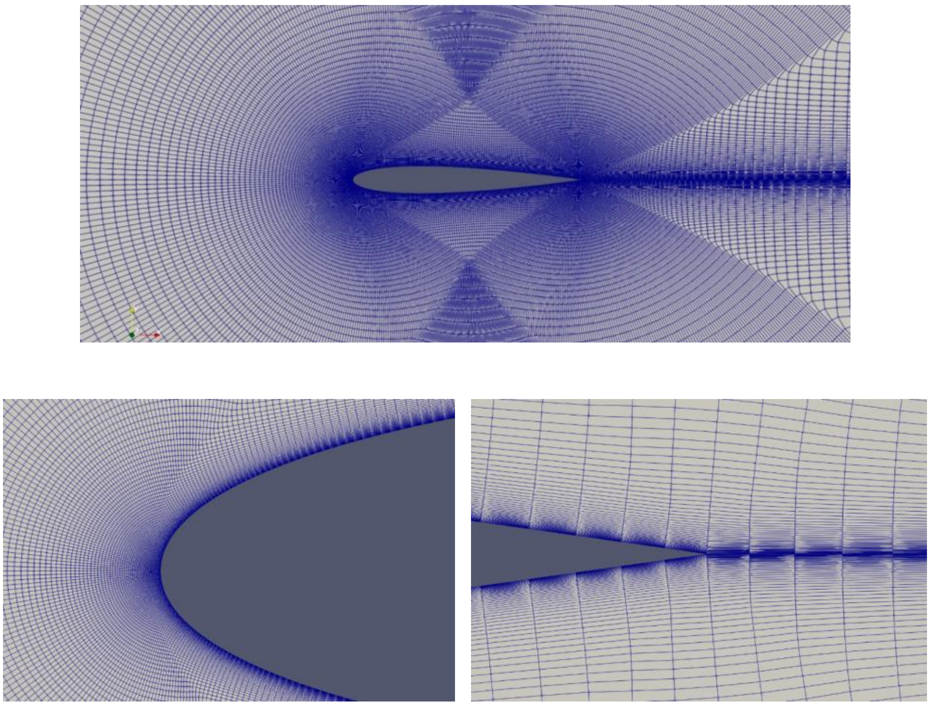

Figure 18: NACA0012 structured mesh

2.1.2. Determination of turbulent model

For determination of the most suitable model in this project, there has been a study on two different models and the results have been assessed.

The determination of turbulent model to use was aimed to obtain the best results for all the airfoil geometries as well as the reference geometry in terms of accuracy of the solution, specifically around stall points. Two different turbulent models have been used and performance of the models have been assessed by comparing the results with the credible CFD simulations and available experimental data [17]. This way, high model accuracy is aimed to achieve.

First model is Spalart-Allmaras Negative one equation turbulence model in which a modification on original SA equation is done to overcome the issues with under-resolved grids and it does not create significant difference when both 'original' and 'negative' models converge [18]. Second model is Menter Shear Stress Transport turbulence model that utilizes two equations for the two variables: turbulent kinetic energy and turbulent dissipation rate.

The two different turbulence models were compared using different grid resolutions from example CFD simulations as will be mentioned in 2.1.4.

2.1.3. Determination of numerical convective scheme

A convection scheme is a numerical method to solve convection-diffusion in the equation resulting from the spatial integration and discretisation of Navier-Stokes equations after relevant manipulations to handle the covariant of velocity fluctuations $\overline{u_i' u_j'}$.

For comparison of convective numerical scheme, grid resolution and turbulence model are fixed, and two schemes are compared: JST (Jameson-Schmidt-Turkel) and ROE. JST is known to generate high accuracy results for the steady state, smooth region of flow where there is no shock etc. Therefore, it retains second order accuracy in smooth regions of flow [19]. On the other hand, ROE is an upwind scheme which is first-order accurate in space. There are other convective schemes that SU2 can utilise some of which are modified versions of the two schemes mentioned above. Since the two mentioned schemes are optimum for airflow and they are the basic versions of their derivatives, they are being considered at this stage of the project.

2.1.4. Determination of mesh resolution

Table 1 shows different resolutions for mesh grids available from NASA [19]. From the table, some of the grid resolutions are selected for the purpose of completing previous as well as seeking the convergence of mesh on the other selected configurations of CFD simulations.

Table 1: Different grid options for NACA airfoils [19]

Grid Class	Grid Resolution	Number of points on airfoil
1	7169 x 2049	4097
2	3585 x 1025	2049
3	1793 x 513	1025
4	897 x 257	513
5	449 x 129	257
6	225 x 65	129
7	113 x 33	65

For the trial of different mesh sizes, grid class ‘3’ was chosen as the starting point. It was also planned that if needed, simulations were going to be repeated with higher resolution of mesh grid.

2.2. Methods on machine learning

Most of the time, CFD analysis that need to be completed to find out the stall take a lot of time and makes the standard design process an expensive procedure. Lennie et. al. states that computationally cheaper solutions have been sought to complete this process [20]. Therefore, it has always been a design aim to find both computationally not expensive and reasonably accurate solutions. For this reason, positive aspects of machine learning algorithms were tried to be combined with CFD simulations to seek the best way to perform the stall analysis.

Since this project is focused on aerodynamic coefficients and other parameters being known, all the data is labelled, and the type of application is supervised here. In the literature, physics guided machine learning models have been mostly found in neural network structures. Neural network models are the optimum for the type of models since results from a simplified model or physical laws are easily being injected in the model.

This project includes the modelling of certain airfoil geometries which creates the development data set for the machine learning model. The constructed neural network surrogate model generalises the prediction of lift coefficient from a range of parameters.

This project consists of labelled data, i.e., the training dataset given to the machine learning algorithm has its headings in which there is a one-to-one match of variables, which means there is single lift coefficient value for one camber, thickness, angle of attack. Therefore, for the development of this project, neural networks structures were constructed. As explained in 1.4, the problem has been handled in two steps.

1. In the first step, smaller dimensional network was trained, Model 1.

After creation of this model, hyperparameter study, and predictions on training data set interval and on unseen data (i.e., interpolation and extrapolation examples) are performed.

2. In the second step, wider network was trained with additional variables of camber and position of maximum camber, Model 2.

Therefore, the first step helps the establishment of the second step where a larger network is trained with parameters other than thickness as well.

The input variables in the first step of the project are

$$\boldsymbol{x} = \{\, thickness, angle\ of\ attack\}$$

whereas in the second step they are

$$\boldsymbol{x} = \{camber\ amount, camber\ position, thickness, angle\ of\ attack\}$$

making it four input model instead of two.

Similarly, the output variable is expressed as follows.

$$\boldsymbol{y} = \{lift\ coefficient\}$$

In this project, as the cost function, mean squared error (MSE) is used which is one of the best choices for regression problems like this. As the gradient descent algorithm, stochastic gradient descent (SGD) is used which has an advantage over the batch gradient descent as explained in 1.3.1 that the convergence of training of parameters are faster, etc.

Explanation of purely machine learning (ML) network and physics guided machine learning (PGML) network is based on the first step of training the neural networks in this project:

smaller dimensional model, Model 1. In this model, thickness of the airfoil and angle of attack are the two inputs of the artificial neural network while the lift coefficient is the output that the training and prediction of the machine learning model will be based on.

2.2.1. Artificial neural networks set up

For both pure machine learning and physics guided machine learning models, following set up is applied.

Mean squared error is used for the cost function. SGD optimiser with a learning rate of 0.01 and momentum of 0.1 is applied. 1000 epochs are used since when the training loss is checked against epochs during the process, this amount seems sufficient. No folding of training data set is done. Fitting and predictions are done using more than one network and their results are averaged. For this reason, random seed has not been set to a fixed value.

For PGML models, XFoil input has been concatenated at the hidden layer which is one layer before the last hidden layer for every possible network except for 2 hidden layered networks for which it is applied at the last hidden layer.

Normalisation is done as explained in 1.3.1. After model training, to make predictions, all the parameters other than angle of attack are fixed for an airfoil to get the predictions with respect to angle variation. Resulting values are denormalised with from scaled outputs to meaningful lift coefficient values.

2.2.2. Pure neural network model, ML

The ML network is illustrated in Figure 19 with representative artificial neural network schematic.

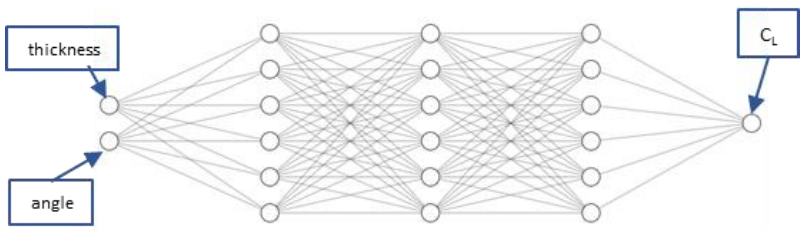

Figure 19: Pure machine learning model (ML)

The artificial neural network in Figure 19 is the most traditional model which can also be called ‘black box’ model since it does not accept intervention of any extra input in the prediction stage.

2.2.3. Physics guided neural network model, PGML

The PGML network is illustrated in Figure 20 with a representative artificial neural network schematic.

Figure 20: Physics guided machine learning model (PGML)

This network has very similar structure to ML network, except that it has additional input in one of the hidden layers, making the number of neurons in that layer $L+1$ and making the network more advanced model by just inputting physics input as such.

Continuation on the wider problem, Model 2, will be quite similar to the steps explained here and will be expansion of the concept, without loss of generality.

2.2.4. Physics guidance from the vortex panel method, XFoil

In this project, a vortex panel method solver called XFoil was used to solve characteristics of airfoil. XFoil uses high-order panel method and can use it with fully coupled viscous interaction method [21]. It uses the vortex panel method explained in 1.2.2.

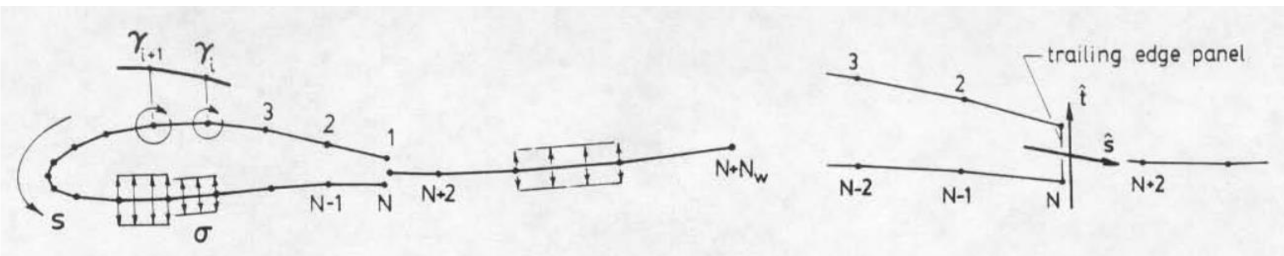

Figure 21: Airfoil and wake paneling with distributions of vorticity, γ and source, σ [22]

The solution procedure relies on panel method that superposes freestream flow, vortex strengths on sheet, and source strengths on sheet into a streamfunction. Depending on the number of panels on the sheet, several variables are introduced for vorticity and source terms. Noting that each panel and wake panel has constant strength of source and requiring the streamfunction to a value at each node a linear system of equations are developed which is followed by requiring the Kutta condition. [22]

$$\gamma_1 + \gamma_N = 0 \tag{19}$$

Limitation of XFoil is that the solutions from panel methods are of lower fidelity with respect to CFD simulations, although on the other hand, RANS simulations cannot resolve the flow eddies in full detail and shows the turbulence as if it is isotropic. Therefore, as a low fidelity solution, XFoil is used to provide physics information in machine learning algorithm, it is not used by itself to validate simulations as it works using the vortex panel method principles explained in 1.2.2.

2.2.5. Selection of dataset for Model 1

Figure 22 shows the description of NACA four-digit airfoils. First digit shows the maximum camber of the airfoil C_{max}, whereas the second digit shows the location of this maximum camber in one tenth of percent of its chord and last two digits show the thickness of airfoil as percent of its chord.

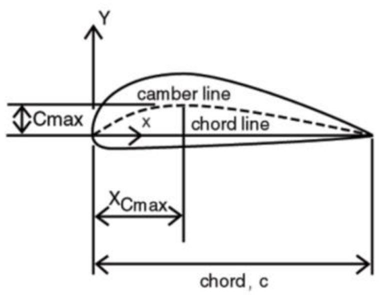

Figure 22: Description for NACA four-digit airfoils [23]

To sample the dataset for this model, it has been benefitted from the last two digit of NACA four-digit airfoil representation. Therefore, keeping the two digits 0 (symmetric airfoils) and changing the last two digits, sampling for the thickness of airfoil was performed.

To test the extrapolation performance of the two models, training data set is not chosen from all thicknesses; instead, data from some thicknesses are selected. Therefore, the data set is split into two sets shown in Table 2.

Table 2: Data set interval for different thicknesses of NACA four-digit airfoils

Data set	Interval of thickness (% of chord)
DS1	10,11,12,13,14,15
DS2	16,17,18,19,20,21

For the training purpose of model, ds1 is selected. 70% of ds1 which is randomly selected is utilised as training data set, while the remaining 30% is used as testing data set. Other data set, ds2, is used to test the performance of the model on unseen data. In other words, from the produced database, thin airfoils are selected to train and test the model whereas thick airfoils are used to assess the extrapolation performance of the designed models.

This study is expected to reveal the generalisation of the machine learning model over other thicknesses that the model hasn't trained for and give indication on how physics information improves the prediction when compared to purely machine learning model where there is no physical input to the neural network. For this purpose, both models are structured and trained using the same conditions, except in the physics guided network, there is physics information concatenated in one of the layers.

2.2.6. Selection of dataset for Model 2

To sample the dataset for this model, NACA four-digit airfoils have been used. This time, all four the digits of NACA four-digit airfoils are varied in order to get a wide range of airfoil geometries.

Table 3: Data set intervals for NACA four-digit airfoils for Model 2

Parameter	Interval
Camber	1,2,3,4,5,6
Position of camber	3,4,5
Thickness	10,11,12,13,14,15,16,17,18,19,20,21

For the selection of geometries that include information about camber, position of camber, and thickness, near-random selection was made using Latin Hypercube sampling.

In LHS, sampling is done to make wide range of variables in their intervals such that samples do not accumulate somewhere in the design space, and they are spread to represent real variability as illustrated schematically in Figure 23.

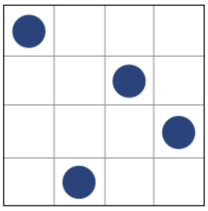

Figure 23: Schematic for Latin hypercube selection [24]

Sampling is done on intervals shown in Table 3 as the mentioned variables. Resulting airfoil geometries after sampling are shown in Figure 24.

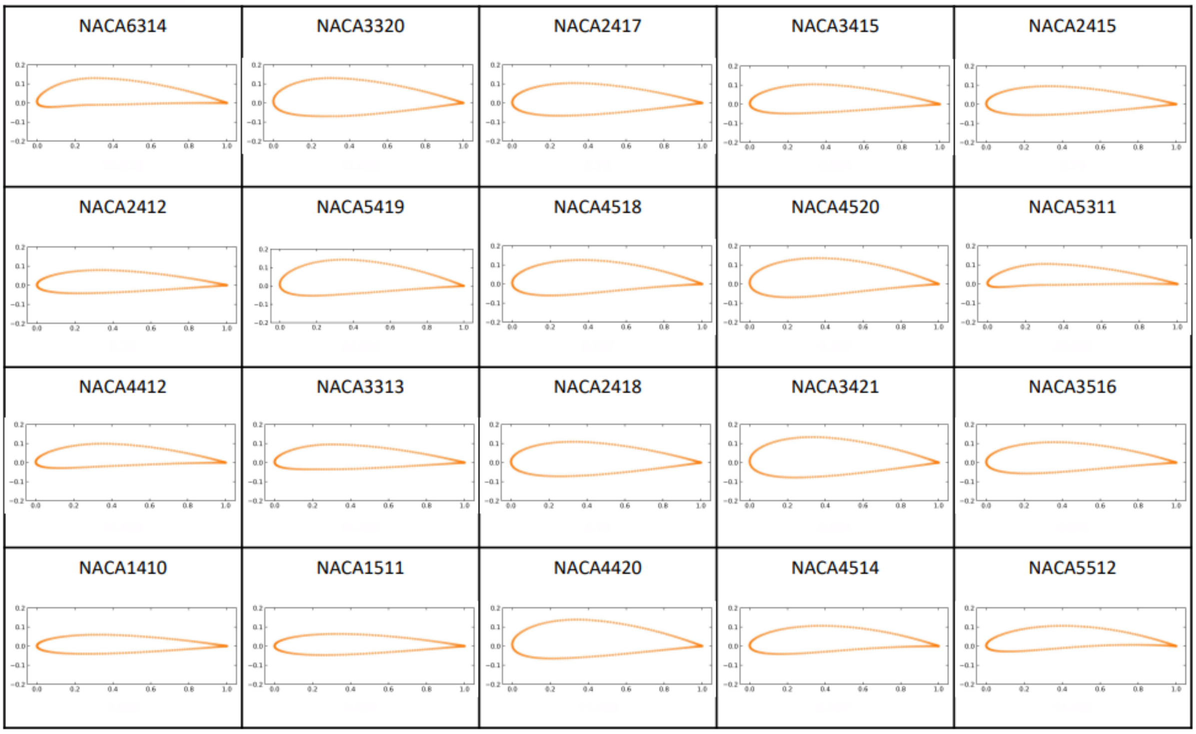

Figure 24: Sampled airfoil geometries for Model 2

For the training of ML and PGML in Model 2, CFD results of these airfoils with different angles of attack were used.

Table 4: Training data set interval for Model 2

Parameter	Interval
Camber	3,4,5
Position of camber	3,4

Parameter	Interval
Thickness	12,13,14,15,16,17,18,19

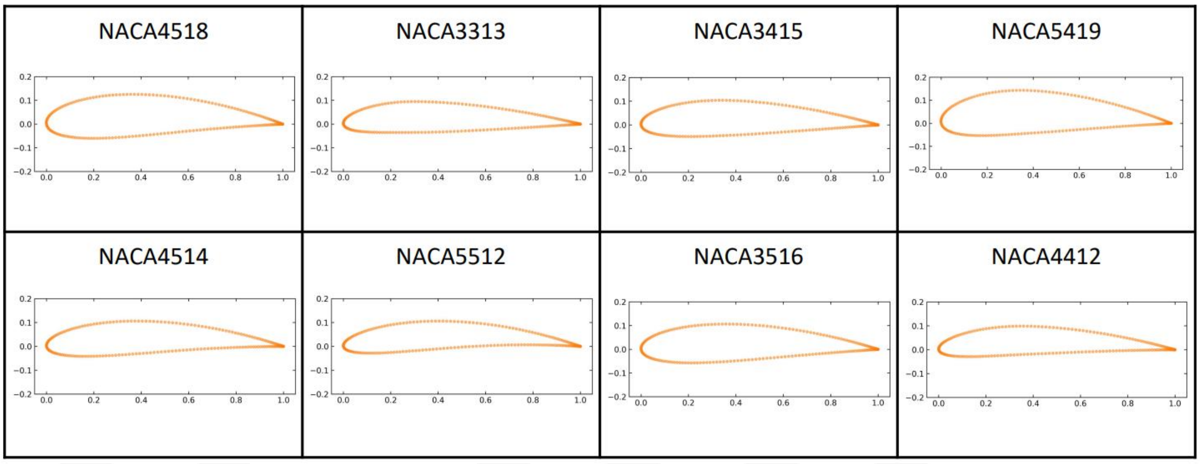

Figure 25: Training dataset of airfoils for Model 2, DS1

Table 4 and Figure 25 show the interval of training airfoils and their geometries among the sampled airfoils, respectively which is labelled as DS1. The remaining airfoils correspond to unseen data with respect to camber, position of camber, or thickness with the geometries shown in Figure 26. They are used to show the extrapolation performance of Model 2.

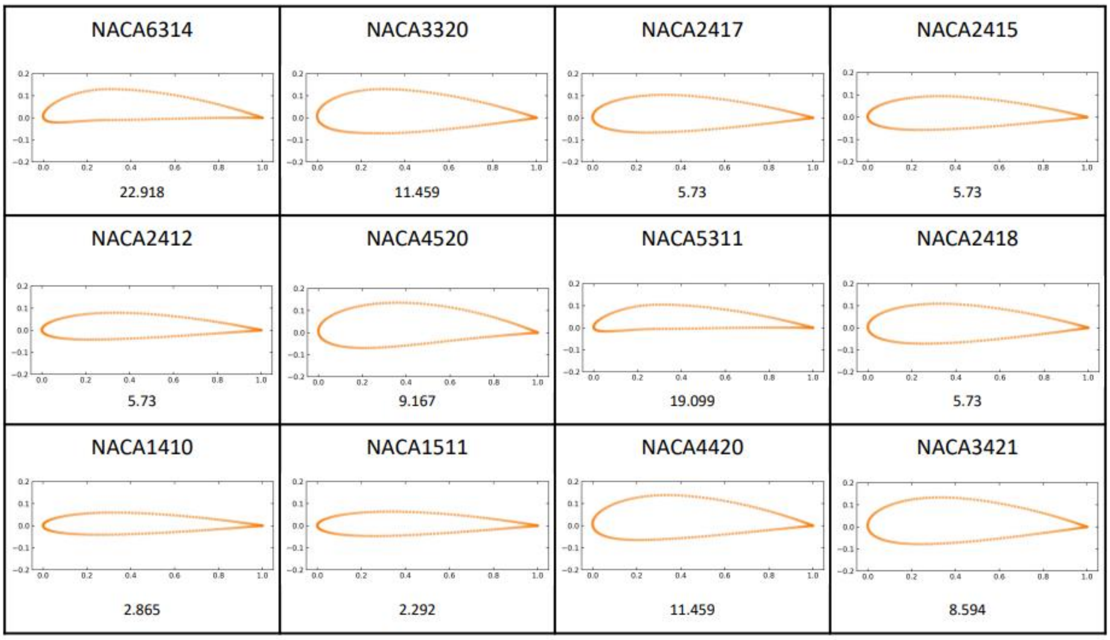

Figure 26: Testing data set of airfoils for Model 2, DS2

2.2.7. Hyperparameter optimisation for Model 1 and Model 2

In this project, among the hyperparameters mentioned, number of layers, number of neurons and weighting of L_2 regularisation are being tuned. Best set of layers and neurons are searched together in grid search. After determining number of layers and neurons together, search on regularisation is performed in a separate algorithm.

Table 5: Hyperparameter intervals

Parameter	Parameter interval
Number of layers	2,3,4,5,6
Number of neurons per layer	2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
L_2 Regularisation	$10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10$

Hyperparameters are tuned according to their intervals selected as in Table 5. Method to determine the layers, neurons and regularisation term is as follows.

- Layers and neurons are changed sequentially in a grid search while keeping the number of neurons in hidden layers the same for all layers.
- L_2 regularisation is separately searched in a sequential search.

Method of getting the validation and training loss value can be explained as follows.

1. Model fitting is completed using 1000 epochs, and SGD optimiser with learning rate of 0.01 and momentum of 0.1.
2. Activation functions for all hidden layers are ‘relu’ standing for rectified linear unit which outputs the value itself if it is positive and zero, otherwise.
3. Validation and training loss at the end of every model fit are taken as a function of epoch and last 50 values of validation and training losses corresponding to last 50 epochs are averaged to get rid of fluctuations.
4. For every possible neural network occurring in the grid search, 15-25 same-networked model are trained, and their loss values are averaged.

3. Results

3.1. CFD Results

3.1.1. Results based on different turbulence modelings

Starting from grid class '3' from Table 1, following results were obtained with numerical convective scheme of ROE as shown in Figure 27.

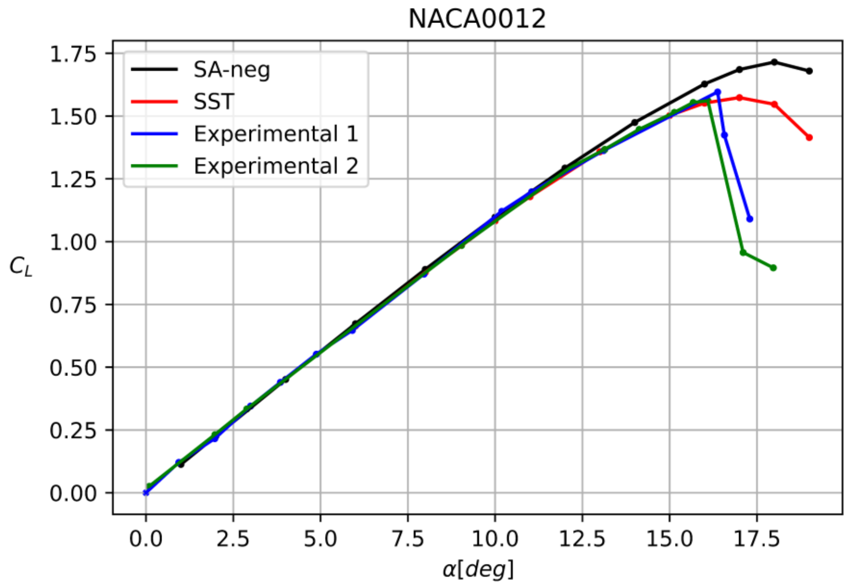

Figure 27: Lift coefficient of NACA0012 using Spalart Allmaras, Shear Stress Transport models with the grid class of **3** from Table 1 with the convective scheme of ROE and experimental lift coefficients [17]

The CFD results using Spalart-Allmaras Negative and k-ω Shear Stress Transport models follow experimental lift coefficients for low angles of attack strictly while there is some difference around stall angle. When the lift around stall point is considered, SA model overpredicts the lift coefficients with respect to experimental coefficients. Moreover, if one needs to get the stall point using first derivative of the resulting curve from CL of CFD simulations, SST model seems to perform more accurate than SA model in that sense.

Similar to the grid resolution of 1793 x 513 (class of '3'), comparison of turbulence models with different resolutions of 897x257 (class '4') and 449x129 (class '5') is shown in Figure 28 and Figure 29.

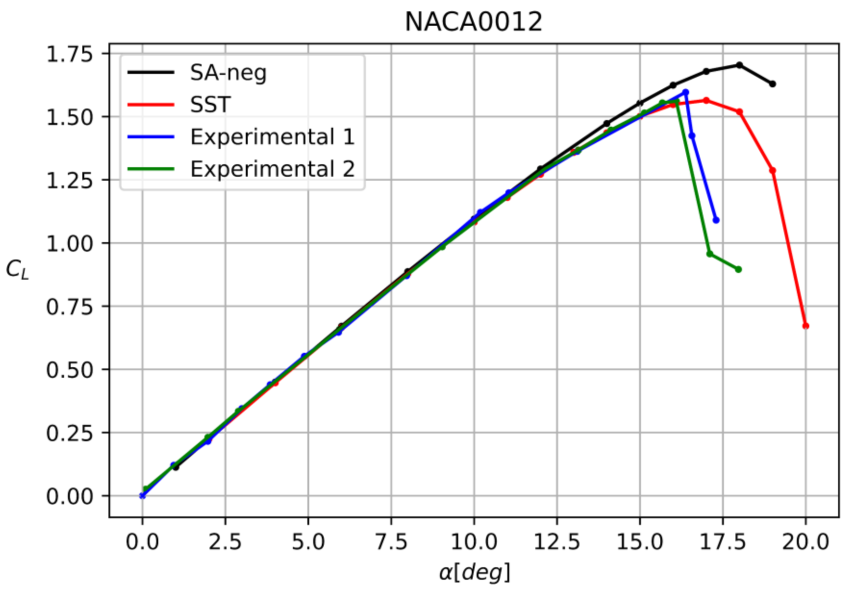

Figure 28: Lift coefficient of NACA0012 using Spalart Allmaras, Shear Stress Transport models with the grid class of **4** from Table 1 and with the convective scheme of ROE and experimental lift coefficients [17]

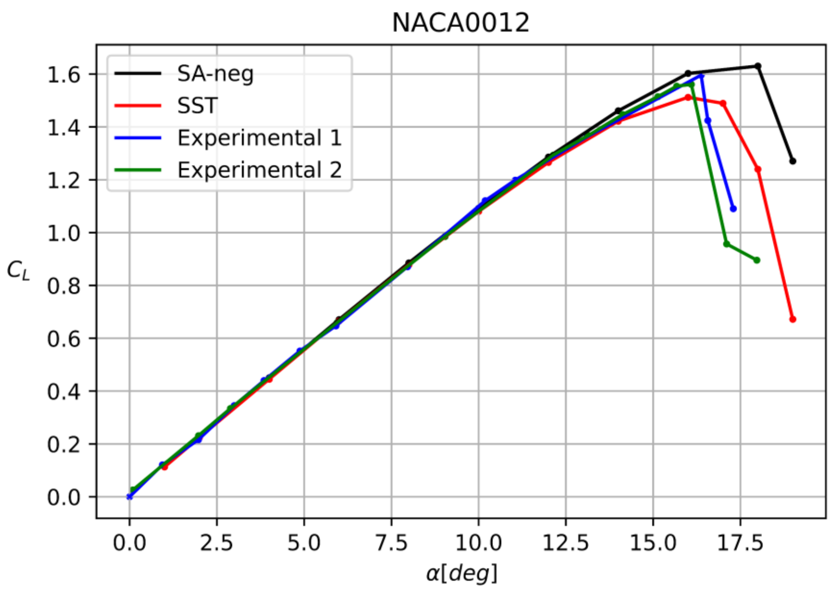

Figure 29: Lift coefficient of NACA0012 using Spalart Allmaras, Shear Stress Transport models with the grid class of **5** from Table 1 and with the convective scheme of ROE and experimental lift coefficients [17]

Looking at all the results, one can say that around stall point, SST underpredicts the lift coefficient; however, in terms of predicting the stall, it has a better performance.

3.1.2. Results based on different convective schemes

Considering the convective schemes mentioned in 2.1.3, the resulting plots are compared as shown in Figure 30.

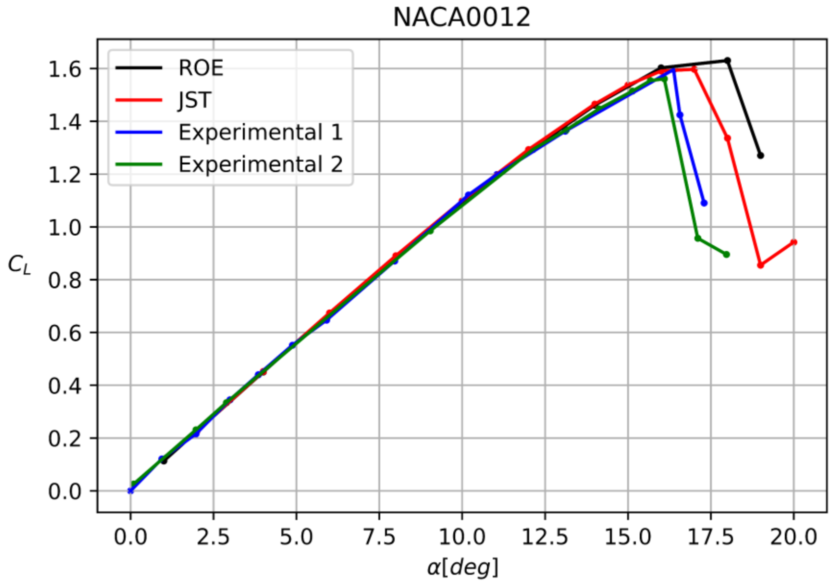

Figure 30: Lift coefficient of NACA0012 using JST and ROE numerical schemes with Negative Spalart Allmaras model with the grid class of **5** from Table 1 and experimental lift coefficients [17]

As the JST convective scheme follows the experimental curve better, it has been chosen for the numerical simulations of all the airfoils.

3.1.3. Results based on different mesh grid resolutions

Having selected turbulence model and numerical convective schemes, results of different grid resolution were compared as in Figure 31. A CFD simulation gets more accurate when the resolution is increased. However, having a quite fine grid increases the computation time and required memory. Therefore, an adequately fine mesh resolution needs to be selected without compromising the solution accuracy.

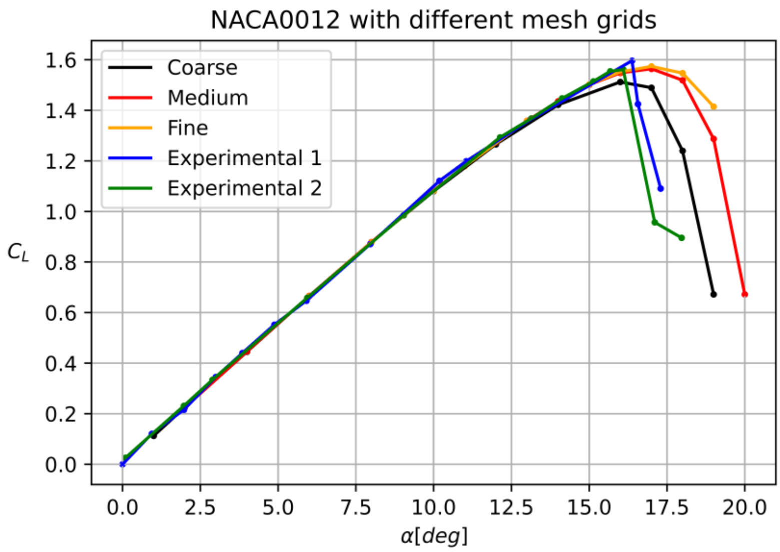

Figure 31: Lift coefficient of NACA0012 with SST turbulence model and JST convective scheme and different mesh grid resolutions and experimental lift coefficients [17]

Looking at this comparison and considering the simulation times, medium grid size that corresponds to grid class **4** from Table 1 has been chosen for this airfoil, and grids of similar size have been utilised for other airfoil geometries. Order of resolution of such value requires computational time of around one hour with 16 CPUs, as well as depending on the number of iterations.

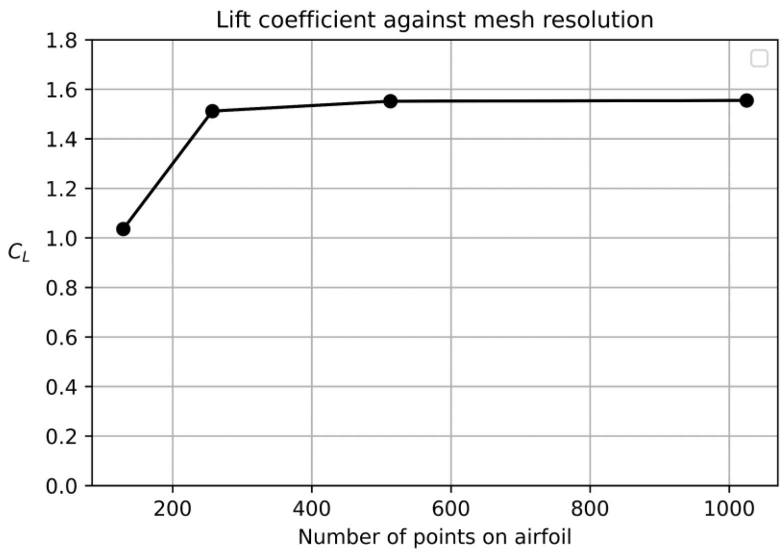

Figure 32: Lift coefficient of NACA0012 at 16° angle of attack with increasing mesh resolution

From another perspective, the lift coefficient of NACA0012 at an angle close to stall point was chosen to show the mesh sensitivity as in Figure 32. It is seen that the curve shows a convergence to ~1.55. In this figure, even coarser mesh is included to show the effect of mesh resolution on the result. The selected mesh size is the one that has 513 grid points on airfoil which is the third point on the curve.

3.1.4. CFD Results for PGML Model 1

Having set all the configurations mentioned, a data set of lift coefficients of different thicknesses of symmetric airfoils was produced for Model 2 of machine learning algorithm which tries to generalise the prediction of unseen geometries of airfoil.

Data set of varying thicknesses of symmetric airfoils was produced using the predetermined configuration of simulations.

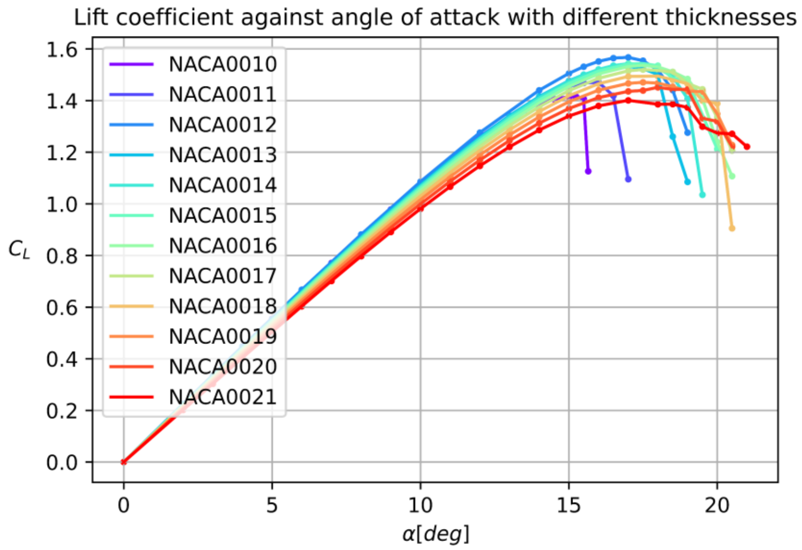

Figure 33: CFD results of lift coefficient against angle of attack curve for symmetric airfoils of different thicknesses with rainbow colour coding

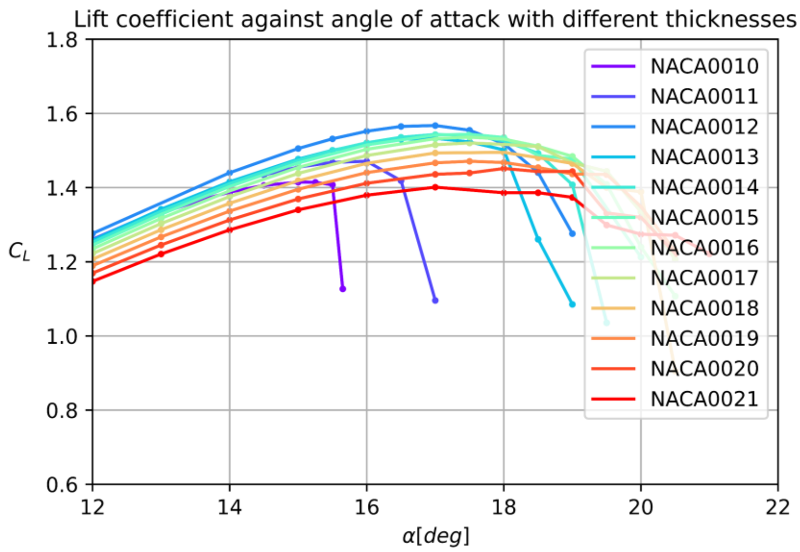

Figure 34: CFD results of lift coefficient against angle of attack curve for symmetric airfoils of different thicknesses with rainbow colour coding (zoomed-in)

The resulting data set is shown in Figure 33 and a zoomed-in version of it in Figure 34.

For this set of lift coefficients, it can be noted that as the thickness of airfoil increases, stall angle increases while the lift coefficient for lower angles slightly decreases. Furthermore, for thin airfoils decrease in the lift coefficient after stall is more abrupt than that of thick ones. This is a result of the fact that thin airfoils show a plate-like behaviour where the flow starts to separate closer to the leading edge resulting in a sudden wake region at top surface of airfoil whereas for thicker airfoils flow separates from the trailing edge which allows the flow to follow upper surface for larger interval of angle of attack.

To illustrate the results, momentum plots and C_P plots of NACA0010, which is the thinnest airfoil in this data set, are shown below.

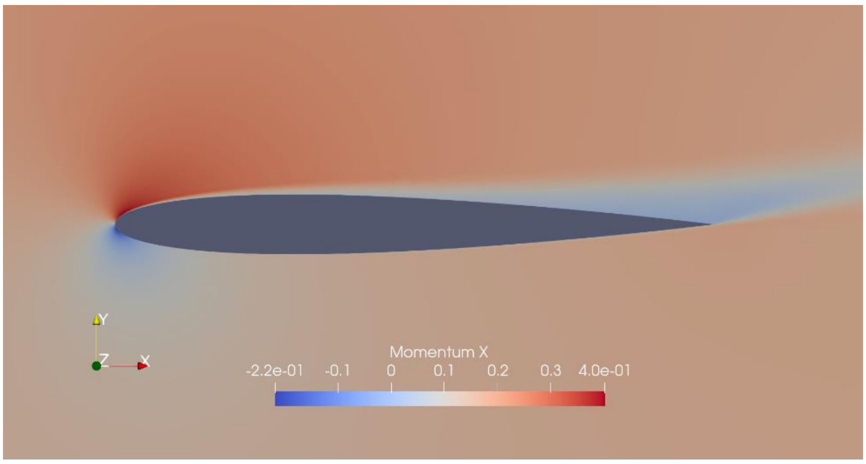

Figure 35: X-momentum contour plot for NACA0010 airfoil at 15.5° angle of attack

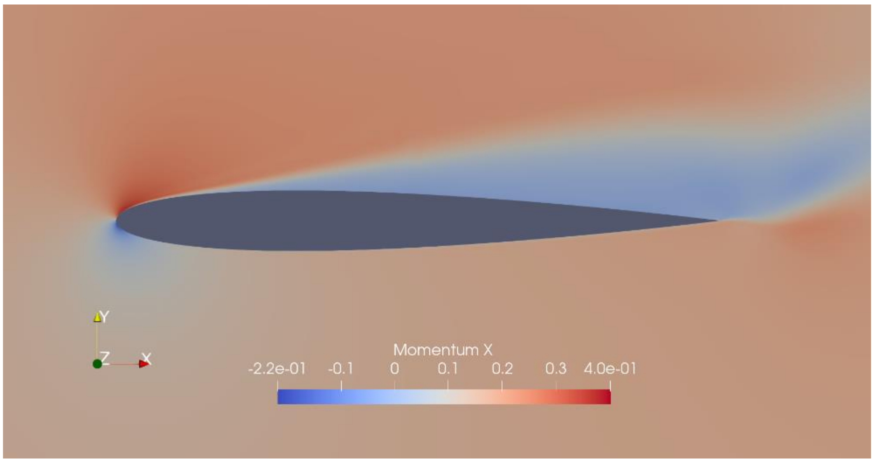

Figure 36: X-momentum contour plot for NACA0010 airfoil at 15.65° angle of attack

It is important to note that the occurrence of wake region at 15.65° angle, shown in Figure 36, take place in very small angle interval and this illustrates how the flow starts to separate and stall from the leading edge for thin airfoils and result in a more abrupt drop in the lift with respect to thicker airfoils shown in Figure 33.

The sudden decrease in the lift coefficient for NACA0010 can be illustrated by the pressure coefficient plot in Figure 37. There is a large increase in the pressure at the top surface because the flow is detached and creates turbulent region.

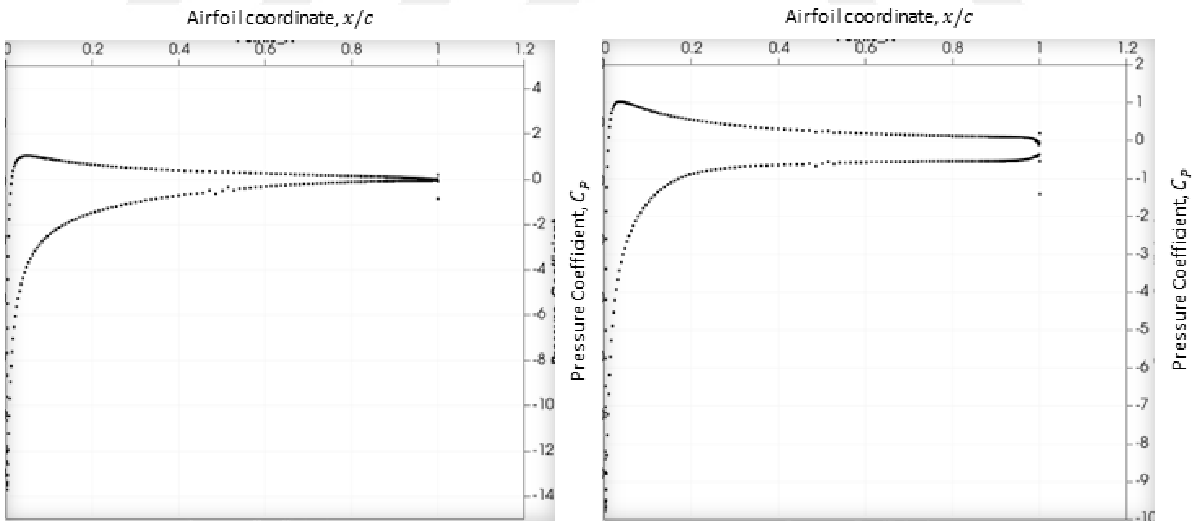

Figure 37: C_P plot of NACA0010 airfoil at 15.5° (left) and 15.65° (right) angles of attack

On the other hand, for NACA0012 airfoil, the angles are chosen apart such that at 16° and 19° were checked, and they are shown in Figure 38 and Figure 39.

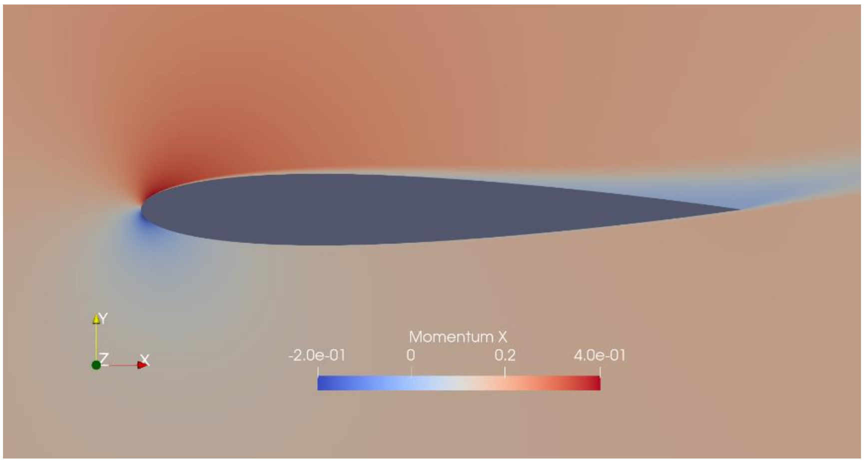

Figure 38: X-momentum contour plot for NACA0012 airfoil at 16° angle of attack

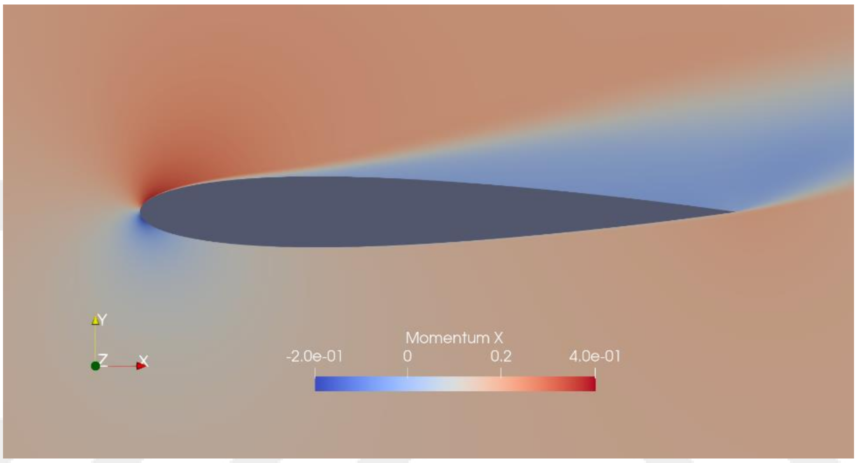

Figure 39: X-momentum contour plot for NACA0012 airfoil at 19° angle of attack

This decrease in lift coefficient in 3 degrees apart cause the airfoil to stall and respective C_P plots are shown in Figure 40.

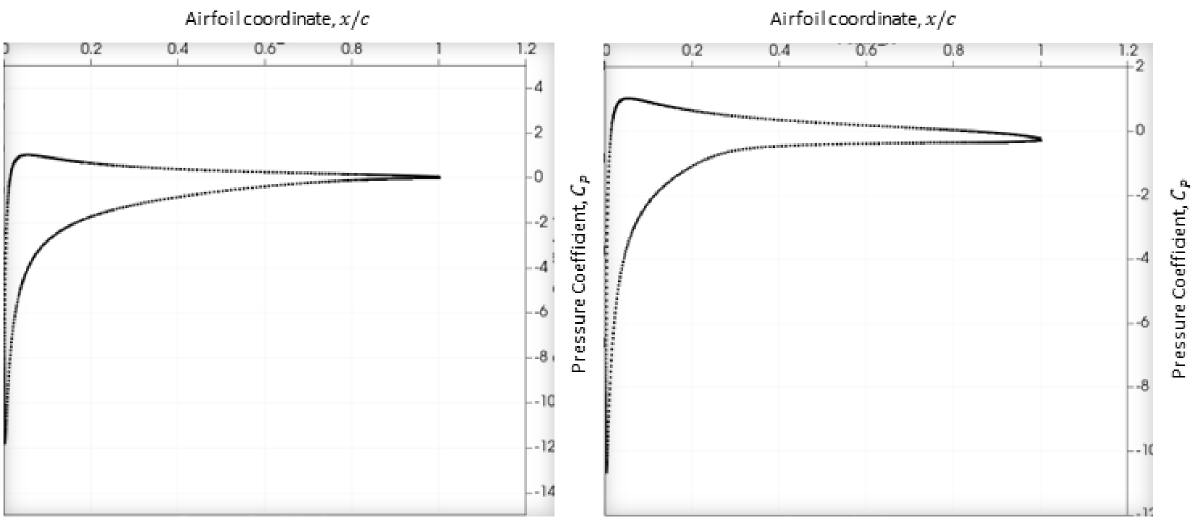

Figure 40: C_P plot of NACA0012 airfoil at 16° (left) and 19° (right) angles of attack

As a summary of CFD simulations, stall angle of symmetric airfoils is presented in the stall against the thickness of airfoil plot in Figure 41. The method to get the stall angle was by fitting a polynomial to each lift curve and taking the angle where the first derivative becomes zero for the resulting fitted polynomial equation.

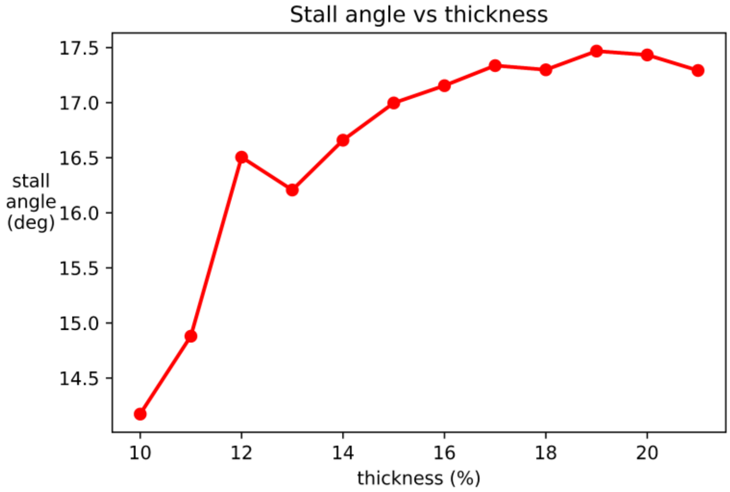

Figure 41: Stall against percent thickness of airfoil with respect to its chord

The result shows the behaviour that the thicker airfoils tend to stall at larger angles of attack while the thin airfoils stall at lower angles with kind of a logarithmic increase in trend.

3.1.5. CFD Results for PGML Model 2

The data set for Model 2 is presented in this section. CFD results of different airfoils of DS1 and DS2 of Model 2 are represented separately for better readability.

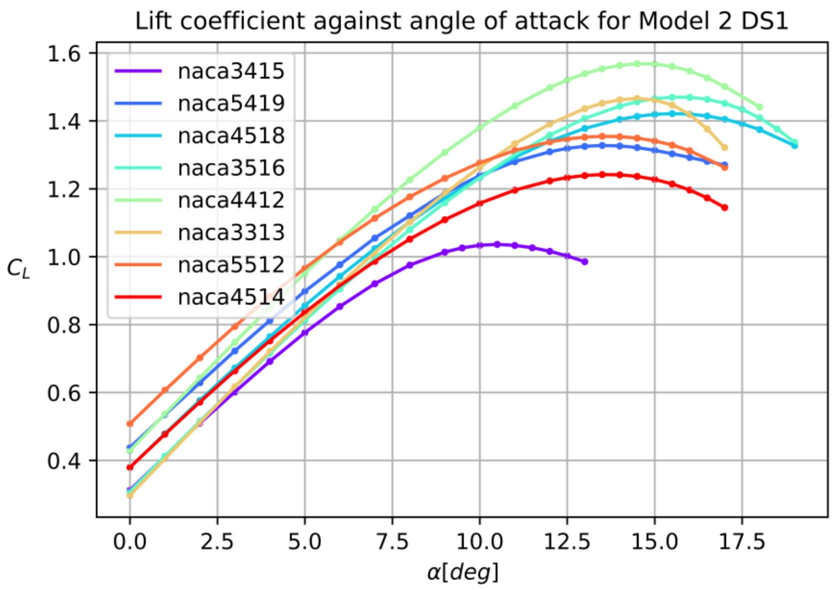

Figure 42: CFD results of lift coefficient against angle of attack for airfoils which compose DS1 of Model 2 with rainbow colour coding

Figure 42 shows the CFD results for the airfoils that are in the training dataset of Model 2 (DS1). As the airfoils are not symmetric anymore, the lift coefficients do not start with zero lift at zero angle of attack, and their behaviour vary according to their camber, position of camber, and thickness.

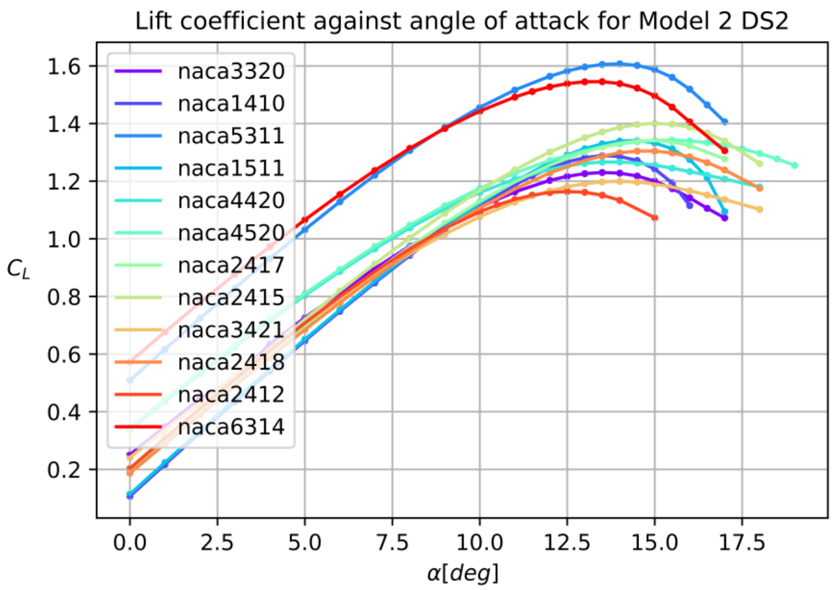

Figure 43: CFD results of lift coefficient against angle of attack for airfoils which compose DS2 of Model 2 with rainbow colour coding

A set of airfoils that are not involved in the training data set is shown in Figure 43 which is solely used for testing airfoils in the model and constitute the DS2 of Model 2.

For all the data sets (airfoils), sampling is increased around the stalling region to make the model capture stalling behaviour of each characteristic curve. Also, all airfoils have certain thickness that flat plate-like stall is not seen in any of the C_L curves in this section. Increasing thickness decreases the decay rate of the lift coefficient at stall while amount of camber affects the lift coefficient at zero angle of attack.

Another important deduction from these results can be that the lift curves do not change so sudden as the results of Model 1. This is because the lift curves in Model 1 data set correspond to symmetric airfoils and they follow a lift curve and stall that the flow separates from the leading edge and this is more obvious for especially thin and symmetric airfoils. On the other hand, airfoils in Model 2 data set are cambered airfoils which tend to stall starting from the trailing edge of the airfoil which makes the airfoil not reduce the lift as the thin and symmetric airfoils do. This is shown for a highly cambered airfoil NACA5419, as shown in Figure 44 and Figure 45 which show the X-momentum contour plots before and after the stall.

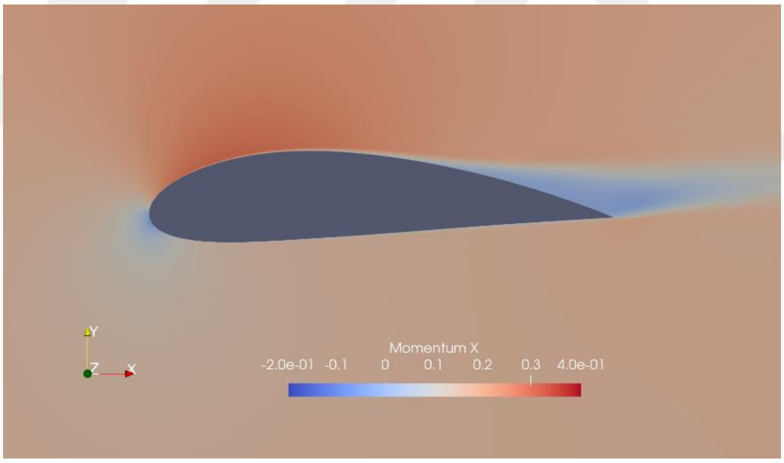

Figure 44: X-momentum contour plot for NACA5419 at 11° angle of attack

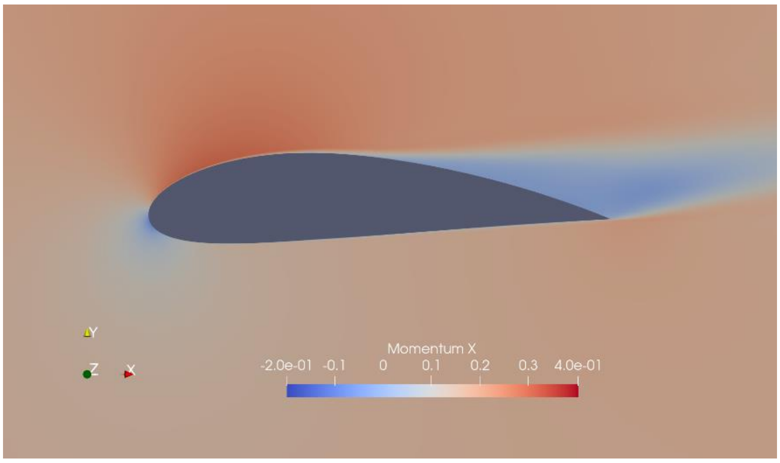

Figure 45: X-momentum contour plot for NACA5419 at 14° angle of attack

From the plots, it is seen that even after the stall, flow is still attached to the leading edge side of upper surface, which makes the lift coefficient do not decrease abruptly.

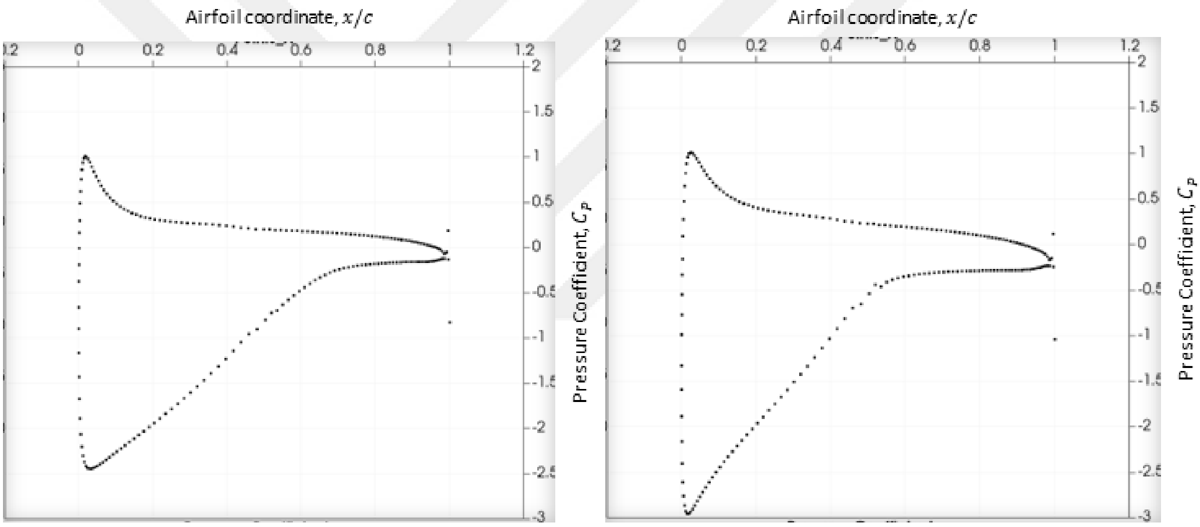

Figure 46: C_P plot of NACA5419 airfoil at 11° (left) and 14° (right) angles of attack

Figure 46 shows the C_P plots for NACA5419 airfoil before and after the stall, and it can be seen that there is no sudden difference when the flows at three degrees apart are investigated when compared to symmetric and especially thin airfoils which have the change in C_P values at the upper surface quite large after stall. On the other hand, in fact, the pressure on upper surface close to leading edge has a decreased value; however, the surface close to trailing edge gets increased pressure because of the wake region being increased after separation and stall.

3.2. PGML Results

3.2.1. PGML Model 1 – Hyperparameter optimisation

Figure 47 shows the study on selection of layers and neurons on purely machine learning model with 70% of DS1 of Model 1 as training data and remaining 30% as the validation data.

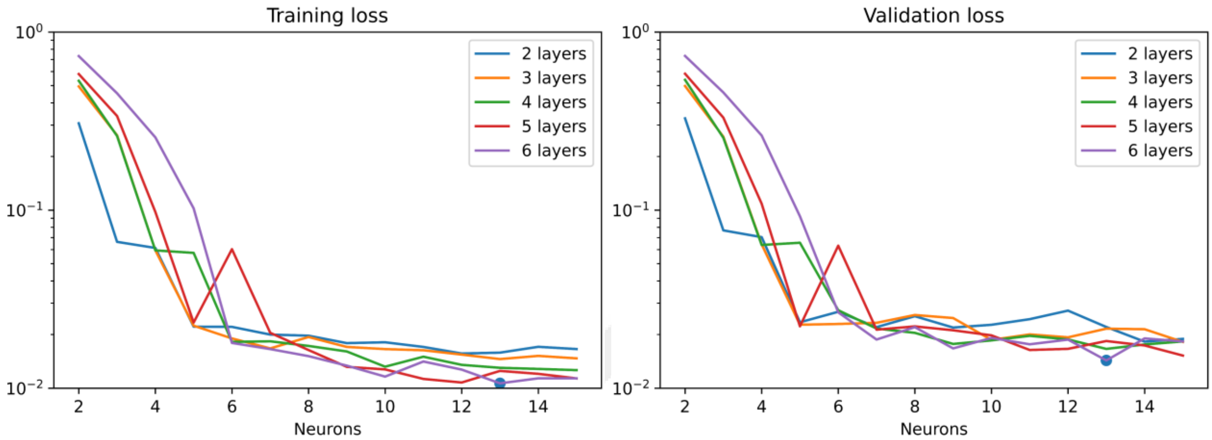

Figure 47: Selection of layers and neurons on ML model of Model 1

As the number of neurons increase, training loss and validation loss decrease, in general. Moreover, there is not much difference seen between using 3 or more layers in the network. Therefore, a small sized network is tried to be selected.

After the selection of 3 layers and 8 nodes at each layer, L2 regularisation search yielded following result on the same data sets in Figure 48.

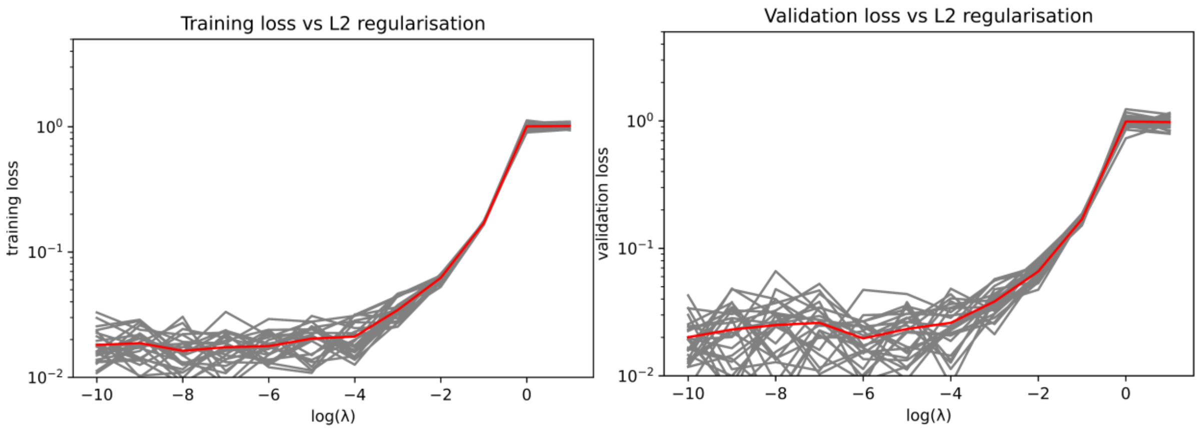

Figure 48: L2 regularisation term optimisation on ML model of Model 1

Grey lines show each of the 15 models' training and validation losses while the red line shows the average of the model losses. This shows that the regularisation of 10^{-6} results in the lowest validation loss on average. Therefore, $\lambda = 10^{-6}$ regularisation is chosen for this network.

3.2.2. PGML Model 1 – Predictions of lift coefficient and stall angles

In this section, predictions of lift coefficients for some airfoils using selected network structure of 3 layers and 8 nodes, and L_2 regularisation weight of 10^{-6} are shown.

Below example shows the results for NACA0012 airfoil which is an interpolation example for this study according to the training data set that is from thicknesses 10 to 15.

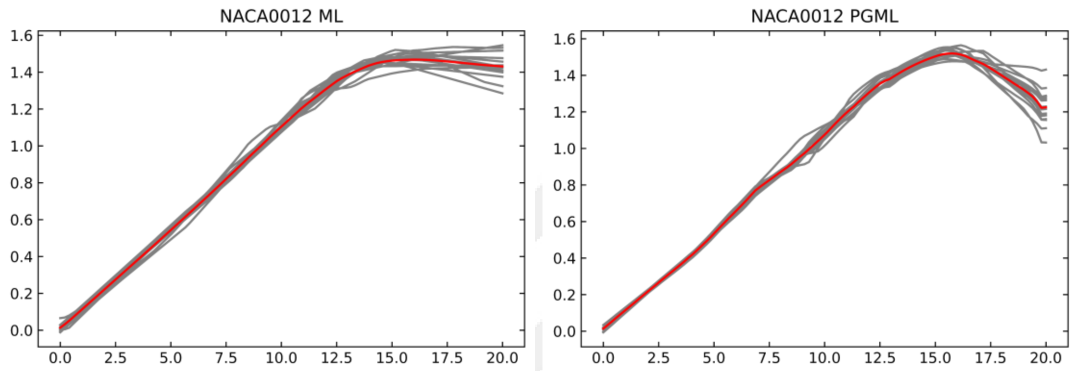

Figure 49: Lift coefficient predictions of NACA0012 from 15 same-networked ML (left) and PGML (right) models

Figure 49 shows the predictions of lift coefficients of 15 models and their averages for this interpolation example. Lift coefficient predictions are better for PGML network around the stall point because of the better shape of lift curve being bent down at the stall region.

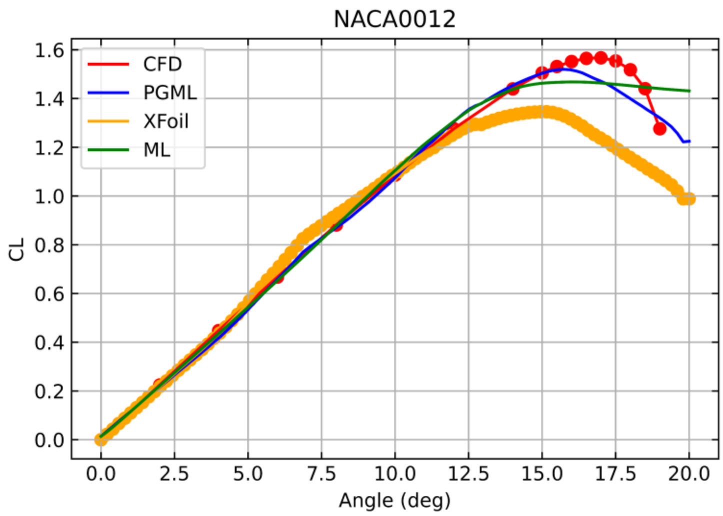

Figure 50: ML and PGML predictions of NACA0012 along with CFD and XFoil results

Figure 50 shows the average predictions of ML and PGML along with CFD results and XFoil predictions. It is seen that the PGML curve is affected by the XFoil input such that it bends down and captures the stalling behaviour for this airfoil.

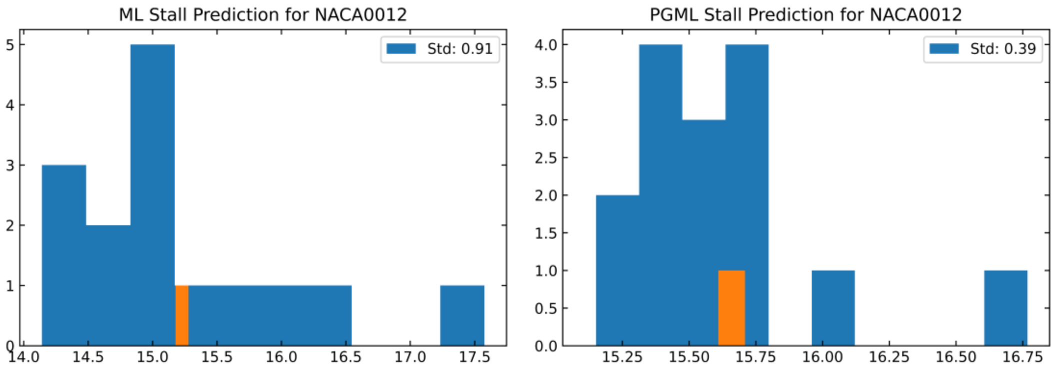

Figure 51: Histograms of stall predictions of NACA0012 from 15 same-networked ML (left) and PGML (right) models

Figure 51 shows the histograms of stall angles calculated using forward difference derivative of the resulting averaged lift coefficient curve for ML and PGML. The orange bins in the histograms represent the average of stall angles. Looking at the results, wider range of stall predictions of ML can be seen with the standard deviation of 0.91° than that of PGML with the 0.39° standard deviation.

Below is the extrapolation example for this model (Model 1).

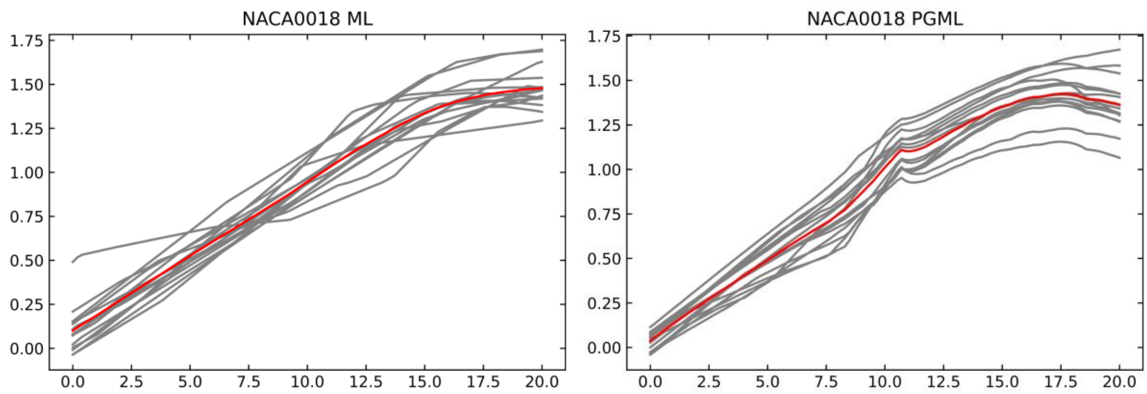

Figure 52: Lift coefficient predictions of NACA0018 from 15 same-networked ML (left) and PGML (right) models

Figure 52 shows the predictions of lift coefficients of 15 models and their averages for this extrapolation example. In both ML and PGML, the predictions are in a wider interval at each angle; however, still the lift coefficient predictions are better for PGML network around the stall point because there is a decreasing behaviour in the lift for PGML whereas ML curve does not really show the typical behaviour of airfoils.

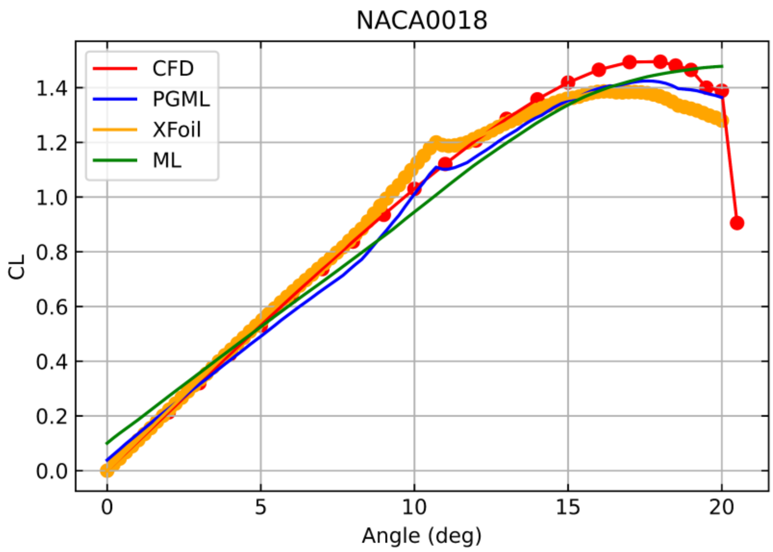

Figure 53: ML and PGML predictions of NACA0018 along with CFD and XFoil results

Similar to interpolation case, Figure 53 shows the average predictions of ML and PGML along with CFD results and XFoil predictions. The PGML curve gets the information from XFoil such that it not only bends down and captures the stalling behaviour but also it gets the small bump from XFoil at lower angles ~11° which does not seem physical.

This behaviour of XFoil does not represent the physical truth in airfoil lift. However, at this Re, Ma and corresponding angles of attack, this behaviour was investigated from XFoil. This shows that physical laws being interfered with the machine learning model should not include unphysical behaviours for the model to make more trusted predictions.

Another important observation from this example is that there is a small amount of lift predicted by ML model at zero angle of attack which should not exist since the airfoil is symmetric and there is no lift at zero angle in reality.

For this example, histograms are not compared since in ML predictions, there is no outcome of stall angle from almost all 15 models. On the other hand, for PGML prediction standard deviation of predicted stall increases to 0.79° which is larger with respect to the stall prediction for NACA0012 airfoil. This is as a result of the predictions being on unseen airfoil geometry.

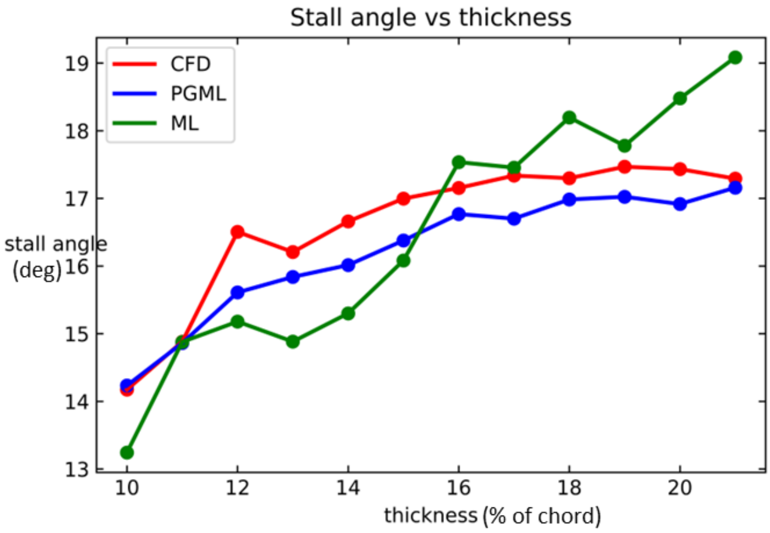

Figure 54: Predictions of stall angles against thickness of the symmetric airfoils using 3 layers and 8 nodes ANN model with stalls from CFD simulations

Figure 54 shows the stall predictions from ML and PGML models of the structure 3 layers and 8 nodes with L_2 regularisation weight of 10^{-6}. This figure summarises all the airfoil geometries (DS1 and DS2) in Table 2. Noting that the stall predictions until thickness of 15% is interpolation, at all thicknesses, PGML model shows a better performance in the sense that it consistently follows CFD results while the ML models overpredicts the stall in extrapolation region.

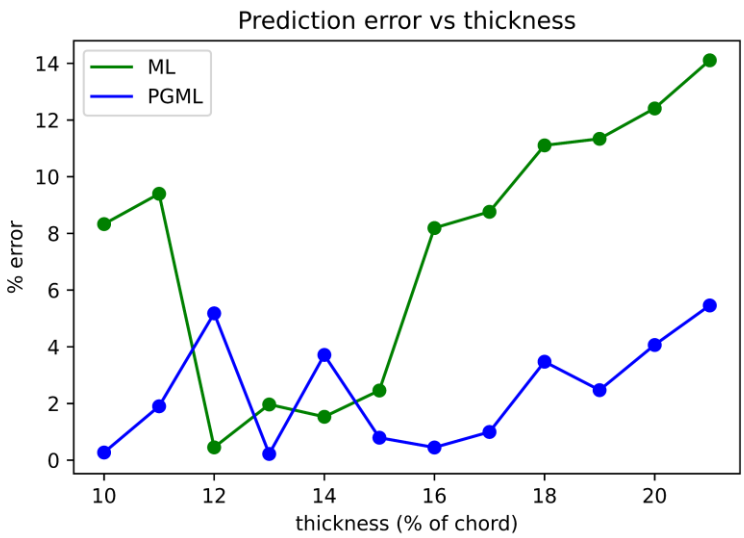

Figure 55: Absolute errors in stall predictions of ML and PGML models against thickness

As a summarising plot, Figure 55 shows the absolute percent error of stall predictions from the average of 15 models of ML and PGML networks with respect to CFD stalls. It is shown that the inclusion of physics in the neural network models decrease the maximum errors from 15% to 5% making the predictions more accurate.

3.2.3. PGML Model 2 – Hyperparameter optimisation

3.2.3.1. Hyperparameter optimisation using ML

Study on selection of layers and neurons on purely machine learning model is shown in Figure 56 with training loss (on left) and validation loss (on right).

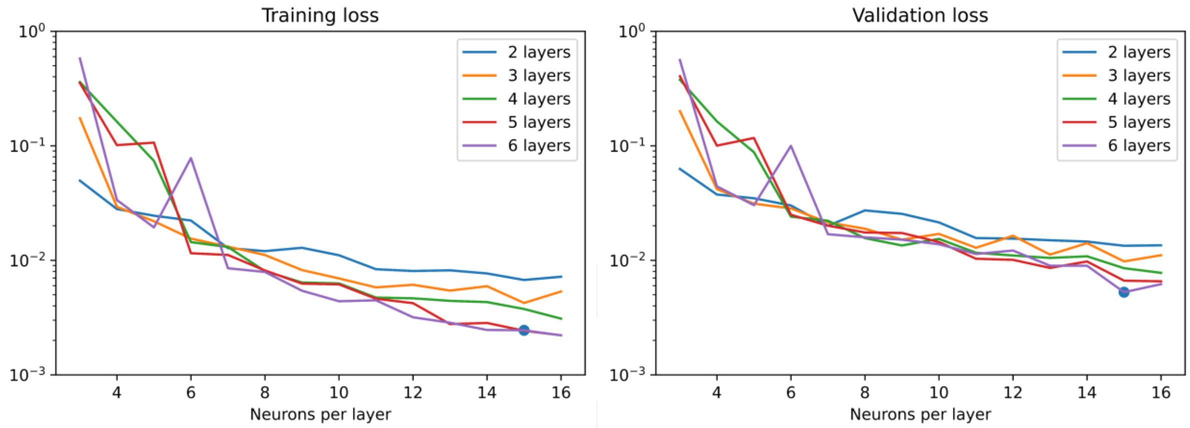

Figure 56: Selection of layers and neurons on **ML** model of Model 2 with validation on remaining 30% DS1

Noting that in this study, 70% of DS1 was used as training stage of pure ML, and 30% of DS1 as validation stage of ML, there is a decreasing trend in validation loss graph; however, there is not much difference between using 2, 3, 4, 5 and 6 layers in the network.

When the study is repeated using training data set of 70% of DS1 and validation data set of DS2, resulting losses are shown in Figure 57.

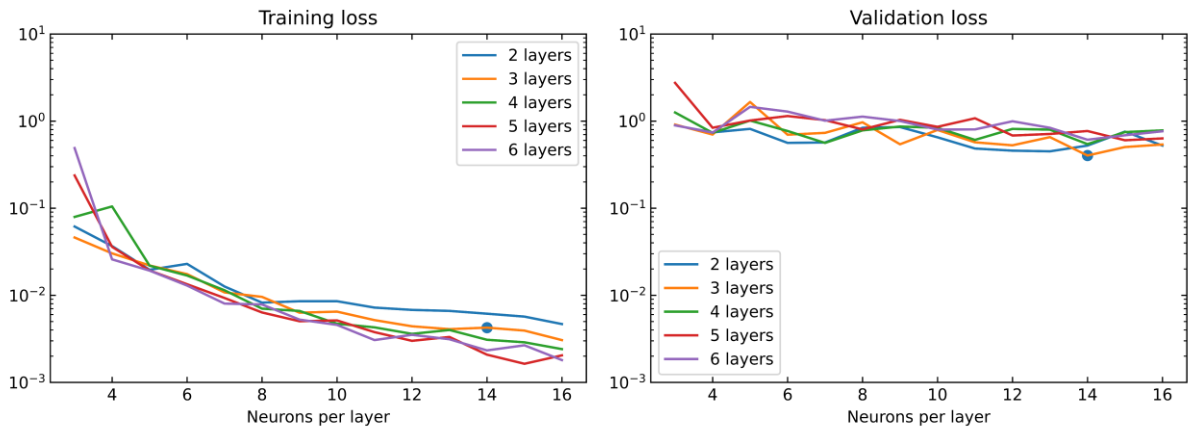

Figure 57: Grid search on layers and neurons on **ML** model of Model 2 with validation on DS2

This graph shows large values for the validation errors which gives an indication of switching to extrapolation performance, the pure ML creates large error values.

3.2.3.2. Hyperparameter optimisation using PGML

Figure 58 shows the selection of number of layers and number of neurons on PGML network with training loss (on left) and validation loss (on right). In this study, as the training data, DS1 is used and DS2 is used as the validation data.

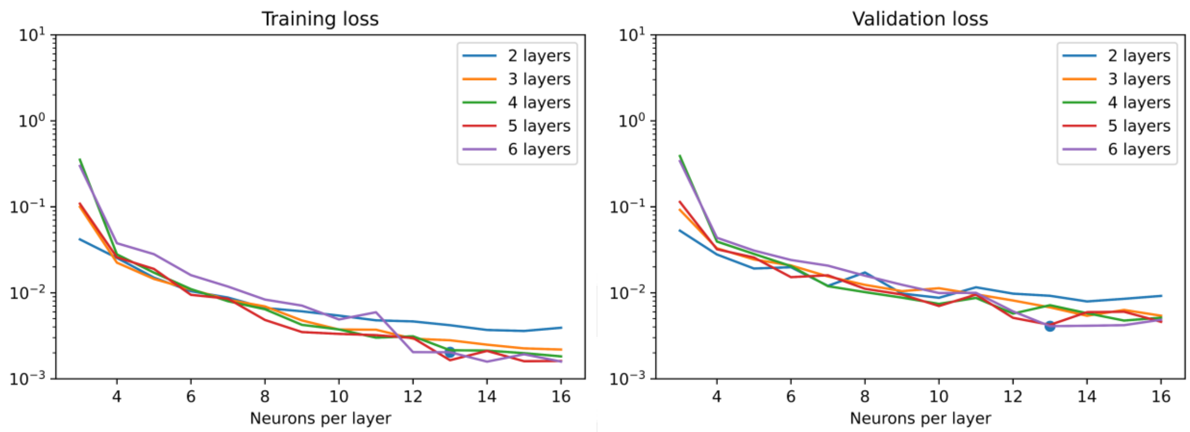

Figure 58: Grid search on layers and neuron on **PGML** model of Model 2 with validation on 30% of DS1

When the grid search on layers and neurons are done using PGML model, the magnitudes of validation errors decrease with respect to that of ML errors.

Looking at validation loss graph of Figure 58 as well as other validation losses in Figure 56 and Figure 57, 4 layers and 12 neurons at each hidden layer yield a good loss values. Hence, the network of 4 layers and 12 neurons has been selected and continued to seach L_2 regularisation term.

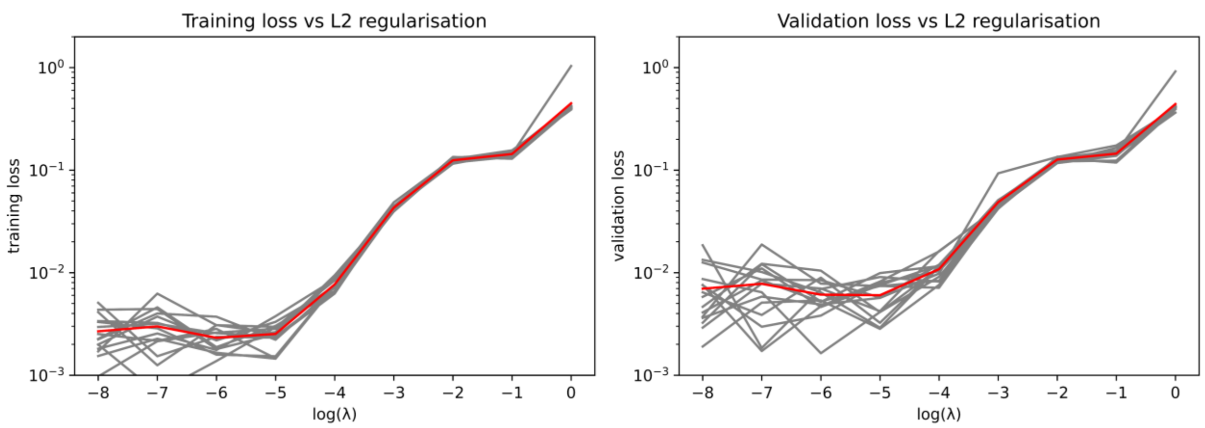

Figure 59: L_2 regularisation term optimisation on **PGML** model of Model 2

Figure 59 shows the training and validation losses with respect to logarithm of L_2 regularisation term. Since the purpose of regularisation is to reduce the overfitting, and increase the generalisability of the model, validation loss is checked against amount of regularisation. Therefore, L_2 regularisation weight of 10^{-5} is chosen for this network.

3.2.4. PGML Model 2 – Predictions of lift coefficient and stall angles

Below is shown an example of airfoil which was outside the training data set because it is outside the interval for thickness for the training set as shown in **Table 4**. The airfoil whose stall point is to be predicted is NACA3421.

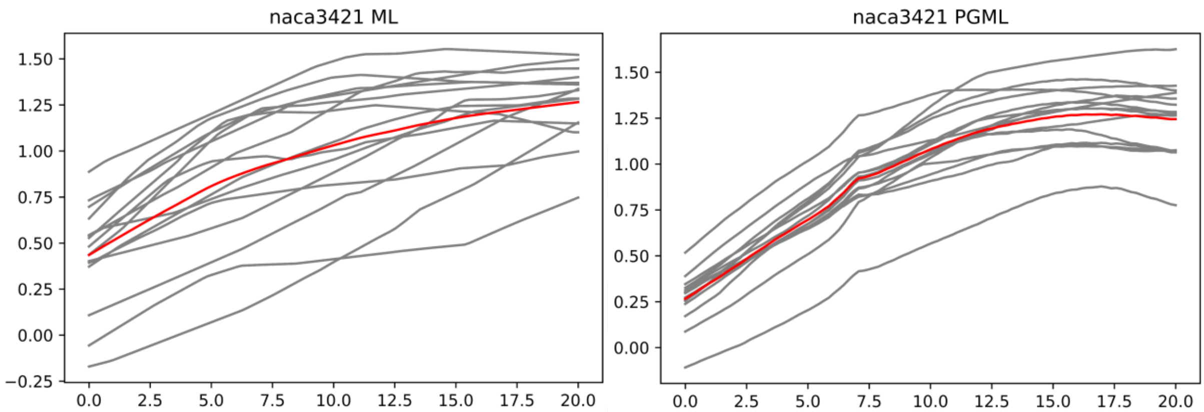

Figure 60: Lift coefficient predictions of NACA3421 from 15 same-networked ML (left) and PGML (right) models

Figure 60 shows the distribution of 15 models for ML and PGML networks. As it can be seen, physics input not only helps the lift curve follow physics but also it narrows the band of predictions making the PGML more confident in the predictions.

The results being in a wide range imply that the model learning has not been completed yet not because of the amount of the epochs but because there are few numbers of airfoils to train the model and make the model learn the pattern effectively. Also, increasing the number of models trained and whose results are averaged from 15 to a larger value would serve averaging for the better prediction.

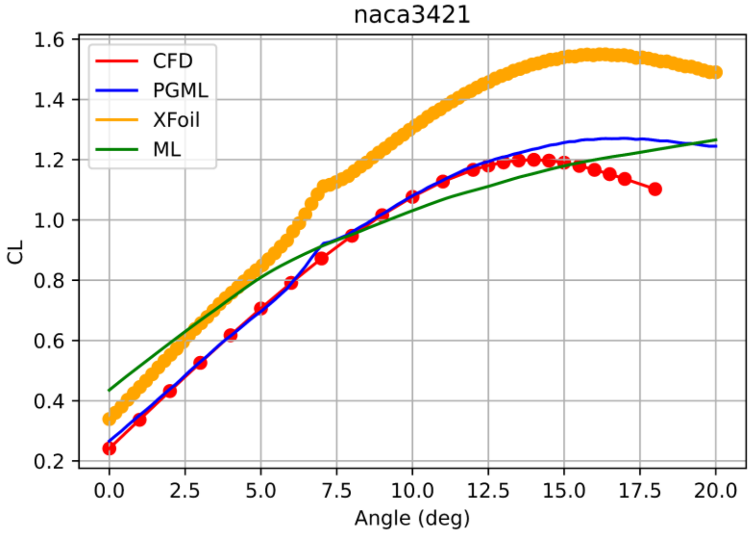

Figure 61: ML and PGML predictions of NACA3421 along with CFD and XFoil results

Figure 61 shows the ML and PGML averages with the CFD and XFoil results. The PGML model successfully captures the physics information in addition to learning from data set given as training. Since XFoil is just a panel method and not a CFD solver, there is a difference between how XFoil predicts the lift and what CFD simulations give. However, the PGML concept proves that the model can be enhanced greatly by including physics knowledge in the model and making it able to predict a stall value while ML model cannot predict the stall as it lacks physics information from the simplified method.

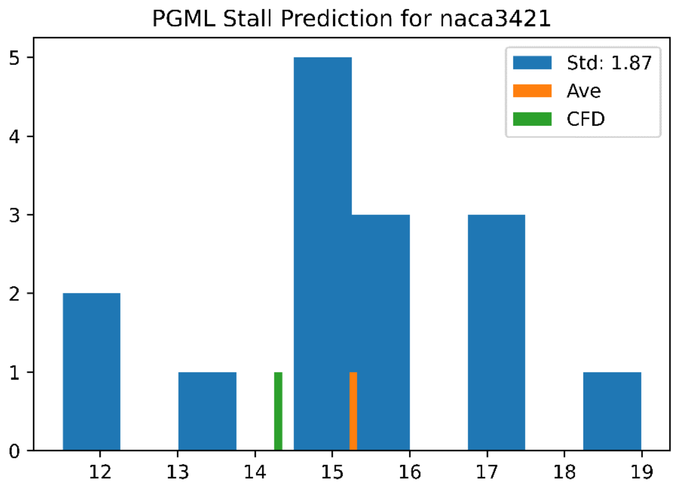

Figure 62: Histograms of stall predictions of NACA3421 from 15 same-networked PGML models

Figure 62 shows the histogram of stall predictions of PGML models for NACA3421 airfoil with the orange being the average of all models and green being the actual stall angle from CFD calculated from forward differencing first derivative of fitted curve. Histogram of ML models was not included here since most of the models cannot predict the stall. This shows the power of including physics into the network once again.

Other examples from DS2 are shown below.

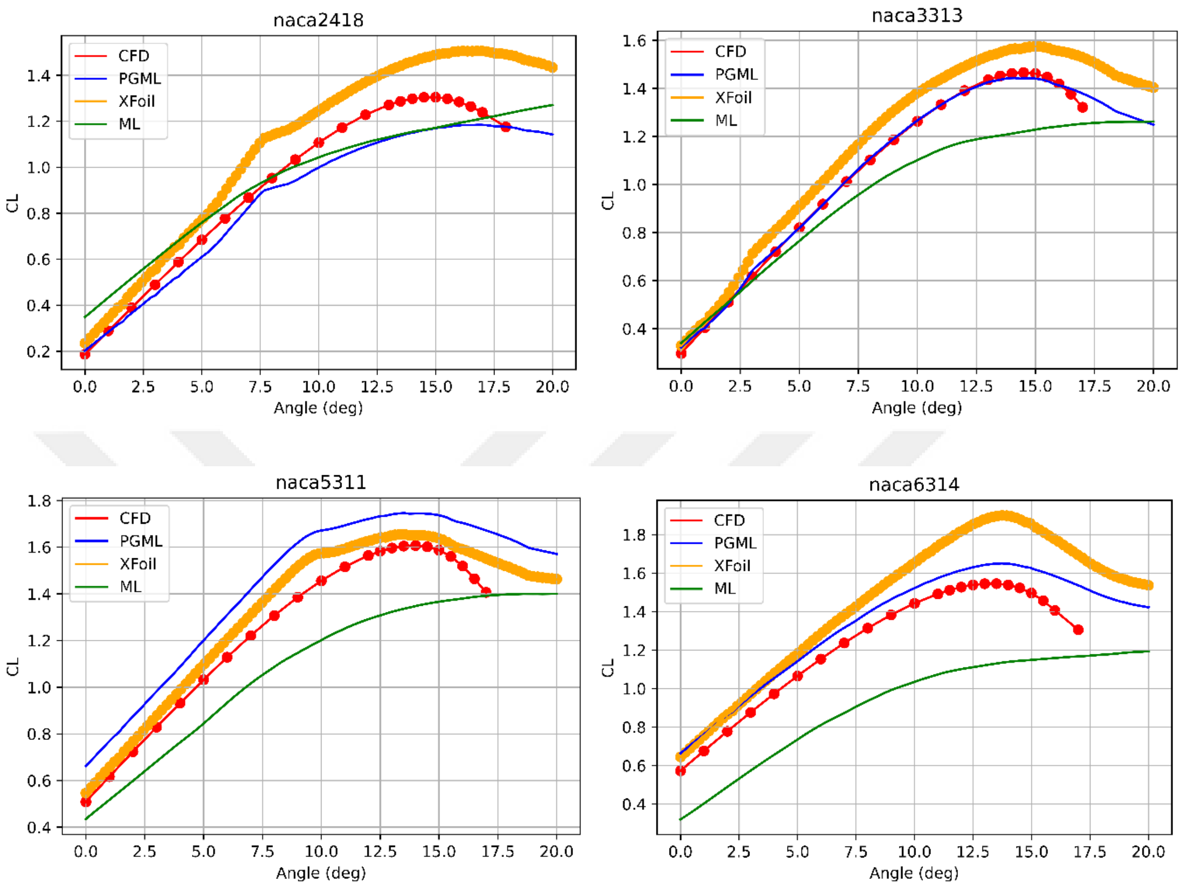

Figure 63: ML and PGML predictions of NACA2418, NACA3313, NACA5311, and NACA6314 airfoils along with CFD and XFoil results

Figure 63 shows the lift coefficient (hence the stall) predictions for NACA2418, NACA3313, NACA5311 and NACA6314. Among them NACA3313 is inside the training data set while the others are in testing data set DS2. The results show that ML models cannot predict the stall as PGML does with the help of physics input.

For some features of the problem to be learned, the models could be trained more precisely by higher resolution data set in a particular interval for a variable of interest. For example, in Figure 63, there is only 2% thickness difference between the airfoils that the bottom two graphs show and maximum camber is a little different, yet the predictions are overshooting both the CFD and the XFoil results for NACA5311 whereas the predictions for NACA6314 is between the CFD and XFoil results. This is an example of how the resolution of training data set could be improved around thicknesses of 10% to, say, 14% to improve pattern detection.

With the help of the machine learning models, it is possible to see the effect of camber even if there is not enough CFD data. In this case, knowing from Model 1 that PGML is more accurate, it is also possible to demonstrate the effect of camber by fixing the position of camber along the airfoil and airfoil thickness. Therefore, for a %13 thickness of airfoil with respect to its chord and position of its maximum camber at 40% along the airfoil chord, camber is being varied and following result was obtained as shown in Figure 64.

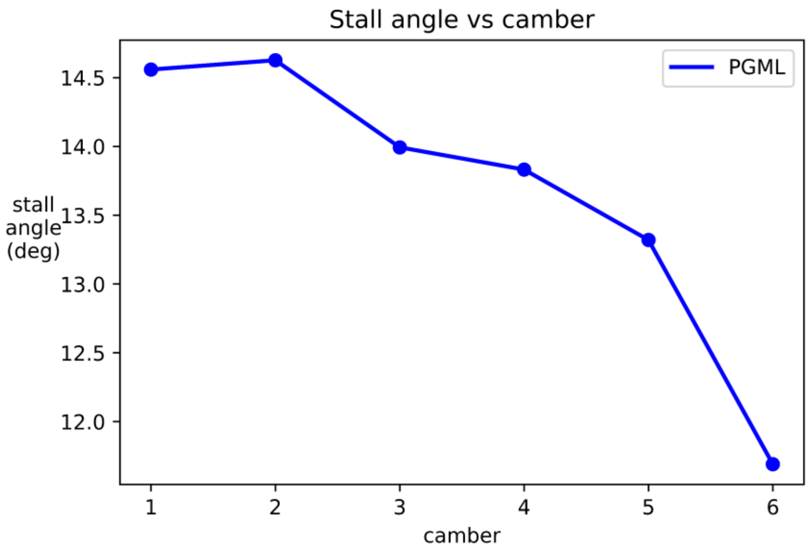

Figure 64: Effect of camber on stall angle of NACA four-digit airfoils with maximum camber location at 40% of airfoil chord from the leading edge and 13% thickness

Here, the ML model could not predict the stall angle for the corresponding airfoils and since for not all airfoils there is CFD results, only PGML predictions are presented. According to the figure, it is seen that as the maximum camber of airfoil increases stall as it is also known that as the airfoil gets more cambered, higher suction peak occurs closer to leading edge of airfoil, and the cambered geometry causes flow to separate close to trailing edge, and this situation is illustrated by the help of physics machine learning model without having CFD simulations for all the airfoils whereas the pure ML cannot predict the stall for those airfoils.

4. Conclusion and future work

The aim of this project was to combine the CFD methods with machine learning algorithms to create a predictive model for the stall of an airfoil. In this project, use of machine learning algorithms together with CFD simulations of airfoil was shown where the machine learning algorithm is specifically artificial neural networks and CFD simulations are RANS simulations.

In the first section of this report, modelling of airfoils with different settings of RANS modelling has been discussed. Also, their resulting graphs were always discussed with the available experimental data. A wide study has been performed to find the best combination of CFD configurations. This was completed on a single airfoil. What would make this part more comprehensive would be including more turbulence modelling approaches and repeating simulations on more airfoils instead of doing for one airfoil only.

Second step of the report includes machine learning algorithm and its use in learning of physics of the problem. At this step of the project, artificial neural networks (ANN) models were trained and tested since ANN has wide variety of design alternatives available with the selection of hyperparameters. Moreover, it allows concatenation of other variable at intermediate steps. Selection of hyperparameters, training with some data from full order data set and testing for the rest of the data have been performed.

In Model 1, half of the data set was used to train the ML and PGML models. Thin airfoils from the data set were used for training purposes. However, there was not quite large difference between the type of stalls for the airfoils, so this is an advantageous situation for both models that the behaviour of stall is similar. Both interpolation and extrapolation performances were discussed for the models. It has been shown that PGML model predictions are greatly improved by the inclusion of physics knowledge in the network. Furthermore, since the spread of the physics guided ML models are lower than that of ML models, as discussed in the results section, it is shown that PGML models yield lower uncertainty due to the existence of physics guidance. This also shows the capability of PGML model in lowering the uncertainty on physical problems.

In Model 2, similar to Model 1, a set of samples from the data set was selected and used for the training of ML and PGML models. In the light of determination of CFD configurations, the data set was generated from the airfoils produced by Latin hypercube sampling to reflect the real-like randomness of variables affecting airfoil geometry. It has been instantiated by the resulting plots that the physics guidance enhances the accuracy of neural network models. This model would have been trained more comprehensive if the data set had more samples as the model has two more inputs compared to Model 1.

Data driven models are excellent in the sense that there is no need to teach physics (implement physical laws) when there is enough data or samples are available. Identification of pattern through data analytics use of data only, has been Moreover, this project illustrates that predictions of those models can be greatly improved when physical inputs/laws are introduced for the machine learning model. As a result, this concept proves that ML models can be advanced by the simplified models even when the data is scarce.

Future work of this study would be providing a higher resolution sampling for the training of model to make sure every pattern when changing any variable of airfoil geometry is captured effectively by the model. Also, increasing the number of samples in the data set would improve both ML and PGML models to result in better predictions. For better averaged results, and making use of randomness, number of the same networked models could be increased. Moreover, in the future, parametrising the airfoil geometry differently would improve the generalisability and allow predictions for airfoils other than NACA four-digit airfoils. This would be done by making use of post processing variables like centre of pressure. Also, more parameters could be included in the network like drag coefficient and vortex shedding frequency as other outputs.

5. References

[1] Francisco Palacios, Thomas D. Economon, Aniket Aranake, Sean R. Copeland, Amrita K. Lonkar, Trent W. Lukaczyk, et al. Stanford University Unstructured (SU2): Analysis and Design Technology for Turbulent Flows. *52nd Aerospace Sciences Meeting.* Reston: American Institute of Aeronautics and Astronautics; 2014.

[2] Mouhsine SE, Oukassou K, Ichenial MM, Kharbouch B, Hajraoui A. Aerodynamics and structural analysis of wind turbine blade. *Procedia manufacturing.* 2018; 22 747-756. 10.1016/j.promfg.2018.03.107.

[3] ANDERSON. Fundamentals of aerodynamics. 6th ed. Columbus, OH: McGraw-Hill Education; 2016.

[4] Conformal Mapping [Internet]. Grc.nasa.gov. Retrieved from https://www.grc.nasa.gov/www/k-12/airplane/map.html

[5] Houghton, E.L., Carpenter, P.W., et. al., Chapter 6 - Thin Airfoil Theory. Aerodynamics for Engineering Students (Seventh Edition). 2017; Pages 391-447. https://doi.org/10.1016/B978-0-08-100194-3.00006-7

[6] 5. Vortex Panel Method [Internet]. Engapplets.vt.edu. Retrieved from https://www.engapplets.vt.edu/fluids/vpm/vpminfo.html#:~:text=The%20vortex%2Dpanel%20method%20is,because%20it%20is%20the%20simplest

[7] Ribando, R.J., Vortex Panel Method for a 2-D Airfoil. *University of Virginia.* 2008. Retrieved from https://www.robertribando.com/wp-content/uploads/2021/02/PanelMethod.pdf

[8] Douvi, E. C. & Margaris, D. P. *Aerodynamic Performance Investigation under the Influence of Heavy Rain of a NACA 0012 Airfoil for Wind Turbine Applications.* Praise Worthy Prize; 2012.

[9] Huthwaite, Peter. Machine Learning Notes. s.l. Imperial College London Department of Mechanical Engineering, 2021.

[10] Hagan M, Demuth H, Beale M, De Jesús O. Neural network design. [S. l.: s. n.]; 2016.

[11] G. Sanderson, Backpropagation calculus — Deep learning, chapter 4 (2017), 3Blue1Brown

[12] Ruder S. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:160904747. 2016

[13] Pawar, S., San, O., Aksoylu, B., Rasheed, A. & Kvamsdal, T. (2020) Physics guided machine learning using simplified theories. Physics of Fluids. 33 (1), 10.1063/5.0038929.

[14] Miyanawala TP, Jaiman RK. A Hybrid Data-Driven Deep Learning Technique for Fluid-Structure Interaction. *Volume 2: CFD and FSI* : American Society of Mechanical Engineers; Jun 09, 201910.1115/OMAE2019-95870.

[15] Rauseo, Marco. Python script that generates 2D mesh for an airfoil. Personal Communication. 2021.

[16] Grids - NACA 0012 Airfoil Case. Available from https://turbmodels.larc.nasa.gov/naca0012_grids.html

[17] 2D NACA 0012 Airfoil Validation. Available from https://turbmodels.larc.nasa.gov/naca0012_val.html

[18] Allmaras SR, Johnson FT, Spalart PR. *Modications and Clarications for the Implementation of the Spalart-Allmaras Turbulence Model.* 2012.

[19] Jameson A. Origins and Further Development of the Jameson–Schmidt–Turkel Scheme. AIAA Journal. 2017;55(5):1487-1510.

[20] Lennie M, Steenbuck J, Noack B, Paschereit C. Cartographing dynamic stall with machine learning. *Wind Energy Science*. 2020;5(2):819-838.

[21] Gudmundsson Snorri. *The Anatomy of the Airfoil.* : Elsevier; 2014. https://app.knovel.com/hotlink/pdf/rcid:kpGAADAMP8/id:kt00C86U42/general-aviation-aircraft/anatomy-airfoil?kpromoter=Summon.

[22] M. Drela, XFOIL: An Analysis and Design System for Low Reynolds NumberAirfoils, Conference on Low Reynolds Number Airfoil Aerodynamics,University of Notre Dame, June 1989

[23] NACA 4-Digit Numbering System - A Description [Internet]. Akiti.ca. Available from: http://www.akiti.ca/NACA4Bkgrnd.html

[24] Randomized Designs — pyDOE 0.3.6 documentation [Internet]. Pythonhosted.org. Available from: https://pythonhosted.org/pyDOE/randomized.html