

---

# **Using Artificial Intelligence for Estimation of Aerodynamic Parameters**

---

## **6301MECH Engineering Project Final Report**

**Name: Benjamin Ian George Gawith**  
**Student Number: 952752**  
**Supervisor: Ava Shahrokhi**  
**Programme: BEng Mechanical Engineering**  
**Date: 08/10/2024**

## Abstract

Accuracy in predicting aerodynamic performance is a crucial part of the design and optimisation process in aerospace, automotive and renewable energy applications. Traditional methods such as computational fluid dynamics (CFD) and wind tunnel testing provide high-fidelity results but rely on long time frames and heavy computational burdens to be utilised effectively. Recent progressions in machine learning (ML), particularly artificial neural networks (ANN), offer increasingly promising alternatives for rapid development in the preliminary stages of design due to the almost instantaneous predictive capabilities of well-trained models. This study investigates the use of deep learning techniques for predicting the lift coefficient as a function of angle of attack based on aerofoil geometries represented by the Class-Shape Transformation (CST) parameterisation method.

The aerofoils for the dataset were compiled from the UIUC Airfoil Database (Selig, 2010), and the aerodynamic data was XFOIL generated from simulations at a Reynolds number of 1,000,000 and gathered from [airfoiltools.com](http://airfoiltools.com) (*Airfoil Tools*, 2019). Fully connected multilayer perceptron, gated recurrent unit and long short-term memory networks were implemented using PyTorch and optimised using Optuna for hyperparameter tuning. A test split of the dataset comprising diverse aerofoil geometries unseen to the model was used for this performance evaluation, using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination ( $R^2$ ).

The NACA 0024, a well-represented symmetric aerofoil, demonstrated exceptional predictive accuracy with a MAE of 0.1466, RMSE of 0.2083, MSE of 0.0434, and an  $R^2$  score of 0.9993, indicating near-perfect agreement with XFOIL data. In contrast, the DRAGONFLY aerofoil, a more complex and underrepresented geometry in the dataset, exhibited a higher MAE of 0.5098, RMSE of 0.7784, MSE of 0.6060, and an  $R^2$  score of 0.9828, showing slight deviations from the ground truth but still maintaining strong correlation.

These results suggest that while the proposed model effectively captures aerodynamic trends across conventional aerofoils, performance degrades for aerofoils with more erratic lift characteristics or sparse representation in the training set. The study reinforces the potential of deep learning models in aerodynamic parameter estimation, offering a computationally efficient alternative to traditional CFD simulations.

## Acknowledgements

I would like to express my most sincere gratitude to my project supervisor Ava Shahrokhi for the invaluable guidance and support throughout the full duration of this project. The insights, expertise, and constructive advice given to me were instrumental to the completion of this work.

I would also like to thank the Liverpool John Moores University School of Engineering academic staff, for providing the foundational knowledge throughout my studies, and for ensuring a learning environment that encouraged both curiosity and innovation.

I am especially thankful to the hosts of the UIUC Airfoil Data Site and Airfoiltools.com, whose open-source contributions provided the foundation and generation of my dataset.

I am also grateful for the developers of open-source tools such as Python, PyTorch and XFOIL, which were crucial to implementing and assessing the models in this project.

I would like also to acknowledge the many research papers, books, and academic resources that have contributed to my learning. The insights and methodologies shared by the research community played a significant role in shaping the direction and depth of this project.

## Table of Contents

Abstract .....	ii
Acknowledgements.....	iii
Table of Figures .....	5
Nomenclature .....	6
Abbreviations .....	7
Introduction .....	8
1. Aims .....	8
2. Objectives.....	8
Literature Review .....	9
1. CFD and Experimental Approaches in Aerodynamics.....	9
2. ANN models for Aerodynamic performance .....	10
2.1 ANN-Based Surrogate Models .....	10
2.2 Physics-Informed Neural Networks (PINNs).....	10
3. Aerofoil Shape Representation and Parameterisation .....	11
3.1 Traditional Coordinate-based Method .....	11
3.2 Bezier Curves, PARSEC and Sobieczky Methods .....	11
3.3 Hybrid Methods (Bezier-PARSEC) .....	12
3.4 Class-Shape Transformation (CST) Method .....	12
4. Conclusion .....	12
Background Theory .....	13
1. Aerodynamics .....	13
1.1 Aerodynamic Parameters.....	14
1.2 Aerofoil Design and Applications .....	15
1.3 Class-Shape Transformation (CST) Method.....	16
2. Artificial Neural Networks .....	18
2.1 Architecture .....	19
2.2 Recurrent Neural Networks.....	21
2.3 Loss Functions Regularization and Evaluation Metrics .....	23
3. Conclusion .....	25
Technical Approach.....	25
1. Data Collection and Processing .....	25

2. Neural Network Development.....	29
3. Model Training and Evaluation .....	31
Results and Discussion .....	39
1. Evaluation of models with different loss functions .....	39
2. Final Model Evaluation and Analysis.....	42
Future Work.....	51
Conclusion.....	51
References .....	53
Appendix 1 .....	61
Appendix 2 .....	65

## Table of Figures

Figure 1: Aerofoil nomenclature including forces. (Anderson, 2017, p.20) .....	14
Figure 2: lift coefficient variation with angle of attack for an aerofoil (Anderson, 2017, p.329) .....	15
Figure 3: Aerofoil nomenclature (Veludurthi, 2020).....	16
Figure 4: Cross-section of a typical aerofoil (Kulfan, 2007, p.3).....	17
Figure 5: General structure of a MLP with one hidden layer (Zhang et al., 2024) .....	19
Figure 6: Optimum CST parameter script.....	27
Figure 7: Optimal model / loss function flowchart. ....	30
Figure 8: NACA 64A210 .....	32
Figure 9: NACA 64A410 .....	32
Figure 10: NACA 0024 .....	34
Figure 11: AH 93-W-300 .....	34
Figure 12: NACA 65-206 .....	35
Figure 13: Drela AG19 .....	35
Figure 14: Full model development process .....	38
Figure 15: MAE across models and loss function.....	40
Figure 16: MSE across different models and loss functions.....	40
Figure 17: RMSE across different models and loss functions.....	41
Figure 18: $R^2$ across different models and loss functions .....	41
Figure 19: Actual vs Predicted $C_L$ .....	43
Figure 20: Actual vs Predicted $\alpha$ .....	43
Figure 21: NACA 64A210 XFOIL vs Prediction .....	44
Figure 22: DRAGONFLY XFOIL vs Prediction .....	45
Figure 23: FX 67-K-150/17 XFOIL vs Prediction .....	45
Figure 24: NACA 0024 XFOIL vs Prediction .....	46
Figure 25: AH 93-W-300 XFOIL vs Prediction .....	46
Figure 26: NACA 65-206 XFOIL vs Prediction.....	47
Figure 27: AG19 XFOIL vs Prediction.....	47
Figure 28: NACA 2410 Predicted vs XFOIL vs CFD .....	50

## Nomenclature

Symbol	Description	Units
$\alpha$	Angle of attack	degrees (°)
$C_L$	Lift coefficient	–
$C_D$	Drag coefficient	–
$C_M$	Moment coefficient	–
$L$	Lift force	N
$D$	Drag force	N
$M$	Pitching moment	Nm
$S$	Aerofoil reference area	m <sup>2</sup>
$\rho$	Air density	kg/m <sup>3</sup>
$V$	Freestream velocity	m/s
$y$	True value (label)	–
$\hat{y}$	Predicted value (model output)	–
$N$	Number of samples or data points	–
$\psi$	Normalised chordwise position (x/c)	–
$\zeta$	Normalised surface coordinate (z/c)	–
$\zeta_T$	Trailing edge thickness (normalised)	–
$C(\psi)$	Class function in CST	–
$S(\psi)$	Shape function in CST	–
$A_i$	Bernstein polynomial coefficients	–
$K_i^N$	Binomial coefficient for CST	–
$\mathcal{L}_{MSE}$	Mean Squared Error loss	–
$\mathcal{L}_{mono}$	Monotonicity loss component	–
$\mathcal{L}_{smooth}$	Smoothness loss component	–
$\mathcal{L}_{zero}$	Zero-lift constraint loss	–
$\lambda_i$	Weight applied to loss components	–

## Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
ANN	Artificial Neural Network
CFD	Computational Fluid Dynamics
CST	Class-Shape Transformation
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
PI	Physics Informed
PINN	Physics Informed Neural Network
$R^2$	Coefficient of Determination
RANS	Reynolds-Averaged Navier–Stokes
Re	Reynolds Number
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SST	Shear Stress Transport (Turbulence Model)
UIUC	University of Illinois Urbana-Champaign

# Introduction

Aerodynamics is a crucial field of study in aerospace, automotive and renewable energy sectors, where the ability to understand and predict forces upon an aerodynamic body is essential to the optimisation in performance and efficiency. Traditional methods such as wind tunnel testing and computational fluid dynamics (CFD) are widely used in aerodynamic evaluations, providing precise and valuable results (Drela and Giles, 1987). However, these evaluation techniques have had limited accessibility and scalability in iterative design processes due to their computationally and financially expensive, and time consuming natures (Fukami, Fukagata and Taira, 2020; Vinuesa and Brunton, 2022). To address this, artificial intelligence (AI) and Machine learning (ML) techniques have been explored as alternative methods in evaluation and prediction (Fukami, Fukagata and Taira, 2020; Vinuesa and Brunton, 2022).

Artificial neural networks (ANNs) are one particular technique that has emerged as a promising tool in aerodynamic analysis. Being leveraged as a powerful surrogate modelling approach for predicting aerodynamic parameters such as lift ( $C_L$ ) and drag ( $C_D$ ) coefficients with great accuracy and efficiency due to their predictive capabilities (Brunton, Noack and Koumoutsakos, 2020; Brunton et al., 2021; Wang et al., 2024). ANNs are data-driven models that learn from large datasets of precomputed aerodynamic testing results. This allows for near-instantaneous predictions of aerodynamic performance once trained, significantly reducing computational costs (Brunton, Noack and Koumoutsakos, 2020; Wang et al., 2024).

This project explores the application of deep learning models for predicting  $C_L$  vs angle of attack ( $\alpha$ ) curves from aerofoil geometry, utilizing the Class-Shape Transformation (CST) parameterisation technique for shape representation. A dataset was generated using XFOIL for varying angles of attack, and different ANN architectures, including a Fully Connected Neural Network (FCNN), a Gated Recurrent Unit (GRU) model, and Long Short-Term Memory (LSTM) model. These were each tested and optimised using Optuna to determine the most effective architecture and hyperparameters (Kulfan, 2007).

The findings from this research could have significant implications for workflows in aerodynamic design across various industries, leading to reduced reliance on expensive simulations and enabling real-time aerodynamic assessments. Integrating AI into aerodynamic prediction has the potential to revolutionise the approach engineers take in design and optimisation of aerofoil-based systems in aviation, renewable energy, and automotive applications.

## 1. Aims

The primary aim of this project is the development and evaluation of ANN models capable of accurately predicting the lift curve for any given aerofoil based on CST parametrised geometry, seeking the optimal model for such task to reduce the dependency on traditional methods.

## 2. Objectives

To achieve this stated aim, the following objectives have been set:

1. Data Collection and Preprocessing
  - Gather aerofoil coordinates from UIUC Airfoil Data Site.



- Utilise CST parameterisation to represent aerofoil geometry efficiently (Kulfan and Bussoletti, 2006).
  - Gather XFOIL generated data for each aerofoil (*Airfoil Tools*, 2019).
2. ANN Development and Optimisation
    - Implement and compare FCNN, GRU, and LSTM models for aerofoil lift prediction.
    - Evaluate each model with different loss functions (MSE, MAE, Log-Cosh, Huber).
    - Utilise Optuna for hyperparameter tuning to determine for the most optimal model architecture and loss function configuration.
  3. Model Training and Evaluation
    - Select five representative aerofoils from the test set, covering a range of families and lift curve characteristics.
    - Conduct qualitative and quantitative analyses comparing model-predicted lift curves against XFOIL and CFD data.
    - Identify strengths and limitations of different architectures in predicting aerodynamic characteristics.

By achieving these objectives, this study aims to establish an optimised, data-driven approach to aerofoil performance prediction, contributing to the advancement of AI applications in aerodynamic analysis.

---

## Literature Review

This literature review exists to critically assess the role of ML-based techniques applied in aerodynamic performance predictions, evaluating limitations, potential improvements over traditional techniques, and their effectiveness. The structure of this review begins with addressing conventional analysis methods, followed by aerodynamic applications of ANNs, parameterisation techniques, and finally addressing drawbacks and solutions.

### 1. CFD and Experimental Approaches in Aerodynamics

This section highlights the computational and experimental methods of aerodynamic analysis.

CFD remains a cornerstone of aerodynamic analysis, solving selected governing equations in order to model fluid behaviour around a body (Woo Oh and Oh, 2010). CFD simulations, particularly high-fidelity Reynolds-Averaged Navier-Stokes (RANS) solvers, demand substantial computational power and are a high financial burden, this limits the availability to many users (Bonnet et al., 2022; Sharpe and Hansman, 2023). Similarly, wind tunnel testing, while offering real-world validation, is financially and time intensive, in turn limiting its practicality in rapid design iterations (Cattafesta, Bahr and Mathew, 2010).

Newer tools such as XFOIL or AIRFRANS provide detailed flow predictions at a much lower cost creating opportunity (Drela, 1989; Bonnet et al., 2022). XFOIL, a free command line based

software, was found to be 1000 times lower in computational cost, while producing a higher accuracy than RANS CFD (Morgado et al., 2016; Sharpe and Hansman, 2023). With the ever-growing advancement of AI, a paradigm shift to integrate the predictive capabilities of AI models into aerofoil design has been taken, with a focus on ML techniques like ANNs (Brunton, Noack and Koumoutsakos, 2020; Brunton et al., 2021; Vinuesa and Brunton, 2022).

Hybrid CFD-ML approaches have attempted to improve the efficiency while ensuring the accuracy stays at the upmost level. For example, Le and Yoon (2023) introduced a ML-accelerated CFD solver for the prediction of pressure drop in cyclone separators, where a correction term between an fine grid and coarse grid dataset is found, allowing for a major decrease in simulation run-time, and ensuring accuracy with errors from validation showing less than 5% (Le and Yoon, 2023).

## 2. ANN models for Aerodynamic performance

In this section, ANNs and versions of themselves are critically examined.

### 2.1 ANN-Based Surrogate Models

ANNs have displayed significant promise in replacement of traditional CFD-based analysis methods. Brunton, Noack and Koumoutsakos (2020) explored machine learning techniques for fluid mechanics, supporting the use of ANNs for aerofoil performance prediction, demonstrating that ANNs trained on CFD-generated datasets can predict aerodynamic coefficients with high accuracy. Similarly, Du, He and Martins (2021) implemented a ANN based parameterisation framework for aerofoil optimisation, which was found to significantly lessen computational costs while maintaining aerodynamic performance (Brunton, Noack and Koumoutsakos, 2020; Du, He and Martins, 2021).

However, the use of ANNs does not come without its challenges. A study on performance prediction of aerofoils through the use of an ANN showed that while capable of predicting aerodynamic coefficients, their performance is crucially dependant on the quality and diversity of the training dataset used (Tyrrell, 2023). A study on the use of a Multilayer Perceptron (MLP) predicting aerofoil lift coefficient values by Zhou, Zhang, and Chen (2025) found again that although successful in its predictions and accuracy, there was such a reliance on the quality of the training dataset. This issue also aligns with concerns raised by Wang et al who noted that training models on insufficiently diverse datasets can lead to poor extrapolation when applied to unseen geometries (Linse and Stengel, 1993; Wang et al., 2024; Zhou, Zhang and Chen, 2025).

### 2.2 Physics-Informed Neural Networks (PINNs)

One major drawback of purely data-driven ANNs is the lack of physical interpretability (Chen, Flores and Li, 2024). Raissi, Perdikaris and Karniadakis (2017) are accredited for the introduction of Physics Informed Neural Networks (PINNs), the implementation of physical laws into the training process of an ANN, where they demonstrated the integration of know partial differential equations (PDEs), such as those governing fluid flow, could be used as constraints in a loss function to guide a neural network in training.

Their work has been extended and applied across various areas of fluid mechanics such as a study by Karniadakis et al. (2021) where more complex Navier-Stokes equations were

incorporated into the loss function, enabling the recovery of hidden flow fields from sparse data. These implementations allow PINNs to not only predict aerodynamic quantities but also ensure consistency with physical behaviour across the flow domain. The inclusion of such physical constraints has been shown to improve both the generalisation of the model and its performance in data-scarce environments (Raissi, Perdikaris and Karniadakis, 2017; Raissi, Perdikaris and Karniadakis, 2019; Karniadakis et al., 2021).

However, the incorporation of differential operators and physical constraints significantly increases the computational burden of training PINNs. Calculating derivatives of network outputs with respect to inputs requires automatic differentiation and backpropagation through the physics terms, which often demands high-performance GPUs and substantial memory. Moreover, training PINNs often involves multi-objective optimisation, as the model must minimise both the traditional supervised loss (for example, MSE between predictions and known values) and the residual of the PDEs. This can lead to issues with gradient imbalance, making it challenging to converge without careful tuning of the weight terms in the loss function (Karpatne et al., 2017; Raymer, 2018; Cai et al., 2021).

### 3. Aerofoil Shape Representation and Parameterisation

Aerofoil parameterisation is crucial for defining shapes that meet specific performance criteria. Balancing complexity with flexibility provides effective techniques, enabling exploration of wide ranges of geometries with controllable numbers of parameters. Several methods exist, with their own particular advantages and disadvantages. Different aerofoil representation and parameterisation techniques are discussed in the following section.

#### 3.1 Traditional Coordinate-based Method

Conventional methods for aerofoil representation such as NACA equations and spline-based interpolation, provide structured approaches for defining aerofoil geometries since the late 1920s (Gray, 1948; Abbott and Von Doenhoff, 1959). These techniques do however lack flexibility for modern optimisation methods, additionally the variance of length and format of coordinate-based datasets creates challenges for standard NN architectures (Mukesh, Lingadurai and Selvakumar, 2014; Liu et al., 2017; Sharpe and Hansman, 2023).

#### 3.2 Bezier Curves, PARSEC and Sobieczky Methods

Bezier curves offer a flexible means of aerofoil representation. Smooth and continuous curves, which are advantageous for aerofoil geometry, are shaped using control points around the aerofoil. The number and location of these points greatly influences the control the designer has over the shape, sometimes leading to impractical shapes as the full control over the slope is not given through this method. Although the versatility of this method, increasing the volume of control points increases the complexity as a whole, creating opportunities for complication in the optimisation process and an increase in computational time (Ava Shahrokhi and Alireza, 2007; Salunke, Ahamad and Channiwal, 2014).

Parameterisation of Aerofoils by Splitting of Edge Curves (PARSEC) is a common approach for aerofoil geometry representation (Sobieczky, 1999). In which 11 design variables are defined and used to calculate polynomial coefficients to represent the geometry of an aerofoil (Mukesh,

Lingadurai and Selvakumar, 2014). This method lacks control over the trailing edge shape, where crucial flow characteristics can be observed. Sobieczky (1998) proposed a solution, in which an increase of the curvature near the trailing edge, potentially reducing the boundary layer decambering effect, while still allowing for enough flexibility in this section of the aerofoil, this led to a 12<sup>th</sup> design variable being incorporated. A potential for overlapping of the upper and lower surfaces of the aerofoil creates a big challenge with this method (Ava Shahrokhi and Alireza, 2007; Della Vecchia, Daniele and D'Amato, 2014).

### 3.3 Hybrid Methods (Bezier-PARSEC)

Combining the PARSEC approach with the geometric flexibility of Bezier curves, the Bezier-PARSEC method is created. Using the PARSEC variables as parameters to define four unique Bezier curves representing both the leading and trailing edge of the camber line and thickness distribution. Employment of second-order continuity in joining the leading and trailing edges ensure smooth transitions and precise control over aerofoil features. This hybrid technique enhances robustness, accelerating convergence in the optimisation process. Unlike standard the PARSEC method, this allows for detailed manipulation of the trailing edge, ensuring a more optimal shape is defined (Derksen and Rogalsky, 2010; Salunke, Ahamad and Channiwala, 2014; Tortora, Concilio and Pecora, 2023).

### 3.4 Class-Shape Transformation (CST) Method

First proposed by Brenda Kulfan and John Bussoletti in 2006, the CST method provides a systematic approach to aerofoil geometry representation. A combination of a class function, which defines the general shape class capturing the general shape of the aerofoil, and a shape function, which captures the more specific features. This allows for an efficient and compact parameterisation, making it possible to represent a large array of aerofoil shapes with a limited set of parameters, whilst still describing the more complex geometric design features. The robustness and simplicity of the CST method makes it a favourable option for use with NNs (Kulfan and Bussoletti, 2006; Kulfan, 2007; Bogue and Crist, 2008; Lane and Marshall, 2009; Sharpe and Hansman, 2023).

## 4. Conclusion

In summary, the main drawbacks found with the application of aerodynamic predictions with ANNs consists of dataset limitations, poor generalisation with overfitting or underfitting, high computational costs of in training and optimal methods for representation of aerofoil shapes when in use with ANNs. Showing that although ANNs present significant advantages in time and availability, challenges do remain. This project will use the CST method to keep complexity to a minimum, ensuring a constant shape which prevents shape errors during the implementation of the algorithm. A fully connected ANN will be used, as will implementation different ANN models, these models will be assessed in their predictive capabilities to find that which is most optimal in predicting lift curves for any aerofoil.

---

## Background Theory

This section exists to provide the reader with an understanding of relevant theoretical ideas necessary to this project.

### 1. Aerodynamics

Aerodynamics is the study of the motion of air the forces upon a body interacting with it. It is a fundamental aspect of aerospace engineering, vehicle design and renewable energy projects, particularly in the optimisation of aerofoil shapes for desired efficiency and performance. It is a necessity to be able to predict parameters such as the lift coefficient ( $C_L$ ) and drag coefficient ( $C_D$ ) in design of aerodynamic surfaces such as wind turbines or wings of an aircraft (Anderson, 2017).

The continuity equation, momentum equation and energy equation are the are the governing equations of aerodynamics, used to represent the fundamental principles of physics of the conservation of mass, conservation of momentum and the conservation of energy respectively, together known as the Navier-Stokes equations. These equations provide a mathematical framework for describing the motion of a fluid (Anderson, 2017; Shivamoggi, 2021).

For viscous, incompressible flows, which are most common in low-speed aerodynamics, the continuity equation ensures conservation of mass:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho V) = 0$$

Equation 1

Where  $\rho$  is the density of air,  $t$  is time and  $V$  is the velocity vector.  $\partial \rho / \partial t$  is the partial derivative of density with respect to time describes how the fluids density is affected over time and  $\nabla \cdot (\rho V)$  describes the divergence of mass flux, which accounts for the fluid flow. This equation states that a change in density within a control volume must be balanced by the mass flux entering or leaving the volume (Anderson, 2017).

$$\rho \left( \frac{\partial V}{\partial t} + V \cdot \nabla V \right) = -\nabla p + \mu \nabla^2 V + \rho g$$

Equation 2

In steady, inviscid flow, Bernoulli's equation provides a relationship between pressure and velocity. However, it does not take viscous effects into account, where the Navier-Stokes equations are then utilised (Ali, 2016; Shivamoggi, 2021). The Navier-Stokes equations, derived from Newton's second law applied to fluid elements, are used to describe the motion of fluids in a more comprehensive manner. The incompressible Navier-Stokes momentum equation in vector form is expressed as seen in equation 2, where  $\rho$  is the density of the fluid,  $V$  is the velocity vector,  $t$  is time,  $p$  is pressure,  $\mu$  is the dynamic viscosity and  $g$  is the acceleration due to gravity, balancing the rate of change of momentum with pressure gradients ( $\nabla p$ ), viscous forces ( $\nabla^2 V$ ) and external body forces such as gravity ( $\rho g$ ). Unlike Bernoulli's equation, the Navier-Stokes formulation captures the viscous shear stresses and flow separation phenomena that are critical

in understanding aerofoil performance, especially near stall conditions and in turbulent or boundary layer flows.

When considering aerofoils, these equations allow for accurate simulation of the pressure distribution over the upper and lower surfaces, enabling the prediction of lift and drag forces. Numerical solutions of these equations are vital in modern aerodynamic analysis.

Without these governing equations, computational and experimental aerodynamic studies that form the basis for simulations used that accurately predict the aerodynamic forces on an aerofoil would not be possible (Abbott and Von Doenhoff, 1959; Wilcox, 2006; Versteeg and Malalasekera, 2007; White, 2011; Anderson, 2017; Shivamoggi, 2021).

## 1.1 Aerodynamic Parameters

An understanding of aerodynamic parameters is vital in evaluation and optimisation of aerodynamic performance. Many parameters contribute to aerodynamic performance of an aerofoil, but for the purpose of this project, the lift coefficient value will be examined as the parameter of interest.

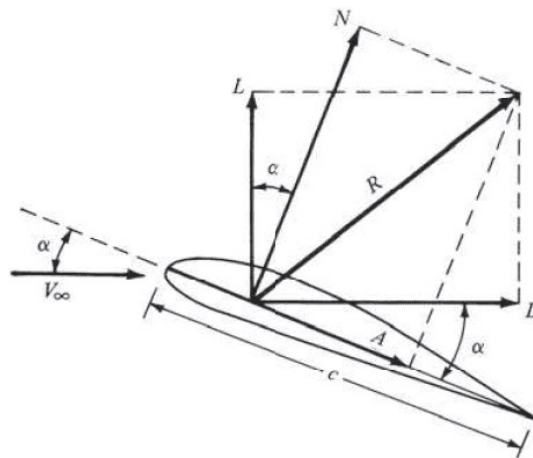


Figure 1: Aerofoil nomenclature including forces. (Anderson, 2017, p.20)

### 1.2.1 Lift Coefficient ( $C_L$ )

The lift is defined as the component of the resultant aerodynamic force on an aerofoil, perpendicular to the freestream velocity, shown as  $R$  on Figure 1. It relies on surface area, true airspeed, pressure, temperature, altitude, humidity, dynamic pressure, and the lift coefficient. The lift coefficient is a dimensionless parameter representing how much lift an aerofoil generates relative to the dynamic pressure and surface area; it is defined mathematically as:

$$C_L = \frac{L}{\frac{1}{2} \cdot \rho_{\infty} \cdot V_{\infty}^2 \cdot S}$$

Equation 3

Where  $C_L$  is the lift coefficient,  $\rho_\infty$  is the density of air ( $\text{kg/m}^3$ ),  $V_\infty$  is the freestream velocity ( $\text{m/s}$ ) and  $S$  is the reference area of the aerofoil ( $\text{m}^2$ ).

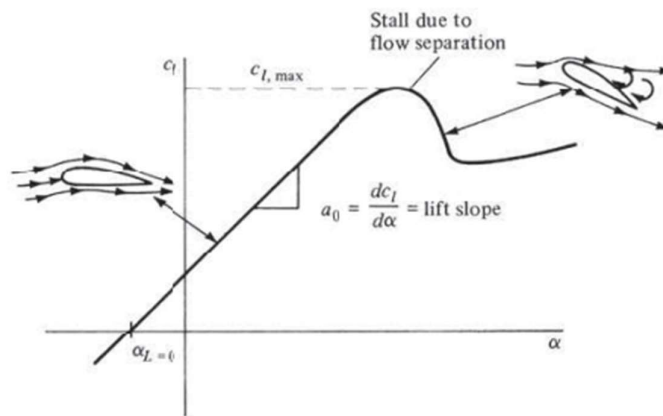


Figure 2: lift coefficient variation with angle of attack for an aerofoil (Anderson, 2017, p.329)

An almost linear trend is shown in the pre-stall region of a plot of  $C_L$  vs  $\alpha$  (stall slope), shown in Figure 2, this is due to smooth flow over the aerofoil, where the flow is attached for the majority of the surface, which can be seen to the left, with the diagram of streamlined flow over an aerofoil. To the right of the curve, a diagram of an aerofoil where the flow has separated from the top surface. This separated region contains flow that is recirculating, where some of this recirculating flow is moving in the opposite direction. This is common at higher angles of attack and the result of this separation is a vast decrease in lift, leading to non-linear relationships in flow which is where the aerofoil is said to have stalled.  $C_{Lmax}$  denotes the maximum lift coefficient, which is just before the stall point, this is a crucial point in aerodynamic analysis as the stalling speed is determined from it. Higher the value is, the lower the stalling speed (Abbott and Von Doenhoff, 1959; Anderson, 2017; Cherry and Hawk, 2024).

## 1.2 Aerofoil Design and Applications

An aerofoil is a cross-sectional representation of an aerodynamic surface, that is designed to generate lift when air flows over it, such as a spoiler on a high performance automobile or the rotor blades of a helicopter (Abbott and Von Doenhoff, 1959).

Various aspects of an aerofoil's geometry are:

- Chord ( $c$ ) - the horizontal, straight-line distance between the leading edge and trailing edge of an aerofoil.
- Leading edge ( $LE$ ) – the foremost point of an aerofoil, where the air flow first interacts with the aerofoil.
- Leading edge radius ( $r_{LE}$ ) – a measure of the curvature of the leading edge.
- Trailing edge ( $TE$ ) – the rear-most point of an aerofoil, where the airflow of the upper and lower surface recombines.
- Thickness ( $t$ ) - the maximum distance between the upper and lower surfaces of an aerofoil.

- Mean Camber Line – a line through the midpoint of the distance between the upper and lower surfaces.
- Camber ( $c_m$ ) - the curvature of the mean camber line.
- Aspect ratio ( $AR$ ) - the ratio of wingspan to mean chord length ( $AR = b/c$ ).
- Angle of attack ( $\alpha$ ) – the angle between the chord line and the relative wind, also known as the freestream velocity.

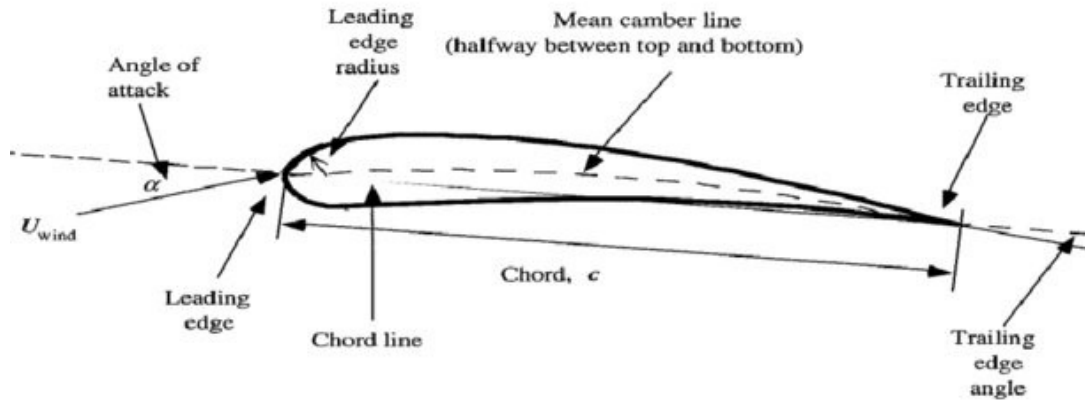


Figure 3: Aerofoil nomenclature (Veludurthi, 2020)

All these aspects directly affect the way in which the air surrounding the aerofoil flows, determining the key aerodynamic parameters such as lift or drag. Thin and symmetric aerofoils are often used in high-speed applications where minimal drag and high stability are priority, whereas, higher camber and thickness tend ratios generate more lift at lower speeds, making it an ideal choice in applications like unmanned aerial vehicles (UAVs) or gliders (Anderson, 2017; Raymer, 2018).

### 1.3 Class-Shape Transformation (CST) Method

The high dimensionality and lack of flexibility found in traditional coordinate-based methods led to the need for advanced techniques, such as the CST method. As outlined in section 3 of the literature review, introduced by Kulfan and Bussoletti, a particularly useful method in computational studies due to its ability to represent complex geometries mathematically as a limited set of parameters, thereby simplifying the input space, keeping complexity to a minimum, lowering computational costs and time requirements (Kulfan and Bussoletti, 2006; Sharpe and Hansman, 2023).



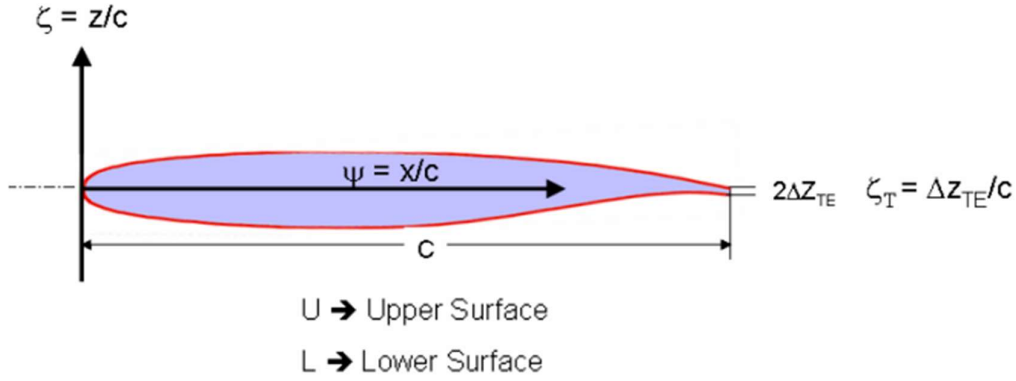


Figure 4: Cross-section of a typical aerofoil (Kulfan, 2007, p.3)

$$\psi = x/c \quad \zeta = z/c \quad \zeta_T = \Delta z_{TE}/c$$

The governing equation in CST parameterisation is expressed as:

$$\zeta(\psi) = C(\psi) \cdot S(\psi) + \psi \cdot \zeta_T + F(\psi)$$

Equation 4

Where  $\zeta(\psi)$  represents the aerofoil surface coordinate at a given chordwise position  $x$ ,  $C(\psi)$  is the class function which dictates the fundamental shape of the aerofoil,  $S(\psi)$  is the shape function, of which the finer details are captured for the aerofoil,  $\psi \cdot \zeta_T$  is the term providing control for the trailing edge thickness,  $F(\psi)$  are the additional terms, such as the trailing edge thickness and the leading edge modification parameter also developed by Kulfan in a later note (Kulfan, 2007; Kulfan, 2010).

The class function defines the general shape of the aerofoil, controlling the behaviour of the leading and trailing edges particularly. This is defined as:

$$C_{N_2}^{N_1}(\psi) = \psi^{N_1} [1 - \psi]^{N_2}$$

Equation 5

$N_1$  controls the curvature at the leading-edge. Typical subsonic aerofoils see  $N_1 = 0.5$ , as this produces a round leading edge.  $N_2$  which defines the curvature of the trailing edge.  $N_2 = 1$  is most common for typical aerofoils as it ensures a sharp trailing edge (Kulfan and Bussioletti, 2006; Lane and Marshall, 2010). The class function ensures that the shape of an aerofoil begins with a leading edge and follows smoothly on to the trailing edge without any discontinuities (Kulfan, 2007; Kulfan, 2008; Lane and Marshall, 2009).

While the class function dictates form of the aerofoil overall, the shape function is employed for refinement of features, introducing curvature, camber, and variations in thickness, typically expressed as a Bernstein polynomial expansion:

$$S_U(\psi) = \sum_{i=0}^N A_U \cdot S(\psi, i)$$

Equation 6

$$S_L(\psi) = \sum_{i=0}^{N_L} A_L \cdot S(\psi, i)$$

Equation 7

A is the set of shape coefficients representing the aerofoils specific geometry, N is the order of the Bernstein polynomial shown as S in this equation. This is defined as:

$$S(\psi, i) = K_i^N \cdot \psi^i \cdot (1 - \psi)^{N-i}$$

Equation 8:

N is chosen to balance model complexity and aerodynamic accuracy. K is the binomial coefficient, this is in direction relation to the order of the polynomial (Kulfan and Bussoletti, 2006). This is defined as:

$$K_i^N = \frac{N!}{i! (N - i)!}$$

Equation 9

Adjusting the shape coefficients ( $A_i$ ), the Shape function provides a refined geometry representation, capturing specific features.

To ensure aerofoils with a finite thickness at the trailing edge are accurately represented a further parameter is required, it is incorporated into the CST equation as  $\psi \cdot \zeta_T$ .  $\zeta_T$  is the trailing edge thickness parameter. It is particularly necessary as the shape at the trailing edge greatly influences flow separation and drag characteristics (Kulfan, 2007; Kulfan, 2008). Later Kulfan introduced another parameter, the leading-edge modification, aiming to further refine the representation of the leading edge radius and curvature (Kulfan, 2010). Boundary-layer development and stall characteristics are greatly impacted by the leading edge curvature, causing need for this (Sharpe and Hansman, 2023; Kulfan, n.d.).

The powerful CST method provides an efficient and compact way of parameterising an aerofoil and is found to be advantageous in computational applications of aerodynamics due to its ability to reduce dimensionality, ensuring a fixed length of data, in turn simplifying problems and making it ideal for use in deep learning problems (Jin et al., 2024).

## 2. Artificial Neural Networks

In the 1940s, studies that were described at the time as cybernetics by McCulloch and Pitts and Hebb, are some of the earliest in the field, this developed on to theories of biological learning

(McCulloch and Pitts, 1943; Hebb, 1949). The first perceptron was implemented in 1958, in turn allowing the training of a singular neuron (Rosenblatt, 1958). Around the 1980s back propagation created opportunity for training with one or two hidden layers (Rumelhart, Hinton and Williams, 1986). Deep learning, the way it is used today mostly started around the 2000s (Hinton, Osindero and Teh, 2006; Ranzato et al., 2007).

ANNs are class of machine learning models whose design was inspired by biological neural networks. Consisting of interconnected neurons processing and transforming data to find complex relationships and patterns resulting in a network that can produce accurate predictions in the field of which it is trained. They have been widely used in fields such as image processing, natural language processing and more related to this project, fluid dynamics, for aerodynamic predictions (Goodfellow, Bengio and Courville, 2016; Liu et al., 2017; Yu, Xie and Wang, 2019; Brunton, Noack and Koumoutsakos, 2020; Russell and Norvig, 2021; Vinuesa and Brunton, 2022).

An outline of the fundamental workings of modern ANNs, the loss functions and different evaluation metrics will be discussed in this section.

## 2.1 Architecture

The architecture of an ANN defines how neurons are structured and connected. One model this project utilises is a MLP, which is a feedforward ANN where data flows in one direction, from the input layer where no processing occurs, through one or more hidden layers that apply transformations using weighted sums and activation functions, to the output layer where the final prediction(s) are given (Popescu et al., 2009; Russell and Norvig, 2021; Zhang et al., 2024).

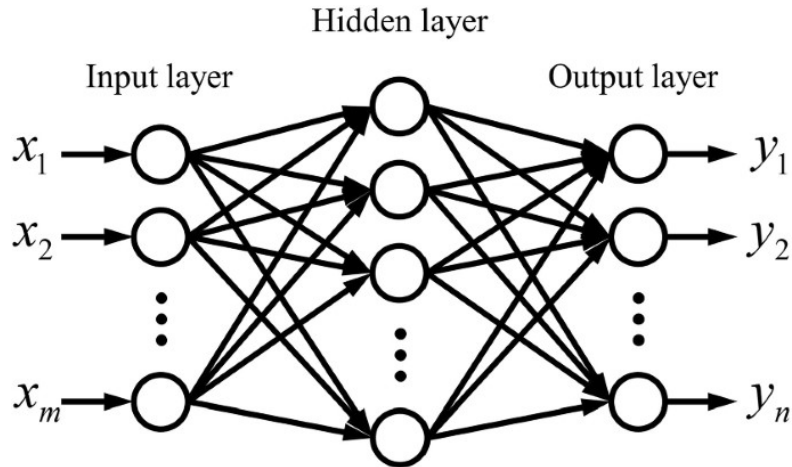


Figure 5: General structure of a MLP with one hidden layer (Zhang et al., 2024)

Each neuron is fully connected to each other neuron between each layer as shown in Figure 5. All neurons in the hidden layers and output layer have activation functions applied to introduce non-linearity, necessary for the model to learn complex relationships (Rosenblatt, 1958; LeCun, Bengio and Hinton, 2015; Goodfellow, Bengio and Courville, 2016). Each neuron in the hidden and output layers follows this mathematical operation:

$$h_i = \sum W_{ij} \cdot x + b_i$$

Equation 10

W represents the weights matrix of the neuron, x is the input vector from the previous layer, b is the bias term, and h is the pre-activation output. Each neuron receives inputs from every neuron in the previous layer, multiplies them by their respective weights and adds a bias term. This is then passed through a non-linear function, the activation function, giving the output (Apicella et al., 2021; Russell and Norvig, 2021). One example of activation function is the rectified linear unit (ReLU):

$$f(h_i) = \max(0, h_i)$$

Equation 11

which is the most commonly used due to its simplicity and capability of reducing the vanishing gradient problem. This is done by the input directly being output if it is positive, or a value of zero being output if negative (Nair and Hinton, 2010). ReLU was used in this project due to its efficiency in regression problems. Once passed through this activation function, the neurons output becomes input for the neurons in the next layer. This process of calculating weighted sums and applying activations is repeated throughout each layer, allowing the network to learn hidden and complex relationships from the input data, until the output is produced (LeCun, Bengio and Hinton, 2015; Russell and Norvig, 2021).

In the training process, the weights are adjusted to minimise error in predictions through a process called backpropagation. Each neuron's contribution to the error is calculated and the weights are adjusted accordingly to minimise the error on the next prediction. The gradient descent algorithm used in updating weights is:

$$W_{i+1} = W_i - lr \cdot \frac{\partial L}{\partial W_i}$$

Equation 12

The updated weight is the current weight minus the learning rate (*lr*), which controls how much the weights can change per iteration, multiplied by the gradient of the loss function with respect to the current weight. A learning rate that is too high can lead to the optimal solution being overshoot, but a learning rate of too small a value can lead to a slow convergence (Rumelhart, Hinton and Williams, 1986; Hecht-Nielsen, 1992; LeCun, Bengio and Hinton, 2015; Russell and Norvig, 2021).

In this project the Adaptive moment estimation (Adam) optimiser was used rather than standard gradient descent. The learning speed and stability of the model is improved by Adam due to its ability to adapt the learning rate for each individual parameter, which is done using momentum and squared gradient averages (Kingma, 2014).

## 2.2 Recurrent Neural Networks

Unlike traditional feedforward networks, Recurrent neural networks (RNNs) retain information from previous layers, proving them to be of immense value when the data is time-dependant or sequential in nature. It can be found that vanishing gradient problems occur when using these models, where gradients diminish exponentially as they propagate through time, putting a limit on their ability to learn long-range dependencies. In an attempt to address this, the Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) architectures were proposed. (Hochreiter and Schmidhuber, 1997; Medsker and Jain, 2001; Cho et al., 2014; Salehinejad et al., 2017).

### 3.2.1 Gated Recurrent Unit (GRU)

GRUs simplify traditional RNNs by introducing two key gating mechanisms: the reset gate and the update gate, which regulate the flow of information through the network. These gates allow GRUs to adaptively retain or forget information, improving both computational efficiency and the ability to model long-range dependencies (Cho et al., 2014).

Mathematical formulation of a GRU cell is as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

Equation 13

where  $r_t$  is the reset gate activation at time step  $t$ ,  $x_t$  is the current input,  $h_{t-1}$  is the previous hidden state, and  $W_r$ ,  $U_r$ , and  $b_r$  are learnable parameters. The sigmoid function  $\sigma$  ensures that the values remain between 0 and 1, allowing the reset gate to determine how much of the past information should be forgotten.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

Equation 14

where  $z_t$  is the update gate activation. The update gate controls how much of the previous hidden state is carried forward, effectively determining the balance between past and current information.

$$\hat{h}_t = \tanh(W_h x_t + U_h(r_t h_{t-1}) + b_h)$$

Equation 15

where  $\hat{h}$  represents the candidate hidden state, incorporating both the new input and the reset hidden state. The tanh activation function ensures the output remains between negative one and one, introducing non-linearity.

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \hat{h}_t$$

Equation 16

where  $h_t$  is the final hidden state at time  $t$ . The update gate decides the proportion of the new candidate state and the previous state that should be retained.

The computational efficiency of GRUs makes them particularly suitable for real-time aerodynamic predictions, as they require fewer parameters than LSTMs while maintaining comparable performance in sequence-based tasks. Wu et al. (2023) applied GRU models to aerofoil aerodynamic force prediction, demonstrating sixty-nine times faster computations compared to traditional CFD solvers, making GRUs a viable alternative for computationally expensive aerodynamic analyses.

Despite these advantages, GRUs are less effective than LSTMs for capturing extremely long-term dependencies, as they lack an explicit memory cell mechanism. As a result, GRUs are often preferred in applications where speed and efficiency are prioritized, whereas LSTMs are used when long-range dependencies are critical (Cho et al., 2014; Chung et al., 2014; Salehinejad et al., 2017).

### 3.2.2 Long Short-Term Memory (LSTM)

LSTM networks were introduced by Hochreiter and Schmidhuber (1997) to address the vanishing gradient problem, which limits the effectiveness of standard RNNs for long-range dependencies. Unlike GRUs, LSTMs incorporate an explicit memory cell that allows information to be stored, updated, and retrieved over extended sequences, making them highly effective for time-dependent aerodynamic modelling (Medsker and Jain, 2001).

For the forward pass of an LSTM, the equations influencing distinct aspects of information flow in LSTMs are as follows:

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

Equation 17

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

Equation 18

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

Equation 19

$i_t$  represents the input gate,  $f_t$  for the forget gate,  $o_t$  for the output gate,  $\sigma$  for the sigmoid function,  $w_x$  are the weights for the neurons of gate  $x$ ,  $h_{t-1}$  is the output of the LSTM block at previous timestamp  $t-1$ ,  $x_t$  is the input of the current timestamp and  $b_x$  are the biases for gate  $x$ .

The forget gate determines how much of the previous memory cell state  $C_{t-1}$  should be retained or discarded at time step  $t$ . The closer  $f_t$  is to zero, the higher the amount of past information disregarded. The input gate decides how much of the new input should be stored in the memory cell.

$$\hat{C}_t = \tanh(w_c[h_{t-1}, x_t] + b_c)$$

Equation 20

$$C_t = f_t \cdot c_{t-1} + i_t \cdot \hat{C}_t$$

Equation 21

$$h_t = o_t \cdot \tanh(c^t)$$

Equation 22

$\hat{C}_t$  is the candidate for cell state at the timestamp  $t$  capturing latest information,  $C_t$  is the memory cell state at timestamp  $t$  updated based on the forget and input gates enabling the LSTM to retain long-term dependencies effectively,  $\tanh$  is the hyperbolic tangent function and  $h_t$  is the updated hidden state. The output gate controls how much of the current memory cell state contributes to  $h_t$  which is then passed to the next time step. GRUs are computationally efficient, making them ideal for real-time aerodynamic predictions where speed is a priority. LSTMs are superior in handling long-range dependencies, making them better suited for complex, nonlinear aerodynamic flows.

For this project, both GRUs and LSTMs will be implemented to assess their ability to predict the  $C_L$ - $\alpha$  curve. The final model selection will be based on prediction accuracy, computational efficiency, and generalization capability across different aerofoil geometries (Hochreiter and Schmidhuber, 1997; Dey and Salem, 2017; Cho et al., 2022).

### 2.3 Loss Functions Regularization and Evaluation Metrics

Loss functions are used to quantify how far the prediction from the model is from the target value, allowing the model to learn from this and minimise the loss, therefore improving the ability to generalise to new data. This project is focused on a regression problem, due to the numeric nature of lift coefficients and angles of attack. Regression specific loss functions such as the mean squared error:

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i^{true} - y_i^{pred})^2$$

Equation 23

the average of the sum of the squared difference between the true value and the prediction. MSE is widely used for regression problems because it penalises outliers more heavily, making it highly effective for preventing deviations from the general trend of the lift curve; this metric is also used in the performance evaluation of the models (Hastie, 2009; Hastie, Tibshirani and Wainwright, 2015; Goodfellow, Bengio and Courville, 2016; Wang et al., 2022; Terven et al., 2023).

Huber loss is particularly useful in handling noisy datasets, this is due to the fact it can transition from quadratic to linear loss based on the error threshold  $\delta$ :

$$\mathcal{L}_{Huber} = \begin{cases} \frac{1}{2}(y_i^{true} - y_i^{pred})^2, & \text{if } |y_i^{true} - y_i^{pred}| \leq \delta \\ \delta \left( |y_i^{true} - y_i^{pred}| - \frac{1}{2}\delta \right), & \text{if } |y_i^{true} - y_i^{pred}| > \delta \end{cases}$$

Equation 24

Where  $\delta$  is a hyper parameter that controls this transition between quadratic and linear loss (Meyer, 2021). Due to the non-conformity to linear trends seen in the post stall region of a lift curve, aerodynamic data can be noisy, this causes the Huber loss to be of great value (Huber, 1992; Wang et al., 2022; Terven et al., 2023).

Another more robust loss function is the log-cosh loss:

$$\mathcal{L}_{log-cosh} = \sum_{i=1}^N \log (\cosh (y_i^{true} - y_i^{pred}))$$

Equation 25

behaving like MSE for smaller errors and becoming more linear for larger errors, in turn being less penalising to outliers, making it valuable as an alternative in training due to its stability (Saleh and Saleh, 2022; Wang et al., 2022; Terven et al., 2023).

To prevent overfitting and improve generalisation of the model, dropout regularisation can be utilised. Dropout is a stochastic regularisation technique that randomly deactivates neurons throughout the training process, this forces the model to learn redundant and robust feature representations (Srivastava et al., 2014).

In order to assess the performance of a model when looking at a regression-based evaluation metrics such as the coefficient of determination (R-Squared score,  $R^2$ ) (Wright, 1921), root mean squared error (RMSE) and the mean absolute error (MAE) are utilised. The  $R^2$  score indicates the proportion of variance and is calculated as:

$$R^2 = 1 - \frac{\sum (y_i^{true} - y_i^{pred})^2}{\sum (y_i^{true} - y_{mean})^2}$$

Equation 26

The sum of the squares of the residuals, the distance between the predicted and actual value, divided by the sum of the square of actual value minus the average value. The value can be in the range  $(-\infty, 1)$  with 1 being the maximum. The score can be negative, this would tell that the model does not follow the trend of the data. A higher R-squared score describes a closer correlation between the actual results and the prediction of the model, thus indicating a better generalisation (Chicco, Warrens and Jurman, 2021; Terven et al., 2023).

The mean absolute error is the mean of the absolute difference between the true values and the predicted values:



$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i^{true} - y_i^{pred}|$$

Equation 27

The closer the MAE is to zero, the better the generalisation is found to be. Best used if outliers do not represent crucial parts of the data, as in comparison to other metrics, such as MSE, outliers are penalised much lower (Chicco, Warrens and Jurman, 2021; Terven et al., 2023).

Finally, the RMSE is found from:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i^{true} - y_i^{pred})^2}$$

Equation 28

A higher weight is given to larger errors than the absolute error, just as in use of the MSE, but the RMSE does however take the prediction back to be of the same magnitude as the dataset, making it easier to interpret (Chicco, Warrens and Jurman, 2021; Terven et al., 2023).

Although there are many options for evaluation metrics, the ones outlined here are those that are used in this project: MSE, RMSE,  $R^2$ , and MAE. Chosen to thoroughly evaluate the model's performance.

### 3. Conclusion

This section outlined key concepts essential to this project, covering aerodynamics, CST parameterisation and ANNs. These theoretical foundations supported the development of the ANN based aerodynamic prediction model.

## Technical Approach

A full outline of the approach taken for the completion of this project is detailed in this section. The development and analysis were performed using Python programming language. For further details on the code, see Appendix 2.

### 1. Data Collection and Processing

As outlined in previous sections, the robustness and accuracy of a ANNs predictions is significantly dependent on the quality and representation of its training data. The UIUC Airfoil Database was the source of aerofoil coordinate files used in generation of the training dataset. From coordinates, the geometries were parameterised using the CST method, this compact

representation was used as the aerofoil features. 48 XFOIL generated  $C_L$  values at 48 angles of attack were used as the targets for the ANN.

Due to being an open-source, extensive collection of both experimental and computational aerofoil geometries, the UIUC Airfoil Database was chosen for the source of aerofoil coordinate files necessary for the fabrication of the training dataset. Hosting aerofoils used in aerospace, renewable energy and automotive applications for example, this database ensures diversity, aiming to give fair representation of different aerofoil types in turn creating opportunity for better generalisation across various aerodynamic designs. 1568 aerofoils were combined into the dataset in total. These raw coordinate files pose issues in machine learning due to their inconsistent lengths and increased complexity, as discussed in section 3.1 of the literature review.

As an example, the A18 aerofoil coordinate file from the UIUC database is forty-one (x, y) pairs in length, whereas the coordinate file for the Davis aerofoil is thirty-four pairs in length. The inconsistency of and the length itself of the coordinate files leads to the necessity of representation in a more consistent and simpler manner (Selig, 2010).

The CST method, outlined in 1.3 of the Background Theory, has been chosen to represent these coordinates mathematically in a set of parameters of a consistent length. To implement this with Python programming language, the function 'get\_kulfan\_parameters' found in the Aerosandbox library (Sharpe and Hansman, 2021) was utilised. The output of this function is a Python dictionary containing N upper weights and N lower weights where N is the number of Bernstein polynomials specified when calling the function, the trailing edge thickness, and the leading-edge weight. These are the parameters used as features to train the neural network.

The website [airfoiltools.com](http://airfoiltools.com) hosts an extensive database of aerofoil data. For this project, the data from XFOIL runs for each aerofoil were of particular interest. Due to the extensive number of aerofoils in the dataset, a simple web scraping script was implemented. The script fetches the angle of attack and lift coefficient at 48 points equally spaced across a csv file containing XFOIL data for each aerofoil in the dataset, across pre-stall, stall and post-stall regions, allowing for a detailed lift curve that captures the trend of  $C_L$  as a function of  $\alpha$ , without unnecessary complexity within the model's architecture (*Airfoil Tools*, 2019).

Optuna is an open-source library available for Python, which is commonly used for optimising hyperparameters in machine learning. By defining an objective function, over a set number of trials, Optuna finds the optimal values through systematically suggesting values within the range suggested. As a way of ensuring an unbiased method for finding the optimal number of upper and lower weights for CST shape representation, this technique was employed. Given that different architectures respond differently to each input representation, Optuna provided an effective mechanism to combat this (Akiba et al., 2019).

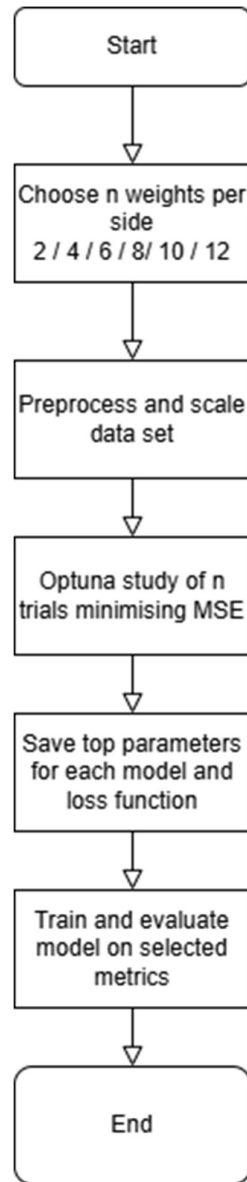


Figure 6: Optimum CST parameter script.

Figure 6 demonstrates the logic for the script used to find the optimal CST parameters. Several configurations of CST parameters were tested using Optuna to determine the optimal number of weights for the upper and lower surfaces. Training datasets with each of 2, 4, 6, 8, 10 and 12 weights per side were created. A fully connected ANN was trained on each of these datasets through an Optuna study aiming to minimise the MSE. The performance was then evaluated for the top results for each dataset using MSE, RMSE,  $R^2$  and MAE, as shown in Table 1.

Total CST Parameters	MSE	RMSE	R2	MAE
6	1.1578	0.8263	0.6935	0.6084
10	0.9645	0.7593	0.7354	0.5451
14	0.8782	0.7253	0.7532	0.5240
18	1.5167	0.9533	0.5332	0.5760
22	1.0398	0.7894	0.7079	0.5814
26	1.4239	0.9202	0.6177	0.6643

Table 1: CST Optimisation Evaluated Metrics

Using 6 weights for each of the upper and lower surfaces, a leading-edge weight and a trailing-edge thickness, 14 CST parameters in total, achieved the lowest MSE of 0.8782, RMSE of 0.7253, MAE of 0.5240 and highest  $R^2$  of 0.7532. These results indicate that 14 parameters manage to effectively capture the aerofoil geometry without unnecessary complexity. Fewer weights (6 or 10) produced worse results, suggesting insufficient geometric representation, whereas the higher number of weights (18, 22, 26) also exhibited worse results most likely due to overfitting. Due to outperformance of all other dataset sizes, 14 CST parameters were chosen as the optimal, balancing geometric accuracy and model generalisation most effectively.

After establishing this dataset, it was split into features and labels, which consist of the CST parameters and the  $C_L$  and  $\alpha$  values respectively. The features were scaled by removing the mean from the sample value and scaling to the unit variance by dividing by the standard deviation. This is called Z-score scaling and is mathematically formulated as show in equation 29. This method preserves the shape of the distribution and can manage outliers well, as they will be scaled to be several standard deviations away from the mean.

$$X_{scaled} = \frac{X - \mu}{\sigma}$$

Equation 29

$$X_{scaled} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Equation 30

For the labels, the Min-max algorithm was used which is shown in equation 30 where the current sample minus the minimum value, divided by the maximum minus the minimum gives the scaled sample value. This method ensures that the scaled value is between 0 and 1 which is most useful for normalising features with different units or scales and avoiding feature dominance, where one feature has a much larger range than others, improving the stability and performance of the algorithm. Due to the difference in scale of the lift coefficients and the angles of attack the min-max algorithm best suited the label data (Patro and Sahu, 2015; Henderi, Wahyuningsih and Rahwanto, 2021).

## 2. Neural Network Development

The ANN models proposed to be assessed in this project are a standard feed-forward fully connected ANN, a LSTM RNN and a GRU RNN. These were implemented using PyTorch, a popular deep learning framework, widely used due to its flexibility and efficiency for training complex architectures whilst remaining computationally efficient (Paszke et al., 2019). In aiming to determine the model with the greatest aerodynamic predictive capabilities, three architectures, loss functions and data preparation methods have been explored. Optuna was leveraged to automate the hyperparameter optimisation process, ensuring the optimal configuration was found. The metric used for this optimisation was MSE due to the magnitude of the effect of major outliers when trying to ensure the general trend of a lift curve is captured. The model's performance was then evaluated on the MSE, MAE, RMSE and  $R^2$ .

The loss functions being assessed are as detailed in section 3.3 of the background theory, the Huber loss, LogCosh loss and the MSE loss. With an additional custom physics informed loss function (PI) aiming to enforce physical attributes into the training process of the neural network.

$$\mathcal{L}_{mono} = \frac{1}{N-1} \sum_{i=1}^{N-1} \max(0, s_{\min} - \frac{dC_L}{d\alpha})^2$$

Equation 31

This custom loss function is built from four components, with the MSE loss (equation 23) as the base of the loss, ensuring that predicted values are as close as possible to the true values. Then three additional physical constraints are introduced in guiding the learning process.

Firstly, within the pre-stall region, that the  $C_L$  should increase with  $\alpha$  and if this condition is violated, a penalty is applied, mathematically shown in equation 31, this term enforces that the discrete slope of the  $C_L$  curve with respect to  $\alpha$  remains above a defined minimum threshold, 0.1 as standard. This ensures the lift slope is non-negative and consistent in the linear region.

$$\mathcal{L}_{smoot} = \frac{1}{N-2} \sum_{i=1}^{N-2} \left( \frac{d^2 C_L}{d\alpha^2} \right)^2$$

Equation 32

The next constraint, formulated as in equation 32, is a smoothness loss where large second derivatives are penalised aiming to reduce the model predicting unrealistic jumps or oscillations throughout the lift curve.

$$\mathcal{L}_{zero} = (C_L(\alpha \approx 0))^2$$

Equation 33

Finally, a zero-lift constraint shown in equation 33. Physically, many symmetric or lightly cambered aerofoils generate roughly zero lift at  $\alpha = 0^\circ$ . The predicted  $C_L$  at the angle closest to zero is penalised if it deviates significantly from zero.

$\lambda_i$  represents hyperparameters where the weight of a specific loss can be set. The final loss is calculated as shown in equation 34. Though not all lift curves follow such a linear or smooth trend, it is more common than not, leading to the implementation of these features (Dawson et al., 2016).  $\lambda_{mono} = 0.05$ ,  $\lambda_{smooth} = 0.01$ ,  $\lambda_{zero} = 0.05$ .

$$\mathcal{L}_{PI} = \mathcal{L}_{MSE} + \lambda_{mono} \cdot \mathcal{L}_{mono} + \lambda_{smooth} \cdot \mathcal{L}_{smooth} + \lambda_{zero} \cdot \mathcal{L}_{zero}$$

Equation 34

This PI loss function aims to enforce adherence to physical aerodynamic laws, leading to a better generalisation in predictions.

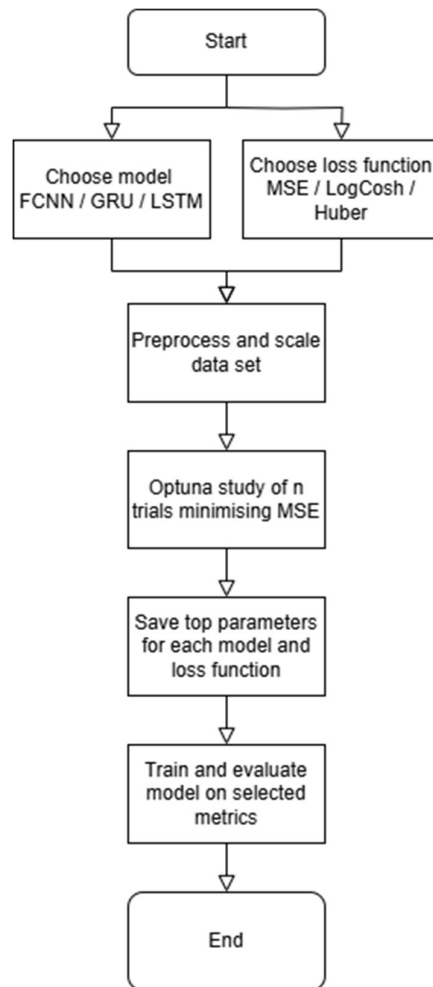


Figure 7: Optimal model / loss function flowchart.

The chart in Figure 7 depicts the logic of the script used to find the model and loss function best suited for predicting the lift curve. The value of n in this study was 250, this is the number of trials per model, per loss function, aiming to optimise the hyperparameters.

### 3. Model Training and Evaluation

Once the optimal model and loss function combination had been established, further tuning was applied to the model, optimising the hyperparameters further. The final configuration was then assessed. The mathematical portion of this assessment was performed using the same evaluation metrics as before,  $R^2$ , MSE, RMSE, MAE. The dataset is split into an 80/20 train/test split, ensuring a fair and robust evaluation. The 80% split for training ensured a large and diverse subset of aerofoils were included for training, allowing the other 20% to be used reserved for final performance assessment and validation.

This final evaluation is performed on the test section of the dataset, analysing the model's performance in a fair and unbiased manner. A selection of five unseen aerofoils were chosen from the test split of the dataset. The selection process was designed to ensure a diverse representation of aerofoil types, covering both common and less represented aerofoils, as well as those exhibiting complex aerodynamic behaviours, allowing for a comprehensive evaluation of the model's generalisation ability across a different aerodynamic shape. A full table describing the volume of each aerofoil from each family in each dataset can be found in Appendix 1 under Table 3.

As seen in Table 3 in Appendix 1, the NACA family hosts 124 aerofoils in the training dataset, with 24 in the test dataset. The chosen aerofoil was the NACA64A210, due to this prominent level of representation. This aerofoil is a member of the heavily studied NACA 6-series that was designed to maximise laminar flow over a significant portion of the aerofoil surface, thereby reducing drag. The NACA64A210, specifically, has been used in high-speed applications due to its favourable aerodynamic characteristics, such as maintaining lift efficiency at high Reynolds numbers and delaying boundary layer transition. Since this aerofoil is well-represented in the dataset, it serves as a benchmark for evaluating the neural network's ability to predict lift characteristics accurately when trained on a large number of similar data points (Abbott and Von Doenhoff, 1959).

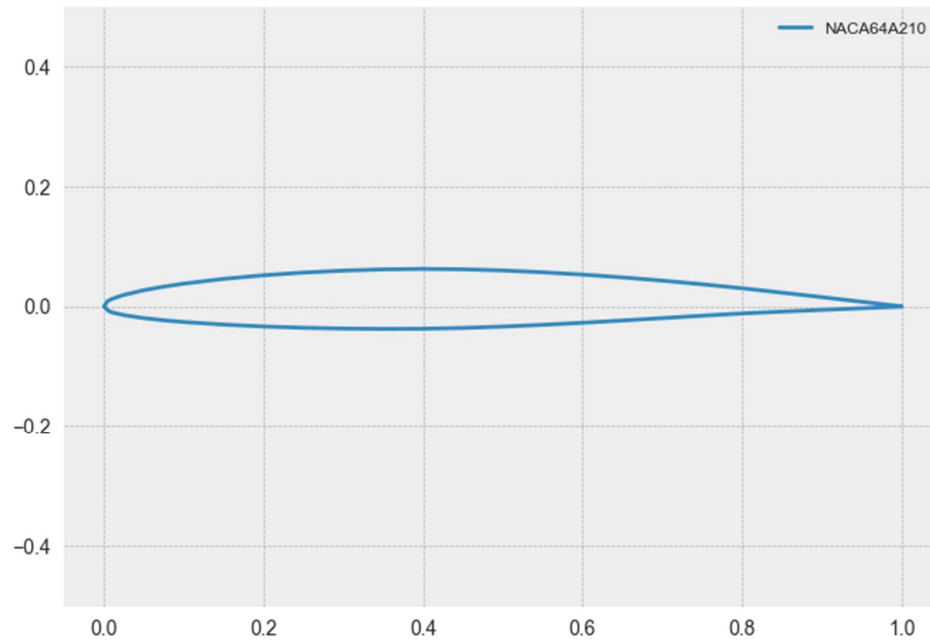


Figure 8: NACA 64A210

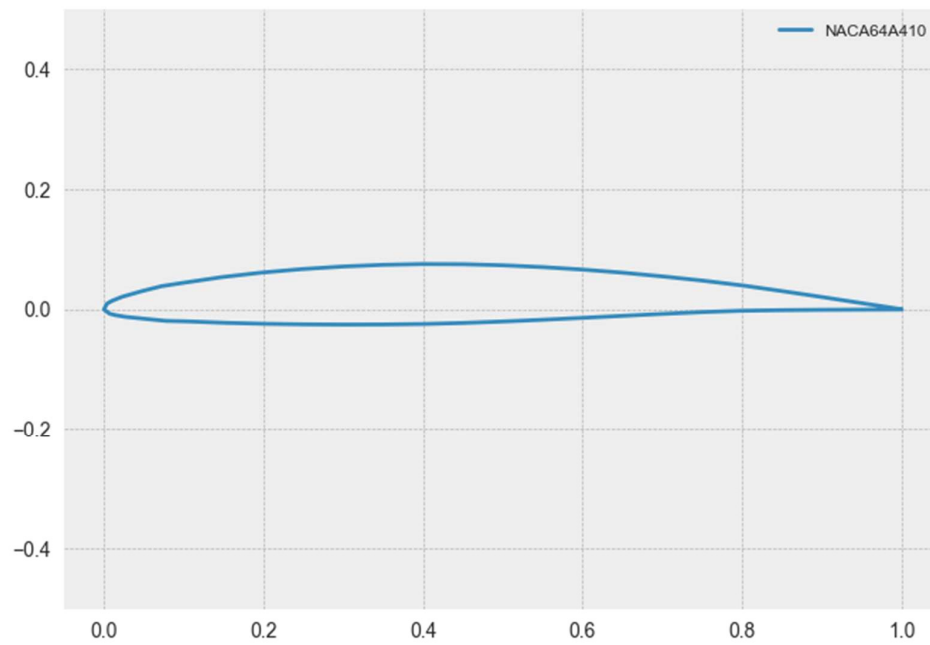


Figure 9: NACA 64A410



Figure 8 shows the NACA64A210 aerofoil and Figure 9 shows the NACA64A410 aerofoil, a similar aerofoil that is a part of the training dataset. One clear similarity seen between the two geometries is the thickness distribution, with both having their maximum thickness at similar chordwise positions. Since the model has seen these multiple similar geometries during training, this aerofoil serves as a benchmark confirming whether the model successfully captures expected lift characteristics.

The Dragonfly aerofoil was chosen to further assess the robustness of the model as it is an aerofoil in which the model has not seen any others of its type in training. This aerofoil is inspired by the wing structure of a dragonfly and is studied primarily in bio inspired aerodynamics, in applications such as small-scale drones. Due to this minimal representation in training, the neural networks ability to accurately predict the lift curve will likely be challenged, making it an excellent candidate for assessment on rare and unconventional aerofoil designs (Ansari, 2008).

Additionally, an aerofoil known to exhibit a non-linear curve was included to evaluate the ability of the model in handling complex aerodynamic behaviours. The FX 67-K-150/17 aerofoil designed by Franz Xavier Wortmann was developed for sailplanes and is widely used in high lift applications. It was selected in this case due to its alterations in lift characteristics, making it a challenging case for prediction. Evaluation on this type of aerofoil makes it possible to determine if the model can successfully capture non-linear aerodynamic phenomena or if it struggles in regions where flow separation occurs (Selig, 1996).

The NACA 0024 aerofoil was selected to represent thick aerofoils in this evaluation. With a maximum thickness of 24% of the chord length, this aerofoil is significantly thicker than conventional aerofoils used in high-speed aerodynamics. The NACA 00XX series is symmetric, meaning it does not generate lift at zero angle of attack but is widely used in vertical stabilisers, control surfaces, and slow-speed aircraft where high structural strength is needed. Along with the AH 93-W-300 aerofoil designed by Althaus for use in wind turbines, with a 30% maximum thickness of the length. Since thicker aerofoils exhibit different pressure distributions and flow characteristics than thinner ones, this aerofoil provides a challenging test case to determine if the neural network can accurately predict the lift characteristics of aerofoils outside the commonly represented mid-range thickness profiles (Abbott and Von Doenhoff, 1959).

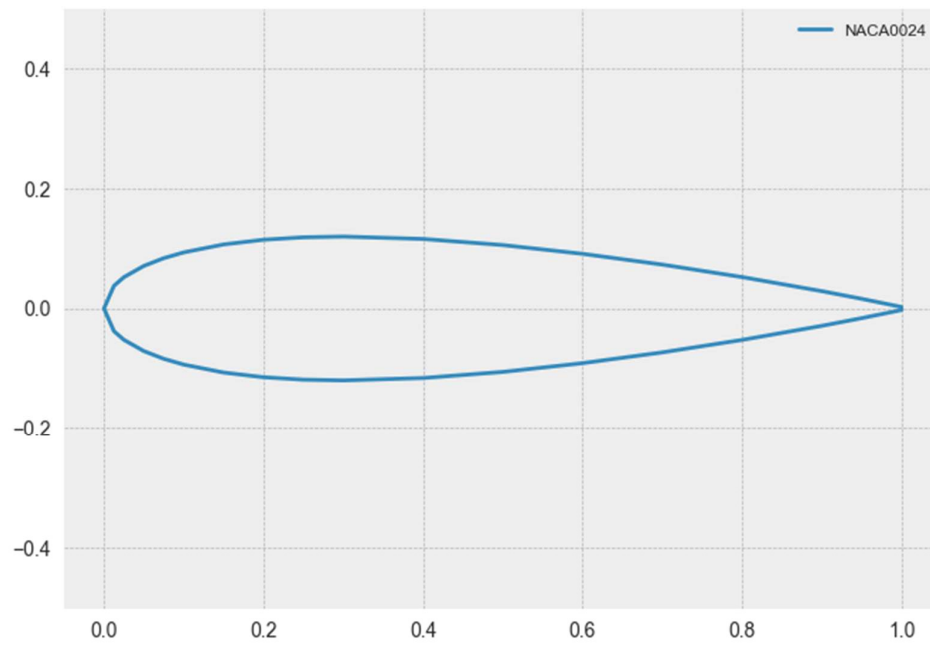


Figure 10: NACA 0024

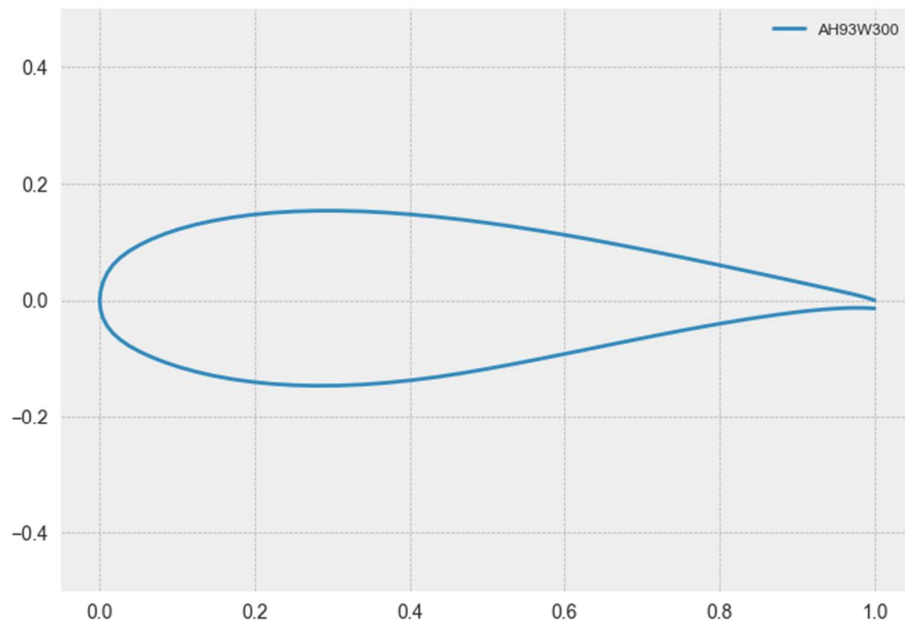


Figure 11: AH 93-W-300

Also, from the NACA 6-series, the NACA 65-206 aerofoil was chosen to test the model's performance with thin aerofoil geometries due to having a maximum thickness of 6% of the chord length. High-speed aircraft applications are most common use cases for aerofoils of this shape due to their abilities of minimising wave drag and flow separation. The Drela AG19 aerofoil was also used in the analysis of thin aerofoils, with a maximum thickness of 5.4%.

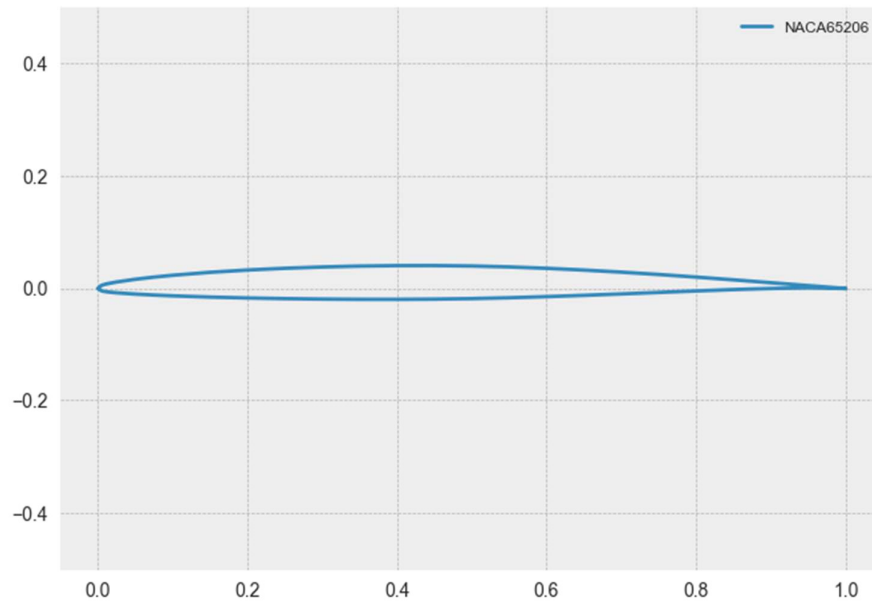


Figure 12: NACA 65-206

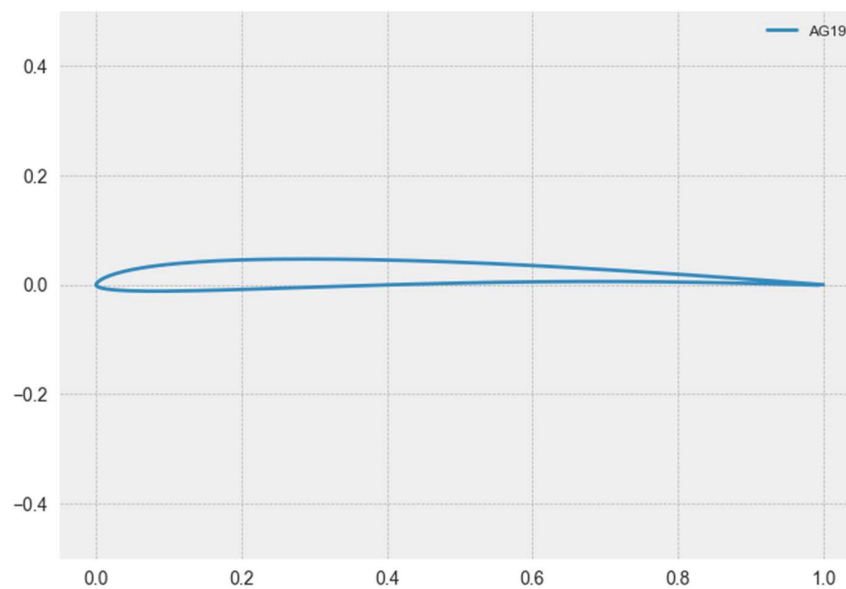


Figure 13: Drela AG19

By selecting these seven aerofoils, a balanced assessment of the model's predictive accuracy can be performed. The next step involves comparing the model's predictions to XFOIL-generated data, followed by a further evaluation of the model's predictions for another aerofoil from the test data against CFD generated results.

This CFD analysis was performed using ANSYS Fluent on the NACA 2410 aerofoil, another aerofoil from the test dataset, to further validate the predictive capabilities of the model. CFD provides a high-fidelity approach to aerodynamic analysis by solving the Navier-Stokes equations to simulate the complex interactions between fluid flow and an aerodynamic body. Although the dataset used in training is compiled from XFOIL simulations, which is based on potential flow theory with boundary-layer corrections, CFD enables a more detailed examination by accounting for viscous effects, flow separation and turbulent phenomena. This additional validation step ensures the model's predictions generalise well beyond the assumptions of inviscid flow models, reinforcing its application in real-world aerodynamic scenarios.

The NACA 2410 aerofoil was chosen for the CFD analysis due to its representation of conventional aerofoil geometries with moderate camber, making it a suitable case for comparison against the neural network's predictions. The study was conducted using steady-state, incompressible RANS equations with the Shear Stress Transport (SST) k- $\omega$  turbulence model. This turbulence model is well-suited for aerodynamic simulations as it accurately captures boundary layer separation and adverse pressure gradients, which are critical in predicting stall characteristics (Menter, 1994).

The use of a turbulence model in this study is necessitated by the relatively high Reynolds number ( $Re = 1,000,000$ ), representative of typical full-scale aerodynamic flows. At Reynolds numbers above  $\sim 100,000$ , laminar flow transitions to turbulent flow, especially over aerofoils where pressure gradients and surface curvature promote boundary layer instabilities. In such cases, turbulence plays a dominant role in dictating lift, drag, and stall behaviour. Direct Numerical Simulation (DNS) of turbulence at this  $Re$  would be computationally prohibitive due to the fine mesh and timestep requirements. Therefore, the SST k- $\omega$  model offers a balance between computational cost and accuracy by resolving the near-wall behaviour effectively while modelling the outer turbulent flow through blending functions (Wilcox, 2006; Versteeg and Malalasekera, 2007)

The simulations were conducted at a Reynolds number of 1,000,000, consistent with the XFOIL data generated for the aerofoils in the dataset and were conducted across angles of attack from  $-10^\circ$  to  $20^\circ$  in  $1^\circ$  increments to capture aerodynamic forces over a range of operating conditions (Abbott and Von Doenhoff, 1959). The freestream velocity was calculated based on this Reynolds number using equation 36.

$$Re = \frac{v \cdot \mu \cdot L}{\rho}$$

Equation 35

$L$  is the characteristic length, in this case it is the chord length, which is 1 m,  $\mu$  is the dynamic viscosity of air, at 288K (15°C) it is equal to  $1.802 \times 10^{-5}$  Pa·s and  $\rho$  is the density of air, which is  $1.225 \text{ kg/m}^3$ . Rearranging equation 36 for  $\nu$  works out to be 14.71 m/s.

The computational domain was set as a C type domain 15 times that of the chord and was discretised using an unstructured mesh with boundary-layer refinement to ensure an accurate resolution of near-wall flow characteristics. The mesh was constructed with a  $y^+$  value of less than 1, ensuring proper turbulence modelling within the boundary layer. The first cell height was determined as:

$$y^+ = \frac{yu_r}{\nu}$$

Equation 36

As an initial check the height of the first layer was made sure to be within the limit for  $y^+ < 1$ , ensuring that the viscous sublayer is well resolved. Where  $y^+$  is equal to 1,  $y$  is equal to  $2.35 \times 10^{-5}$  and the corresponding first layer spacing was approximately  $6.32 \times 10^{-6}$  meters, which is well within the limit. The final mesh consisted of approximately 250,000 elements, with a higher density of nodes near the aerofoil surface to resolve boundary-layer effects effectively.

The far-field boundary was set as a velocity inlet, while the aerofoil surface was treated as a no-slip wall condition to enforce realistic aerodynamic interactions. A pressure outlet was defined at the far boundary to allow for natural flow development.

The CFD-generated  $C_L$  values were compared with the neural network's predictions across the range of angles of attack, providing insight into the model's ability to generalize aerodynamic behaviour using the same metrics as before, and also plotted for a visual comparison.

This multi-faceted comparison ensures that the model is rigorously assessed across a variety of conditions, providing insight into its strengths and limitations in capturing aerodynamic trends. Furthermore, error metrics such as MSE, RMSE, MAE, and  $R^2$  will be computed for each aerofoil to quantitatively assess performance across different test cases.

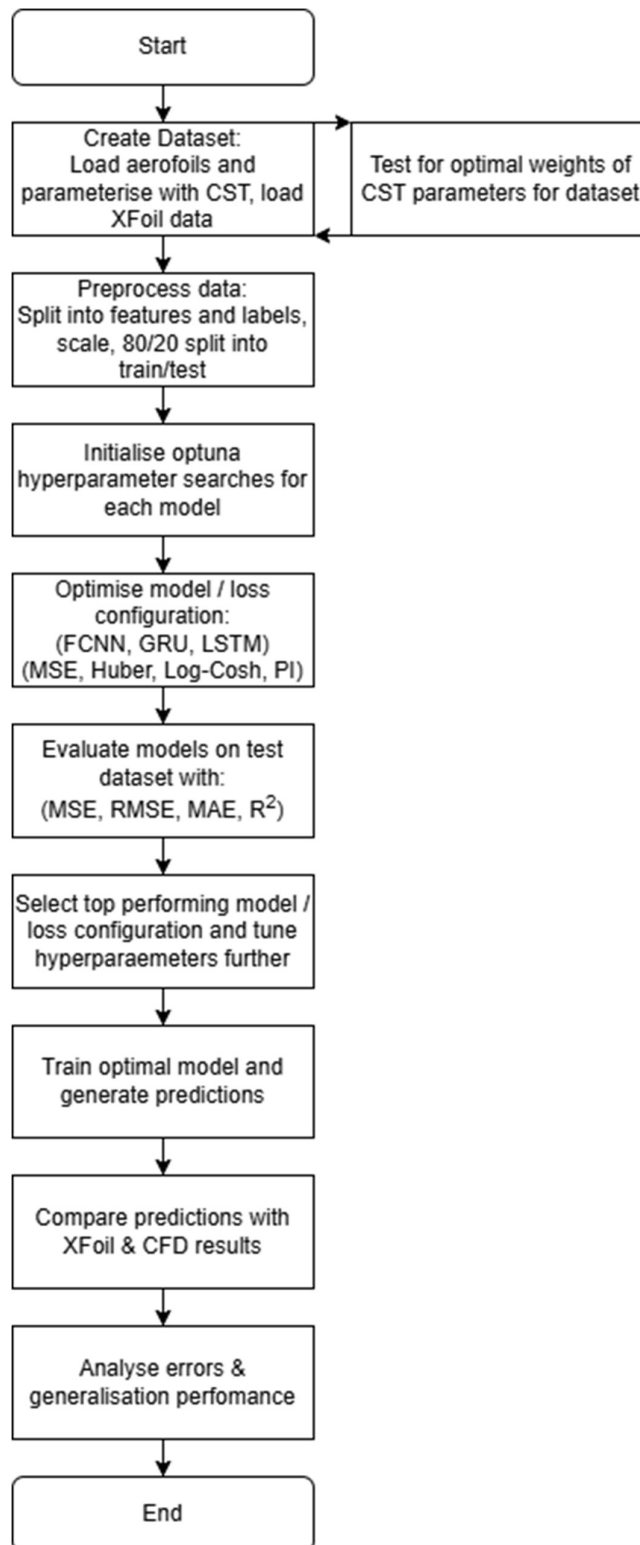


Figure 14: Full model development process

Figure 14 describes the full development process for the model, from dataset generation to full evaluation.

## Results and Discussion

This section states the results found from this study, first highlighting the performance of different neural networks with different loss functions.

### 1. Evaluation of models with different loss functions

As discussed in section 2 of the Technical Approach, a variety of loss functions were tested across GRU, LSTM and fully connected ANN (FCNN) models to determine the best combination for predicting lift curves. These loss functions are the mean squared error, Huber loss, Log-Cosh loss and a custom physics informed loss. Table 2 summarizes the evaluated performance metrics for each model across different loss functions.

Model	Loss Func	R2	MSE	RMSE	MAE
FCNN	PI	0.7795	0.8160	0.6966	0.5029
	MSE	0.7691	0.8775	0.7206	0.5149
	Huber	0.7643	0.8857	0.7248	0.5235
	Log-Cosh	0.7714	0.8754	0.7193	0.5222
GRU	PI	0.7681	0.8573	0.7147	0.5168
	MSE	0.7898	0.7581	0.6752	0.4823
	Huber	0.7768	0.8305	0.7038	0.5137
	Log-Cosh	0.7946	0.7360	0.6657	0.4800
LSTM	PI	0.7941	0.7561	0.6723	0.4912
	MSE	0.7934	0.7477	0.6697	0.4843
	Huber	0.7857	0.8058	0.6921	0.5078
	Log-Cosh	0.7927	0.7389	0.6674	0.4830

Table 2: model/loss combination results (Best performance for each loss function highlighted)

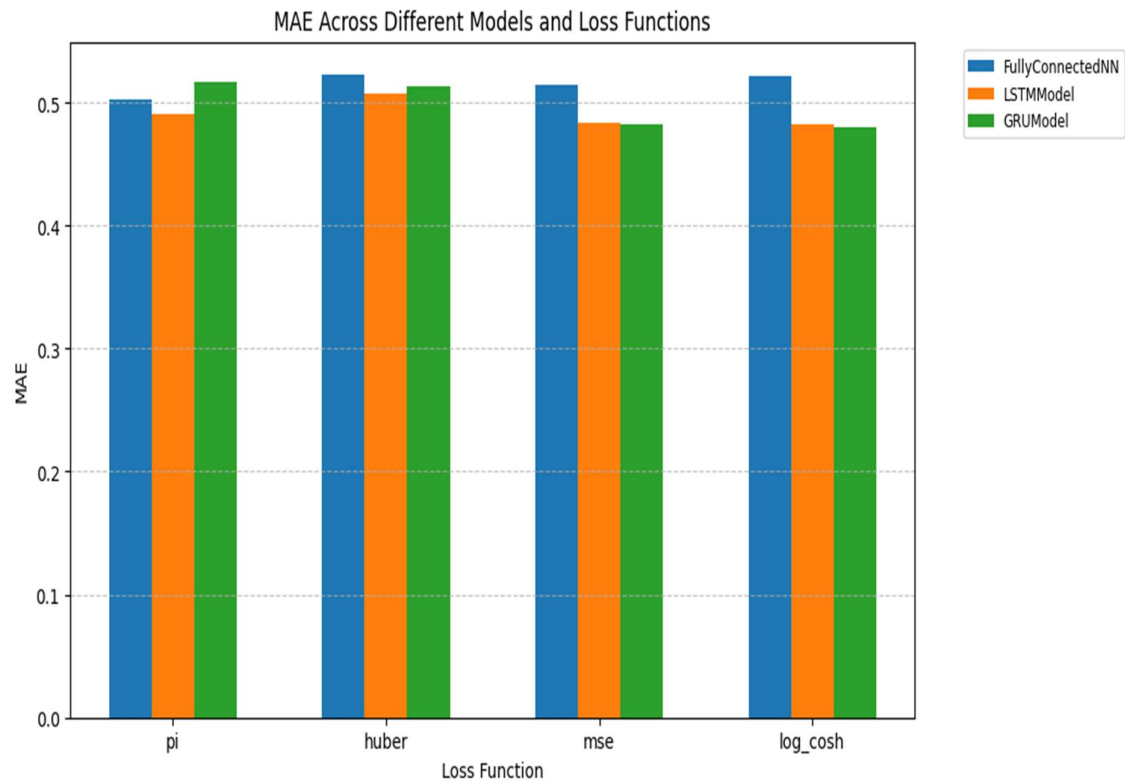


Figure 15: MAE across models and loss function

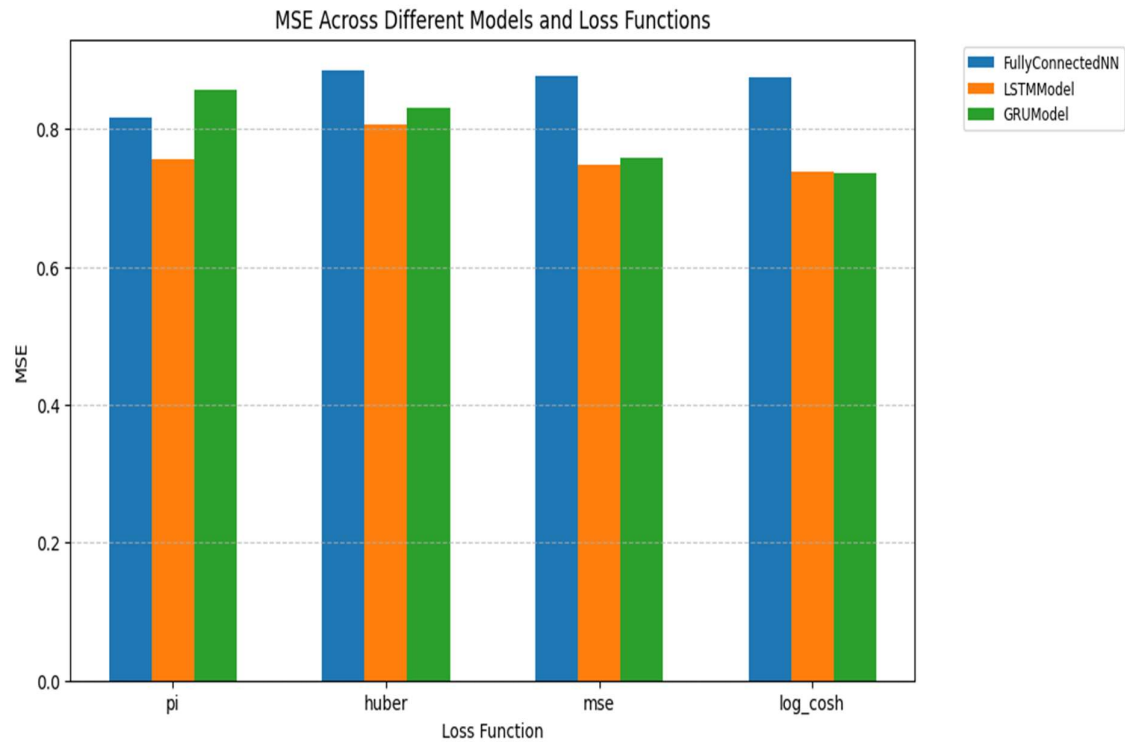


Figure 16: MSE across different models and loss functions



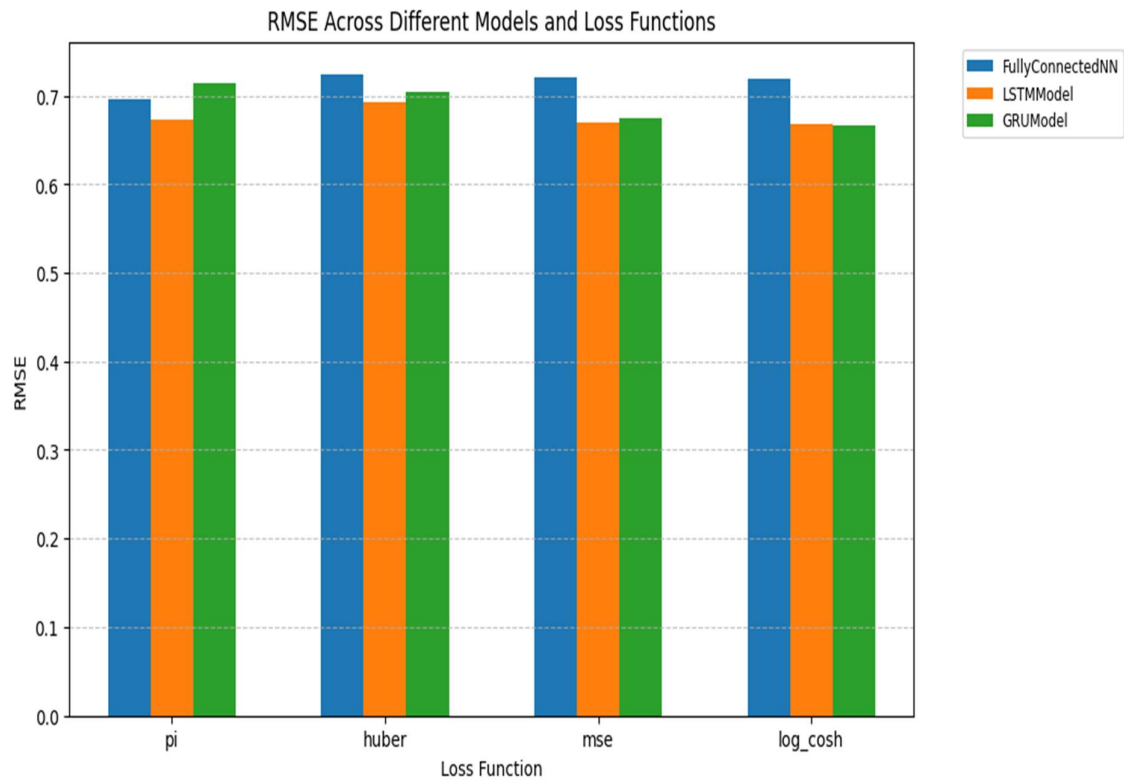


Figure 17: RMSE across different models and loss functions

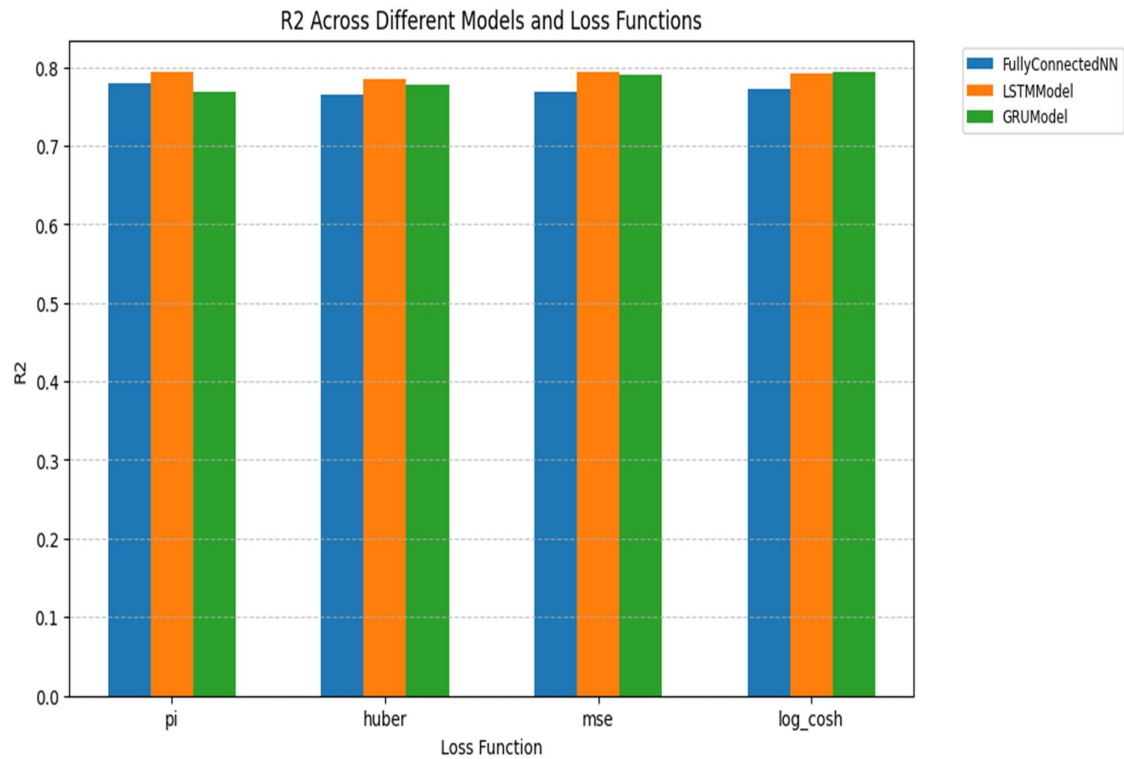


Figure 18:  $R^2$  across different models and loss functions

The results shown in table 2 and Figures 15 through 18 display that both the GRU and LSTM models outperform the standard FCNN across almost all of the evaluation metrics. This aligns with previous literature outlined in section 2.3 of the Literature Review and 3.2 of the Background Theory demonstrating the effectiveness of recurrent neural networks in capturing sequential relationships in aerodynamic data (Cho et al., 2014; Chung et al., 2015). The best performing configurations were found to be the GRU trained on the Log-Cosh loss and the LSTM trained on MSE, which consistently achieved lower errors and higher accuracy than all others.

The MAE achieved by the GRU model trained on Log-Cosh loss is clearly seen to be the lowest in Figure 15, achieving a MAE of 0.4800 with the GRU model trained with MSE closely following with a MAE of 0.4823 and the LSTM trained with MSE just behind that with a MAE of 0.4843. The MSE and RMSE on Figures 16 and 17 show a very similar trend with the GRU model with the Log-Cosh loss producing the lowest MSE and RMSE of 0.7360 and 0.6657 respectively, with the next closest of 0.7695 for MSE, and 0.6800 for RMSE for the LSTM model with Log-Cosh, and then 0.7477 for MSE and 0.6697 for RMSE for the LSTM with the MSE loss.

In terms of variance in predictions on Figure 18 the  $R^2$  scores show the highest as 0.7964 for the GRU with Log-Cosh, and again remarkably close with LSTM model with the PI loss function which achieved an  $R^2$  of 0.7941 and then the LSTM with MSE was next closest with a value of 0.7934.

Additionally, the time taken for training and evaluation of each of the two best performing models mentioned earlier was also considered. It was found to take 220.18s for the LSTM model with MSE and 107.08s for the GRU model with Log-Cosh. The GRU model reducing computational time by over 50% and requiring less computational resources supports that found by (Wu et al., 2023), along with lower error values for MSE, RMSE and MAE, as well as achieving the highest  $R^2$  score makes it the preferred architecture and the most suitable for computationally efficient predictions of  $C_L$  vs  $\alpha$  curves.

## 2. Final Model Evaluation and Analysis

With this optimal configuration of a GRU RNN with the Log-Cosh loss function, further hyperparameter tuning was the next step aiming to reduce the errors further by finding the most efficient architecture. Once again, Optuna was leveraged for this as a way of automating the process. The final model architecture was found to be 4 hidden layers each with 96 neurons with a dropout rate of roughly 0.14 and a learning rate of roughly 0.00037.

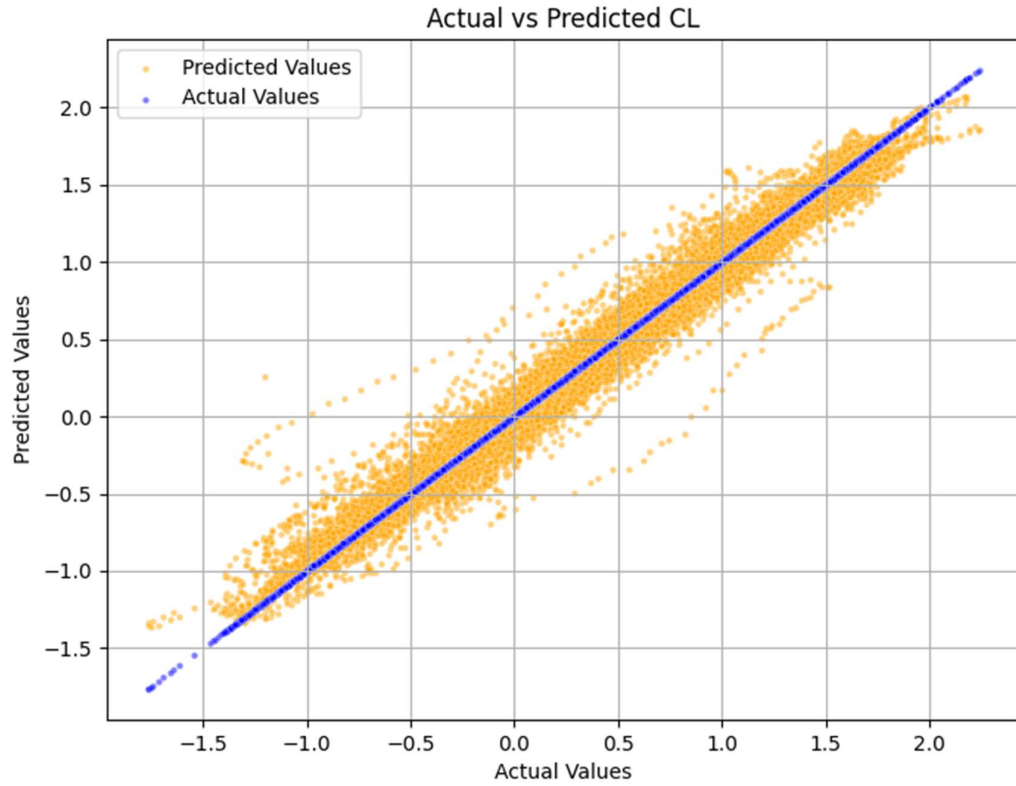


Figure 19: Actual vs Predicted  $C_L$

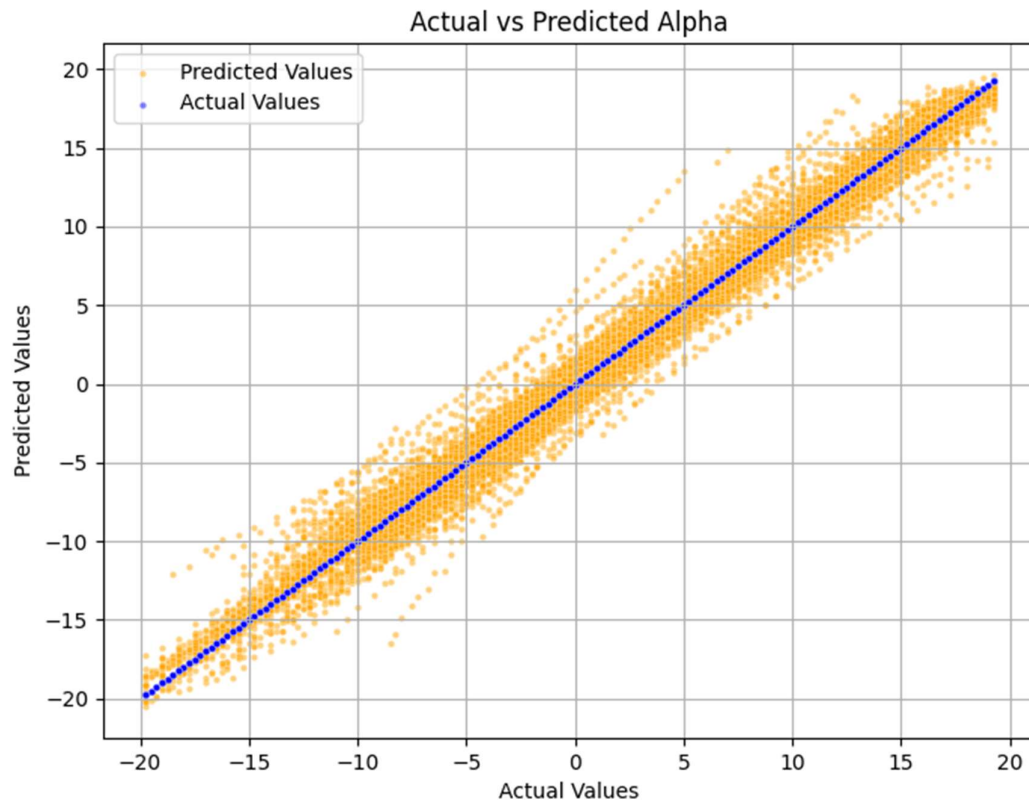


Figure 20: Actual vs Predicted  $\alpha$

Figures 19 and 20 are plots depicting the variance between the true values (blue) and predicted values (orange) of the model across the full testing dataset. The  $C_L$  variance is shown in Figure 19, and it can be seen that the model struggles with predicting values at the beginning and end of the lift curve. This implies that the unpredictability of the stall region and beyond does influence the model's predictive capabilities. This trend can be seen again on Figure 20 for the  $\alpha$  variance. The model tends to generalise better across the dataset within the pre-stall region, as the pre-stall region is typically linear.

As mentioned in the third section of the Technical Approach, the seven aerofoils chosen from the test dataset for the deeper evaluation aiming to assess the model's generalisation ability across multiple geometries are the NACA 64A210, DRAGONFLY, FX 67-K-150/17, NACA 0024, AH 93-W-300, NACA 65-206 and the AG19 aerofoils. The error analysis will be conducted via MSE, RMSE, MAE,  $R^2$  quantitatively and qualitatively compared through plots.

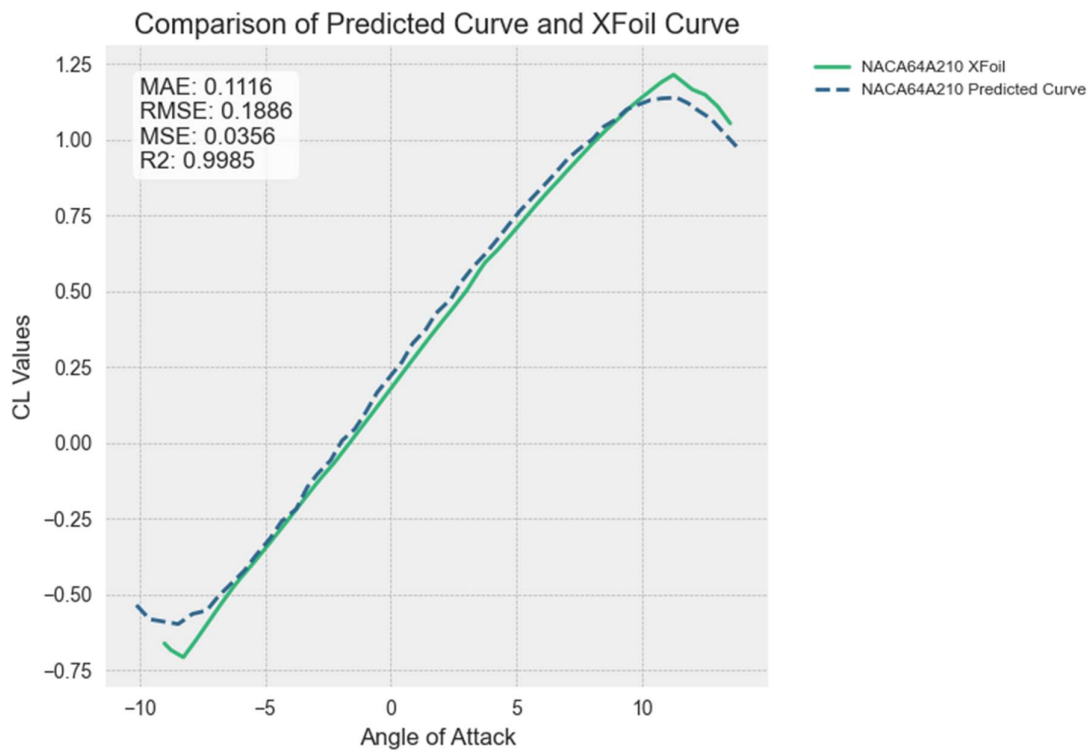


Figure 21: NACA 64A210 XFOIL vs Prediction

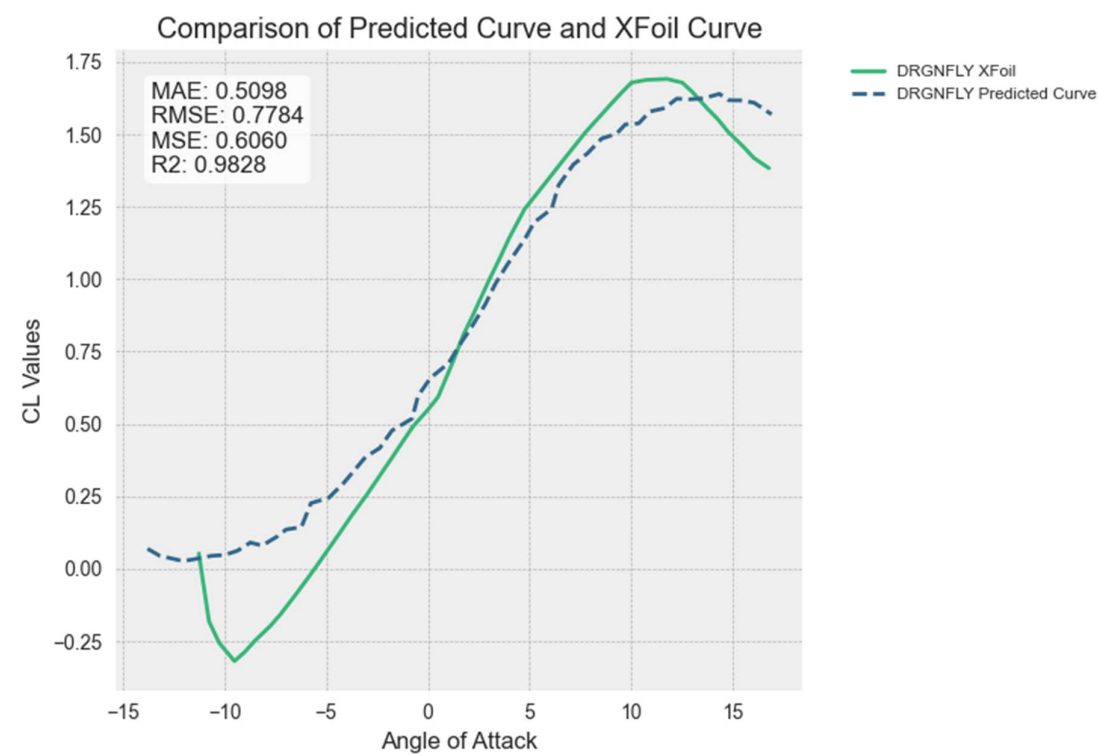


Figure 22: DRAGONFLY XFOIL vs Prediction

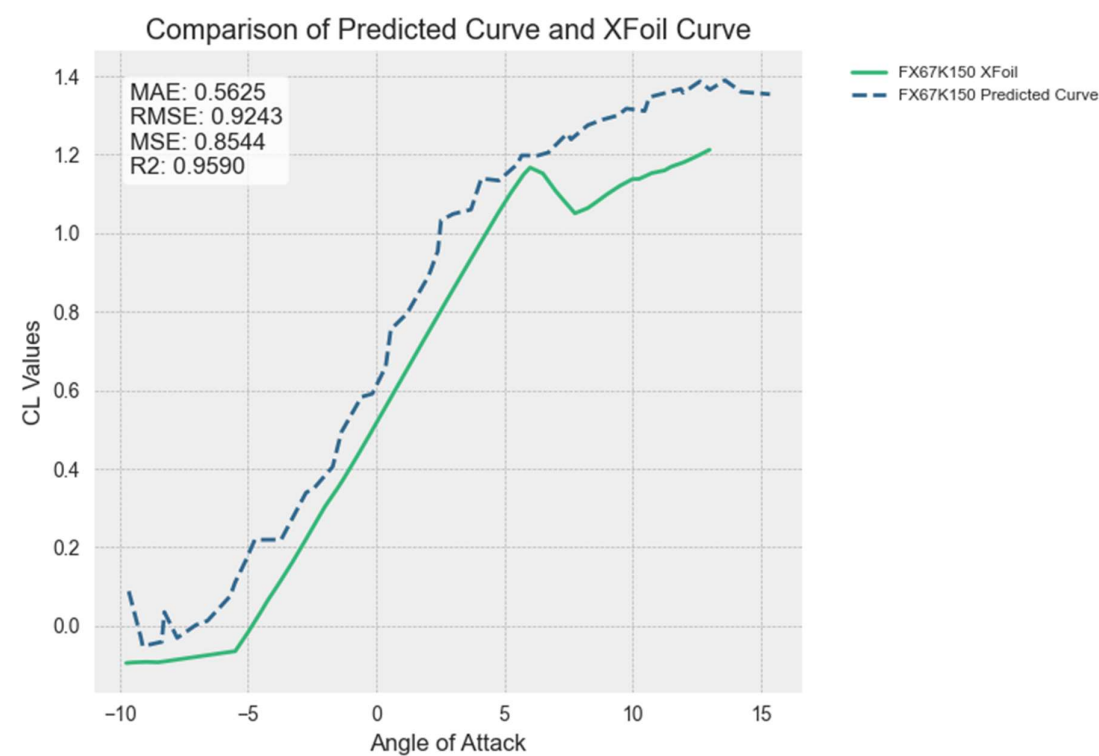


Figure 23: FX 67-K-150/17 XFOIL vs Prediction

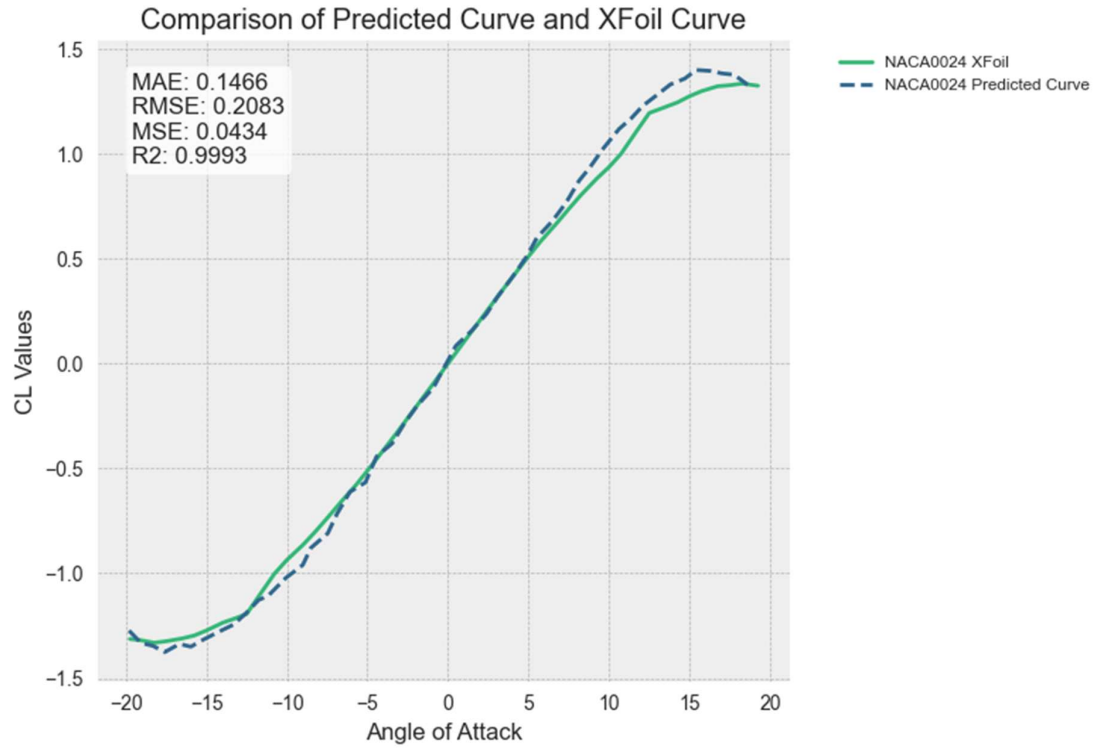


Figure 24: NACA 0024 XFOIL vs Prediction

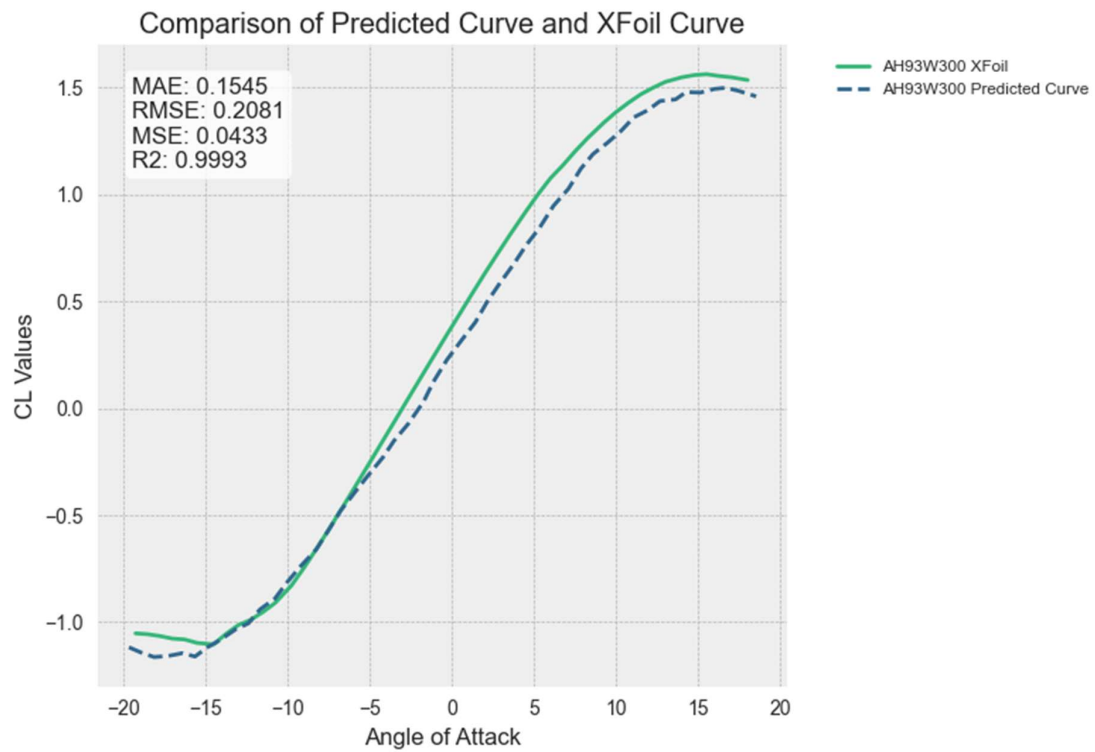


Figure 25: AH 93-W-300 XFOIL vs Prediction

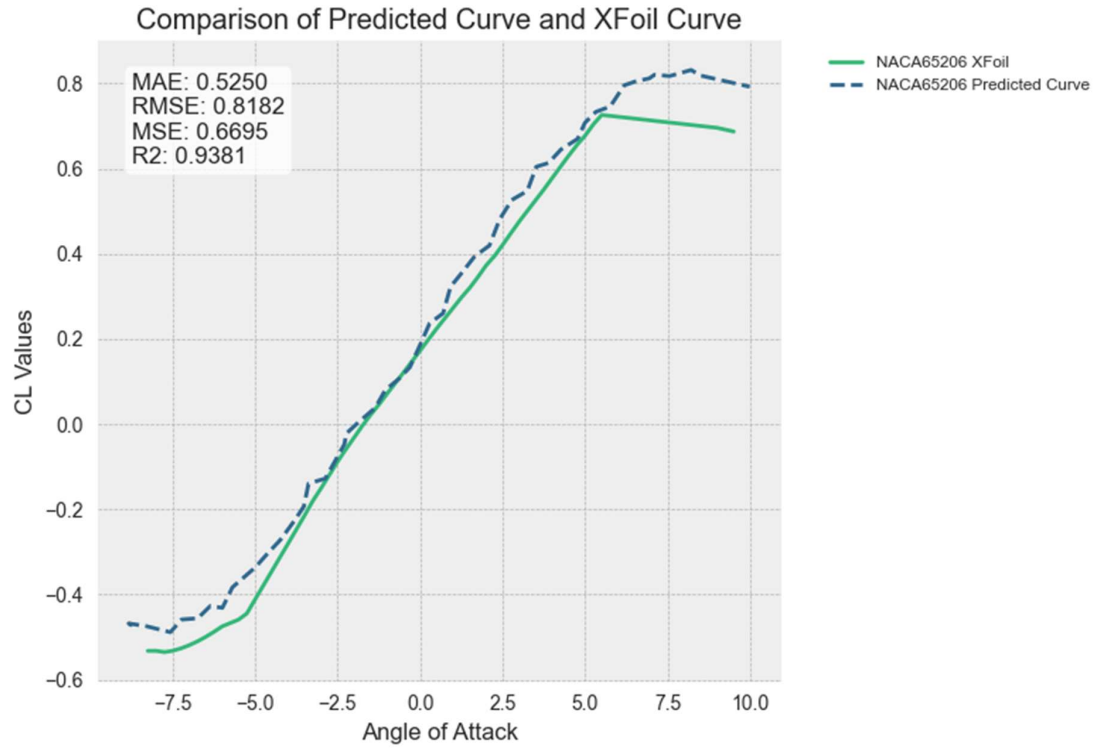


Figure 26: NACA 65-206 XFOIL vs Prediction

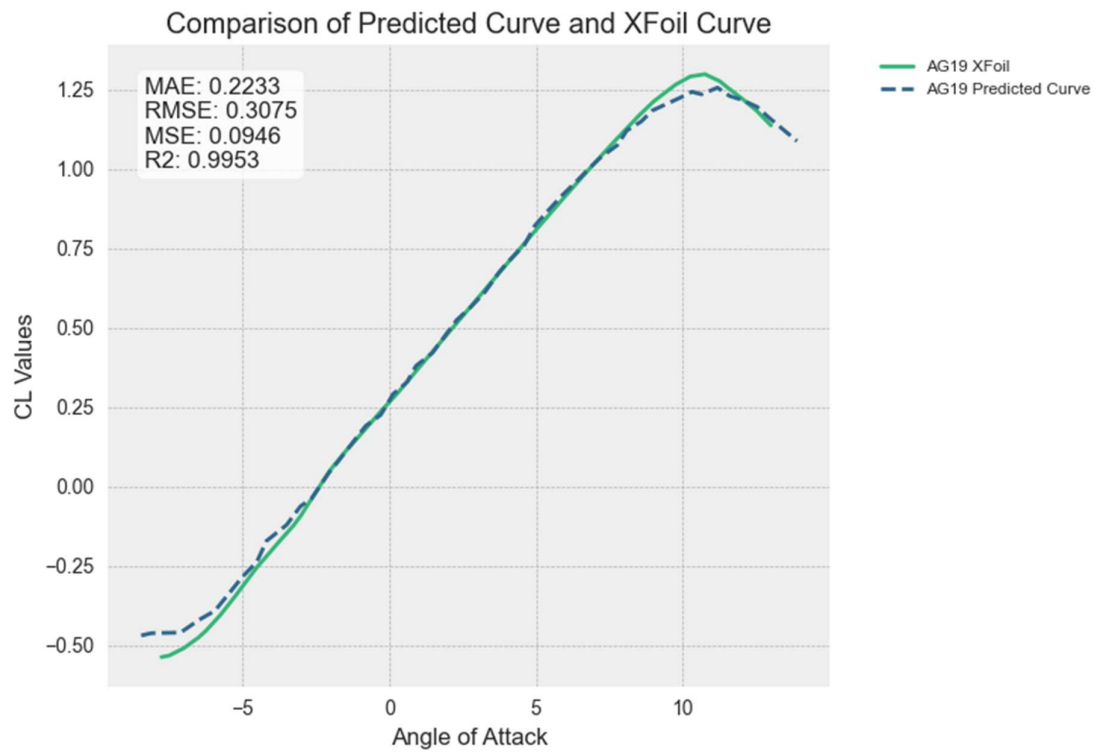


Figure 27: AG19 XFOIL vs Prediction

Figures 21 to 27 display the  $C_L$  vs  $\alpha$  curves for XFOIL calculated data (solid green line) and the model's predictions (dotted blue line). The quantitative error evaluation metrics are summarised within the plots.

The NACA 64A210 aerofoil, is part of a well-represented aerofoil family within the training dataset. As expected, the model's agreement with the XFOIL generated curve can be clearly seen on Figure 21, demonstrating its ability to accurately predict well-characterised aerofoil behaviours. The numerical evaluation further confirms this with a MAE of 0.1116, a RMSE of 0.1886, and a remarkably low MSE of 0.0356, indicating minimal deviation from the true values. The  $R^2$  score of 0.9985 shows a near perfect fit, reinforcing the model's robustness in predicting lift curves for well represented aerofoil shapes.

The DRAGONFLY aerofoil represents a unique case as it is not a part of any major aerofoil family and had no others of its type in the training dataset. On Figure 22 it can be seen visually that the model struggles with highly negative angles of attack, not capturing the distinct drop in lift and also an over-estimate of the stall angle can be seen. The error metrics highlight these challenges, with a much higher value of MAE, MSE and RMSE, 0.5098, 0.7784 and 0.6060 respectively, than that of the NACA 64A210 aerofoil. These values emphasize the deviations from the true curve. While still relatively high, the lower  $R^2$  score of 0.9828 does indicate discrepancies in capturing the lift curves non-linearity.

The FX 67-K-150/17, a high-lift aerofoil commonly used in applications requiring increased camber and delayed stall, presents a more complex aerodynamic profile. As seen in Figure 23, the model captures the general trend of the XFOIL curve, though it exhibits deviations throughout, the most noticeable are seen in the post-stall region where flow separation becomes dominant. The MSE of 0.8544 and RMSE of 0.9243 indicate moderate discrepancies and a higher level of outliers, with a MAE of 0.5625. The  $R^2$  of 0.9590 suggests that a general trend was captured but the ability to generalise for more complex lift characteristics requires further refinement (Van Dam, Associates and Center, 1984).

Chosen to evaluate the generalisation of thicker aerofoil shapes, the NACA 0024 aerofoil displayed an exceptional performance in its predictions, achieving a MAE of 0.1466 and RMSE of 0.2083 indicating an extremely close fit to the XFOIL-generated data, with an MSE of 0.0434 further supporting this observation. The  $R^2$  value of 0.9993 suggests a near-perfect correlation. Figure 24 displays this slope and the close agreement and minor deviations to the XFOIL generated curve correlate with these metrics. This impressive performance is likely due to the presence of similar thick NACA profiles in the training dataset, allowing the model to interpolate effectively for aerofoils of similar geometries.

Also chosen for the thicker aerofoil geometries, the AH 93-W-300 aerofoil also displayed exceptional predictive abilities from the model. Figure 25 shows the close agreement between the prediction and XFOIL data. The performance metrics of 0.0433 for MSE, 0.2081 for RMSE and 0.1545 for MAE being so low all align with this, and an  $R^2$  score of 0.9993 further assures the minute deviations between the predictions and XFOIL data.

The NACA 65-206 was selected as the thin aerofoil, featuring a maximum thickness of 6% chord. The network's performance on Figure 26 profile reveals notable discrepancies, with an MAE of



0.5250 and RMSE of 0.8182, indicating higher deviations than the well-represented aerofoils. The MSE of 0.6695 further suggests a lack of precision, while the  $R^2$  value of 0.9381, though relatively high, is the lowest among the NACA profiles. Qualitatively, the model struggles in capturing the high-lift regime, with clear divergence at higher angles of attack. This may be due to the underrepresentation of thin, high-speed aerofoils in the training data, affecting the model's ability to generalise well for these cases.

Another aerofoil chosen for the thinner aerofoil geometry assessment was the AG19 aerofoil with a max thickness of 5.4%, shown on Figure 27. The model was able to predict this aerofoil with great accuracy, it can be seen that the model captures the post stall behaviour almost entirely, with a slightly lower  $C_L$  at the stall angle, but predicting the stall angle very closely. The error metrics again align with what can be seen with a MAE of 0.2233, RMSE of 0.3075, MSE of 0.0946 and  $R^2$  score of 0.9953. The fact of the aerofoil being well represented in the training dataset will be one reason for this strong generalisation.

Key trends emerge from the qualitative and quantitative analyses of the trained model's predictions against the XFOIL generated values. The model performs outstandingly with aerofoils well-represented in the training dataset, evidenced by almost perfect predictions for the NACA 64A210 and NACA 0024 aerofoils. However, uncommon, or underrepresented geometries lead to deterioration in performance, with significantly increased errors, shown by DRAGONFLY and NACA 65-206. Aerofoils exhibiting sharp lift variations, such as FX 67-K-150/17, also pose a challenge, likely down to the unpredictability of lift characteristics in post-stall regions and the areas in which flow separation has occurred, which is difficult for the model to capture based on a limited dataset.

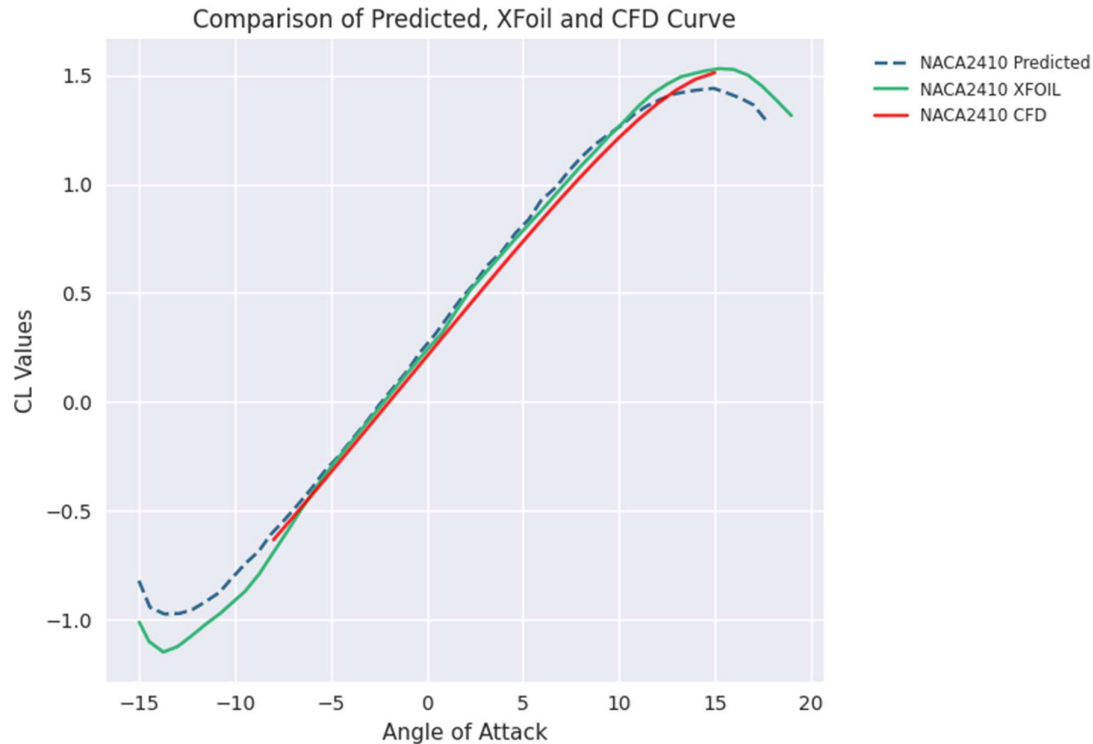


Figure 28: NACA 2410 Predicted vs XFOIL vs CFD

To further validate the model's predictions beyond the XFOIL generated data, ANSYS Fluent was used to conduct a CFD based analysis, as described in section 3 of the Technical Approach, on the NACA 2410 aerofoil chosen from the test dataset. The simulation was to be between  $-10^\circ$  to  $20^\circ$  in  $1^\circ$  increments, however, due to the inherent numerical challenges associated with flow separation and stall, the simulation failed to converge for angles above  $15^\circ$ , as well as highly negative angles, below  $-8^\circ$ . As seen on Figure 28, for the XFOIL curve, which is based on experimental data, stall occurs around  $15^\circ$ , while nonlinear behaviour comes evident below  $-8^\circ$ . Despite attempts to resolve this, through a transient assessment and alternative turbulence models, for the Reynolds number of one million, non-physical or unstable results were obtained, rendering them unsuitable for inclusion. Issues with convergence are well documented limitations of CFD when approaching stall and post-stall regimes, particularly in high angle of attack regions, where unsteady flow dynamics dominate (Versteeg and Malalasekera, 2007).

Nonetheless, the results obtained provide a robust validation region, covering the full linear portion of the pre-stall region on the lift curve, up to the onset of the stall. As seen on Figure 28, the model's prediction aligns remarkably well with that of the CFD-generated lift coefficients, with a slightly lower predicted value at the stall point, though it does predict the stall angle with high accuracy. This demonstrates the model's ability to replicate realistic aerodynamic behaviour under conditions beyond that of its training, reinforcing its effectiveness for practical use cases.

While the inability to simulate the lift curve is a limitation for this validation, it also highlights the value of the ANN model, which, unlike CFD, can provide accurate predictions across the full range of angles of attack almost instantaneously. The discrepancies observed in the post stall region of earlier evaluations with XFOIL remain consistent with the expected complexity of these

flow regimes, validating both the physical realism of the dataset used in training and the challenges posed by high-angle predictions.

---

## Future Work

This study demonstrates the potential of neural networks for aerodynamic lift prediction, but certain improvements could enhance accuracy and robustness.

One key area is further refining the physics-informed constraints in the loss function. Additional aerodynamic stability parameters, such as moment coefficients or pressure distributions, could be integrated to improve accuracy, particularly in regions with high flow separation or dynamic stall effects.

Expanding the dataset through data augmentation could also enhance generalisation. By generating synthetic aerofoils with slight modifications to camber, thickness, and leading-edge radius, the training data could be diversified beyond the limits of current publicly available databases, improving performance on unseen designs. These synthetic geometries could be assessed both using traditional CFD and by comparing predicted lift curves directly with model outputs, allowing for iterative validation and potential optimisation loops.

Validating the model against a broader set of high-fidelity CFD simulations, with a full range of angles of attack, would provide a more rigorous assessment. While XFOIL data has been used for training, comparing predictions against more RANS simulations in ANSYS Fluent would help identify areas where the model struggles, particularly in stall and post-stall conditions.

Finally, extending the scope of the model to predict drag coefficients ( $C_D$ ) and moment coefficients ( $C_M$ ) alongside  $C_L$  would create a more comprehensive aerodynamic tool. Such an expansion could be valuable in a wide range of applications including aircraft conceptual design, control surface tuning, and wind turbine performance analysis, where multi-dimensional aerodynamic understanding is crucial.

---

## Conclusion

This project demonstrates the successful development of a deep learning model capable of accurately predicting the aerodynamic lift coefficient as a function of angle of attack for a broad range of aerofoil geometries. Leveraging the CST parameterisation method for aerofoil shape representation and training neural network models using lift curve data generated through XFOIL, the aim of the study is the production of a computationally efficient alternative to traditional simulation-based aerodynamic analysis. Through extensive experimentation, three neural network architectures were compared, a fully connected multilayer perceptron, gated recurrent unit (GRU) recurrent neural network and a long short-term memory (LSTM) recurrent neural network, trained across four different loss functions. Performance was evaluated using four

established metrics: mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and the coefficient of determinations ( $R^2$  score).

Initial results demonstrate the recurrent architectures outperform that of the simpler standard fully connected neural network architecture across almost all evaluation metrics, confirming findings from previous literature regarding the advantages of temporal modelling in sequential tasks. The optimal configuration was determined to be a GRU model trained on the Log-Cosh loss function, achieving the lowest MAE of 0.4800, MSE of 0.7360 and RMSE of 0.6657, and the highest  $R^2$  score of 0.7946 among all model-loss configurations tested. Furthermore, the GRU's computational efficiency, requiring just over 100 seconds for training and evaluation compared to over 220 seconds for the LSTM model, made it the most suitable model for practical aerodynamic prediction tasks (Chung et al., 2014; Mienye, Swart and Obaido, 2024).

Once an optimal hyperparameter configuration was found from Optuna studies, the final model was then evaluated across the dataset as a whole, along with a diverse selection of individual aerofoils chosen to challenge distinct aspects of the generalisation ability. The NACA series aerofoils demonstrated the model's exceptional generalisation to familiar geometries, but issues arise for those with less representation or more non-linear lift profiles, such as that of the DRAGONFLY aerofoil or the FX 67-K-150/17.

The first key conclusions that can be drawn from these observations, was the success of the GRU model in predicting lift curves which confirms the suitability of recurrent neural networks in modelling aerodynamic phenomena, particularly when trained on compact but efficiently expressive input representations such as CST parameters. Secondly, unmistakable evidence supporting the importance of dataset diversity is shown through the far greater predictive accuracy of well represented geometries over more complex and rare geometries. Finally, although the model exhibits a robust performance overall, near stall and post stall regions are areas of struggle, where sharp, unpredictable, nonlinear changes in lift occur and flow separation becomes prominent, highlighting areas for future research and improvement.

Importantly, validation against CFD-simulated lift data for the NACA 2410 aerofoil provided further evidence of the model's real-world applicability. The model not only predicted the stall angle with high accuracy but also closely followed the CFD curve throughout the entire linear region of the lift curve, highlighting strong adherence to physical aerodynamic behaviour. This agreement reinforces the credibility of the model beyond synthetic datasets and demonstrates its potential for practical aerodynamic design applications.

This study provides a robust, data-driven framework capable of estimating the lift coefficient vs angle of attack curve with a high degree of accuracy and efficiency. The reduction in computational costs and time in comparison to computational fluid dynamics simulations and experimental methods makes it a valuable tool for early-stage aerodynamic design and analysis workflows. The place the model holds in aerodynamic performance estimation is reinforced by the predictive accuracy across a wide range of aerofoil geometries. Overall, the success of this study not only demonstrates the potential of deep learning in aerodynamic modelling but also contributes toward the development of accessible, open source and computationally efficient tools for engineers and researchers.

---

## References

Abbott, I.H.A. and Von Doenhoff, A.E. (1959) *Theory of wing sections including a summary of airfoil data*. [New ], Dover ed. New York: Dover Publications.

*Airfoil Tools* (2019) [online]

Available at: <http://airfoiltools.com/>

[Accessed: 8 Nov]

Akiba, T., Sano, S., Yanase, T., Ohta, T. and Koyama, M. (2019) Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* [online], 2623-2631

Available at: <https://arxiv.org/pdf/1907.10902>

[Accessed: 10 Dec 2024]

Ali, O. (2016) Chapter 1 - Fluid Mechanics and Biofluids Principles. In: Ali, O. (ed.) *Biofluid Mechanics*. Academic Press. pp. 1-60.

Anderson, J.D. (2017) *Fundamentals of aerodynamics*. Sixth edition.

International student edition. ed. Columbus, OH: McGraw-Hill.

Ansari, S.a.K.K. (2008) Insectlike Flapping Wings in the Hover Part I: Effect of Wing Kinematics. *Journal of Aircraft - J AIRCRAFT*, 45, 1945-1954.

Apicella, A., Donnarumma, F., Isgrò, F. and Prevete, R. (2021) A survey on modern trainable activation functions. *Neural Networks*, 138, 14-32.

Ava Shahrokhi and Alireza, J. (2007) Airfoil shape parameterization for optimum Navier–Stokes design with genetic algorithm. *Aerospace Science and Technology*, 11 (6), 443-450.

Bogue, D. and Crist, N. (2008) CST transonic optimization using Tranair++. *46th AIAA aerospace sciences meeting and exhibit* [online], 321

Available at: <https://www.calmarresearch.com/STG/Tranair/media/AIAA-96013.pdf>

[Accessed: 19/11/24]

Bonnet, F., Mazari, J., Cinnella, P. and Gallinari, P. (2022) Airfrans: High fidelity computational fluid dynamics dataset for approximating reynolds-averaged navier–stokes solutions. *Advances in Neural Information Processing Systems*, 35, 23463-23478.

Brunton, S.L., Nathan Kutz, J., Manohar, K., Aravkin, A.Y., Morgansen, K., Klemisch, J., Goebel, N., Buttrick, J., Poskin, J. and Blom-Schieber, A.W. (2021) Data-driven aerospace engineering: reframing the industry with machine learning. *AIAA Journal*, 59 (8), 2820-2847.

Brunton, S.L., Noack, B.R. and Koumoutsakos, P. (2020) Machine learning for fluid mechanics. *Annual review of fluid mechanics*, 52 (1), 477-508.

Cai, S., Mao, Z., Wang, Z., Yin, M. and Karniadakis, G.E. (2021) Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37 (12), 1727-1738.

Cattafesta, L., Bahr, C. and Mathew, J. (2010) Fundamentals of wind-tunnel design. *Encyclopedia of aerospace engineering*, 1-10.

Chen, H., Flores, G.E.C. and Li, C. (2024) Physics-informed neural networks with hard linear equality constraints. *Computers & Chemical Engineering*, 189, 108764.

Cherry, S. and Hawk, K. (2024) Using Artificial Neural Networks to Calculate Lift Coefficients for an Unknown Airfoil. *2024 Regional Student Conferences* [online], 84470

Available at: <https://arc.aiaa.org/doi/epdf/10.2514/6.2024-84470>

[Accessed: 10 Nov 2024]

Chicco, D., Warrens, M.J. and Jurman, G. (2021) The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *Peerj computer science*, 7, e623.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Cho, M., Kim, C., Jung, K. and Jung, H. (2022) Water level prediction model applying a long short-term memory (lstm)-gated recurrent unit (gru) method for flood prediction. *Water*, 14 (14), 2221.

Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Dawson, S.T., Hemati, M., Floryan, D.C. and Rowley, C.W. (2016) Lift enhancement of high angle of attack airfoils using periodic pitching. *54th AIAA Aerospace Sciences Meeting* [online], 2069

Available at: <https://arc.aiaa.org/doi/10.2514/6.2016-2069>

[Accessed: 10 Nov 2024]

Della Vecchia, P., Daniele, E. and D'Amato, E. (2014) An airfoil shape optimization technique coupling PARSEC parameterization and evolutionary algorithm. *Aerospace Science and Technology*, 32 (1), 103-110.

Derksen, R. and Rogalsky, T. (2010) Bezier-PARSEC: An optimized aerofoil parameterization for design. *Advances in engineering software*, 41 (7-8), 923-930.

Dey, R. and Salem, F.M. (2017) Gate-variants of gated recurrent unit (GRU) neural networks. *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)* [online], 1597-1600

Available at: <https://arxiv.org/pdf/1701.05923>

[Accessed: 2 Jan 2025]

Drela, M. (1989) XFOIL: An analysis and design system for low Reynolds number airfoils. *Low Reynolds Number Aerodynamics: Proceedings of the Conference Notre Dame, Indiana, USA, 5-7 June 1989* [online], 1-12

Available at: [https://web.mit.edu/drela/Public/papers/xfoil\\_sv.pdf](https://web.mit.edu/drela/Public/papers/xfoil_sv.pdf)

[Accessed: 30 Jan 2025]

Drela, M. and Giles, M.B. (1987) Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA journal*, 25 (10), 1347-1355.

Du, X., He, P. and Martins, J.R. (2021) Rapid airfoil design optimization via neural networks-based parameterization and surrogate modeling. *Aerospace Science and Technology*, 113, 106701.

Fukami, K., Fukagata, K. and Taira, K. (2020) Assessment of supervised machine learning methods for fluid flows. *Theoretical and Computational Fluid Dynamics*, 34 (4), 497-519.

Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep learning*. Cambridge, Massachusetts ;; The MIT Press.

Gray, G.W. (1948) *Frontiers of flight: The story of NACA research. (No Title)*.

Hastie, T., Tibshirani, R. and Wainwright, M. (2015) Statistical learning with sparsity. *Monographs on statistics and applied probability*, 143 (143), 8.

Hastie, T.a.T.R.a.F.J. (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*.

Hebb, D.O. (1949) The first stage of perception: growth of the assembly. *The Organization of Behavior*, 4 (60), 78-60.

Hecht-Nielsen, R. (1992) Theory of the backpropagation neural network. In: (ed.) *Neural networks for perception*. Elsevier. pp. 65-93.

Henderi, H., Wahyuningsih, T. and Rahwanto, E. (2021) Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. *International Journal of Informatics and Information Systems*, 4 (1), 13-20.

Hinton, G.E., Osindero, S. and Teh, Y.-W. (2006) A fast learning algorithm for deep belief nets. *Neural computation*, 18 (7), 1527-1554.

Hochreiter, S. and Schmidhuber, J. (1997) Long short-term memory. *Neural computation*, 9 (8), 1735-1780.

Huber, P.J. (1992) Robust estimation of a location parameter. In: (ed.) *Breakthroughs in statistics: Methodology and distribution*. Springer. pp. 492-518.

Jin, S.-Y., Chen, S.-S., Feng, C. and Gao, Z.-H. (2024) Deep learning for airfoil aerodynamic-electromagnetic coupling optimization with random forest. *Physics of Fluids*, 36 (1).

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S. and Yang, L. (2021) Physics-informed machine learning. *Nature Reviews Physics*, 3 (6), 422-440.

Karpatne, A., Atluri, G., Faghmous, J.H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N. and Kumar, V. (2017) Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering*, 29 (10), 2318-2331.

Kingma, D.P. (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kulfan, B. (n.d.) *Modification of CST Airfoil Representation Methodology* [online]

Available at:

[https://www.brendakulfan.com/\\_files/ugd/169bff\\_16a868ad06af4fea946d299c6028fb13.pdf](https://www.brendakulfan.com/_files/ugd/169bff_16a868ad06af4fea946d299c6028fb13.pdf)

[Accessed: 13/02]

Kulfan, B. and Bussoletti, J. (2006) "Fundamental" parametric geometry representations for aircraft component shapes. *11th AIAA/ISSMO multidisciplinary analysis and optimization conference* [online], 6948

Available at: <https://arc.aiaa.org/doi/epdf/10.2514/6.2006-6948>

[Accessed: 11 Nov 2024]

Kulfan, B.M. (2007) The "CST" universal parametric geometry representation method, Recent Extensions and Applications. *Proc R Aeronaut Soc Conf* [online], 114 (1135), 157-176



Available at:

[https://www.brendakulfan.com/\\_files/ugd/169bff\\_99860eaf4a9d4860a468e03fe54ec6ff.pdf](https://www.brendakulfan.com/_files/ugd/169bff_99860eaf4a9d4860a468e03fe54ec6ff.pdf)

[Accessed: 28 Jan 2025]

Kulfan, B.M. (2008) Universal parametric geometry representation method. *Journal of aircraft*, 45 (1), 142-158.

Kulfan, B.M. (2010) Recent extensions and applications of the 'CST' universal parametric geometry representation method. *The Aeronautical Journal*, 114 (1153), 157-176.

Lane, K. and Marshall, D. (2009) A surface parameterization method for airfoil optimization and high lift 2D geometries utilizing the CST methodology. *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition* [online], 1461

Available at: <https://arc.aiaa.org/doi/epdf/10.2514/6.2009-1461>

[Accessed: 07 Feb 2025]

Lane, K. and Marshall, D. (2010) Inverse airfoil design utilizing CST parameterization. *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* [online], 1228

Available at: <https://arc.aiaa.org/doi/epdf/10.2514/6.2010-1228>

[Accessed: 07 Feb 2025]

Le, D.K. and Yoon, J.Y. (2023) A hybrid CFD–Deep learning methodology for improving the accuracy of pressure drop prediction in cyclone separators. *Chemical Engineering Research and Design*, 190, 296-311.

LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep learning. *nature*, 521 (7553), 436-444.

Linse, D.J. and Stengel, R.F. (1993) Identification of aerodynamic coefficients using computational neural networks. *Journal of Guidance, Control, and Dynamics*, 16 (6), 1018-1025.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y. and Alsaadi, F.E. (2017) A survey of deep neural network architectures and their applications. *Neurocomputing*, 234, 11-26.

McCulloch, W.S. and Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 115-133.

Medsker, L.R. and Jain, L. (2001) Recurrent neural networks. *Design and Applications*, 5 (64-67), 2.

Menter, F.R. (1994) Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA journal*, 32 (8), 1598-1605.

Meyer, G.P. (2021) An alternative probabilistic interpretation of the huber loss. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* [online], 5261-5269

Available at: <https://arxiv.org/pdf/1911.02088>

[Accessed: 11 Nov 2024]

Mienye, I.D., Swart, T.G. and Obaido, G. (2024) Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, 15 (9), 517.

Morgado, J., Vizinho, R., Silvestre, M. and Páscoa, J. (2016) XFOIL vs CFD performance predictions for high lift low Reynolds number airfoils. *Aerospace Science and Technology*, 52, 207-214.

Mukesh, R., Lingadurai, K. and Selvakumar, U. (2014) Airfoil shape optimization using non-traditional optimization technique and its validation. *Journal of King Saud University-Engineering Sciences*, 26 (2), 191-197.

Nair, V. and Hinton, G.E. (2010) Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th international conference on machine learning (ICML-10)* [online], 807-814

Available at: <https://www.cs.toronto.edu/~fritz/absps/reluICML.pdf>

[Accessed: 15 Feb 2025]

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N. and Antiga, L. (2019) Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Patro, S. and Sahu, K.K. (2015) Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.

Popescu, M.-C., Balas, V.E., Perescu-Popescu, L. and Mastorakis, N. (2009) Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8 (7), 579-588.

Raissi, M., Perdikaris, P. and Karniadakis, G.E. (2017) Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*.

Raissi, M., Perdikaris, P. and Karniadakis, G.E. (2019) Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378, 686-707.

Ranzato, M.A., Huang, F.J., Boureau, Y.-L. and LeCun, Y. (2007) Unsupervised learning of invariant feature hierarchies with applications to object recognition. *2007 IEEE conference on computer vision and pattern recognition* [online], 1-8

Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4270182>

[Accessed: 15 Feb 2025]

Raymer, D. (2018) *Aircraft Design: A Conceptual Approach, Sixth Edition*. Reston, VA: AIAA Education Series.

Rosenblatt, F. (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 (6), 386.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) Learning representations by back-propagating errors. *nature*, 323 (6088), 533-536.

Russell, S. and Norvig, P. (2021) *Artificial Intelligence: A Modern Approach*. 4 ed. Harlow, UNITED KINGDOM: Pearson.

Saleh, R.A. and Saleh, A. (2022) Statistical properties of the log-cosh loss function used in machine learning. *arXiv preprint arXiv:2208.04564*.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E. and Valaee, S. (2017) Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*.

Salunke, N.P., Ahamad, R.J. and Channiwala, S. (2014) Airfoil parameterization techniques: A review. *American Journal of Mechanical Engineering*, 2 (4), 99-102.

Selig, M. (2010) *UIUC Airfoil Data Site* [online]

Available at: [https://m-selig.ae.illinois.edu/ads/coord\\_database.html](https://m-selig.ae.illinois.edu/ads/coord_database.html)

[Accessed: 15/03]

Selig, M.a.G.J.a.B.A.a.G.P. (1996) *Summary of Low Speed Airfoil Data, Volume 1*.

Sharpe, P. and Hansman, R. (2023) NeuralFoil: An airfoil aerodynamics analysis tool using physics-informed machine learning. [online]

Available at: <https://github.com/peterdsharpe/NeuralFoil/blob/master/paper/out/main.pdf>

Sharpe, P.D. and Hansman, R.J. (2021) *Aerosandbox: A differentiable framework for aircraft design optimization*thesis.

Shivamoggi, B.K. (2021) *Introduction to theoretical and mathematical fluid dynamics*. Hoboken, New Jersey: John Wiley & Sons.

Sobieczky, H. (1999) Parametric airfoils and wings. In: (ed.) *Recent development of aerodynamic design methodologies: inverse design and optimization*. Springer. pp. 71-87.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15 (1), 1929-1958.

Terven, J., Cordova-Esparza, D.M., Ramirez-Pedraza, A., Chavez-Urbiola, E.A. and Romero-Gonzalez, J.A. (2023) Loss functions and metrics in deep learning. *arXiv preprint arXiv:2307.02694*.

Tortora, G., Concilio, A. and Pecora, R. (2023) Airfoil Shape Morphing through a Novel Parameterization and Fitting Optimization Method Based on Uniform Non-Rational B-Spline Functions. *Designs*, 7 (1), 28.

Tyrrell, D. (2023) Airfoil Performance Prediction Through an Artificial Neural Network.

Van Dam, C.P., Associates, V.R. and Center, L.R. (1984) *Natural Laminar Flow Airfoil Design Considerations for Winglets on Low-speed Airplanes*. National Aeronautics and Space Administration, Scientific and Technical Information Branch.

Veludurthi, A.a.B.V. (2020) Experimental Study on Modal and Harmonic Analysis of Small Wind Turbine Blades Using NACA 63-415 Aerofoil Cross-Section. *Energy Engineering*, 117, 49-61.

Versteeg, H.K. and Malalasekera, W. (2007) *An introduction to computational fluid dynamics : the finite volume method*. 2nd ed. Harlow: Pearson Prentice Hall.

Vinuesa, R. and Brunton, S.L. (2022) Enhancing computational fluid dynamics with machine learning. *Nature Computational Science*, 2 (6), 358-366.

Wang, H., Cao, Y., Huang, Z., Liu, Y., Hu, P., Luo, X., Song, Z., Zhao, W., Liu, J. and Sun, J. (2024) Recent advances on machine learning for computational fluid dynamics: A survey. *arXiv preprint arXiv:2408.12171*.

Wang, Q., Ma, Y., Zhao, K. and Tian, Y. (2022) A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*, 9 (2), 187-212.

White, F.M. (2011) *Fluid mechanics*. 7th ed. New York, NY: McGraw-Hill.

Wilcox, D.C. (2006) *Turbulence Modeling for CFD*.

Woo Oh, H. and Oh, H.W. (2010) *Computational fluid dynamics*. Rijeka, Croatia: IntechOpen.

Wright, S. (1921) Correlation and causation. *Journal of agricultural research*, 20 (7), 557.

Yu, B., Xie, L. and Wang, F. (2019) An improved deep convolutional neural network to predict airfoil lift coefficient. *International Conference on Aerospace System Science and Engineering* [online], 275-286

Available at: <https://www.utias.utoronto.ca/wp-content/uploads/2019/07/19-An-Improved-Deep-Convolutional-Neural-Network.pdf>

[Accessed: 29 Jan 2025]

Zhang, W., Shen, X., Zhang, H., Yin, Z., Sun, J., Zhang, X. and Zou, L. (2024) Feature importance measure of a multilayer perceptron based on the presingle-connection layer. *Knowledge and Information Systems*, 66 (1), 511-533.

Zhou, J., Zhang, R. and Chen, L. (2025) Prediction of Airfoil Lift Coefficient Based on Multilayer Perceptron. *International Journal of Aeronautical and Space Sciences*, 26 (1), 13-20.

## Appendix 1

Family	Train Count	Test Count	Train Percentage	Test Percentage
GOE	322	74	81.31%	18.69%
E	146	43	77.25%	22.75%
NACA	124	24	83.78%	16.22%
FX	89	19	82.41%	17.59%
S	57	11	83.82%	16.18%
MH	42	8	84.00%	16.00%
HQ	33	11	75.00%	25.00%
AH	28	11	71.79%	28.21%
AG	25	4	86.21%	13.79%
M	22	5	81.48%	18.52%
SC	21	5	80.77%	19.23%
USA	18	8	69.23%	30.77%
SD	16	1	94.12%	5.88%
RAF	14	4	77.78%	22.22%
RG	11	1	91.67%	8.33%
B	11	1	91.67%	8.33%
C	10	1	90.91%	9.09%
RC	9	2	81.82%	18.18%
CLARK	9	2	81.82%	18.18%
RAE	8	3	72.73%	27.27%
EH	6	3	66.67%	33.33%

HS	6	0	100.00%	0.00%
VR	6	6	50.00%	50.00%
SG	6	0	100.00%	0.00%
LS	6	1	85.71%	14.29%
LWK	5	0	100.00%	0.00%
V	5	0	100.00%	0.00%
SA	4	2	66.67%	33.33%
HOR	4	0	100.00%	0.00%
HT	4	1	80.00%	20.00%
RHODESG	4	0	100.00%	0.00%
R	4	2	66.67%	33.33%
KC	4	1	80.00%	20.00%
MARSKE	4	2	66.67%	33.33%
EIFFEL	4	1	80.00%	20.00%
DAE	4	0	100.00%	0.00%
OA	4	0	100.00%	0.00%
NPL	4	1	80.00%	20.00%
UA	4	1	80.00%	20.00%
COANDA	3	0	100.00%	0.00%
A	3	0	100.00%	0.00%
YS	3	0	100.00%	0.00%
NLF	3	3	50.00%	50.00%
DEFCND	3	0	100.00%	0.00%
ARAD	3	1	75.00%	25.00%
LA	3	0	100.00%	0.00%
P	3	1	75.00%	25.00%
EA	3	0	100.00%	0.00%
STCYR	3	1	75.00%	25.00%
AS	3	0	100.00%	0.00%
DGA	2	0	100.00%	0.00%
AMES	2	1	66.67%	33.33%
AMSOIL	2	0	100.00%	0.00%
BE	2	0	100.00%	0.00%
K	2	3	40.00%	60.00%
LRN	2	0	100.00%	0.00%
ISA	2	2	50.00%	50.00%
MA	2	0	100.00%	0.00%
DU	2	1	66.67%	33.33%
SSCA	2	0	100.00%	0.00%
SUPERMARINE	2	0	100.00%	0.00%
DSMA	2	0	100.00%	0.00%
DF	2	0	100.00%	0.00%
NL	2	0	100.00%	0.00%
OAF	2	3	40.00%	60.00%
NLR	2	0	100.00%	0.00%
TEMPEST	2	1	66.67%	33.33%

WB	2	0	100.00%	0.00%
PSU	2	0	100.00%	0.00%
TSAGI	2	0	100.00%	0.00%
GEMINISM	1	0	100.00%	0.00%
GIHA	1	0	100.00%	0.00%
TH	1	0	100.00%	0.00%
STF	1	0	100.00%	0.00%
PFCM	1	0	100.00%	0.00%
RCSC	1	0	100.00%	0.00%
SPICASM	1	0	100.00%	0.00%
CAST	1	0	100.00%	0.00%
BAMBINO	1	0	100.00%	0.00%
BACXXX	1	0	100.00%	0.00%
CAP	1	0	100.00%	0.00%
BQM	1	0	100.00%	0.00%
DAYTONWRIGHTT	1	0	100.00%	0.00%
DBLN	1	0	100.00%	0.00%
ATR	1	0	100.00%	0.00%
AUGUST	1	0	100.00%	0.00%
AVISTAR	1	0	100.00%	0.00%
BOE	1	1	50.00%	50.00%
AQUILASM	1	0	100.00%	0.00%
BW	1	0	100.00%	0.00%
BACJ	1	0	100.00%	0.00%
DFVLRR	1	0	100.00%	0.00%
CURTISC	1	0	100.00%	0.00%
DAVIS	2	0	100.00%	0.00%
CR	1	1	50.00%	50.00%
DAYTONWRIGHT	1	0	100.00%	0.00%
DOA	1	0	100.00%	0.00%
DH	1	0	100.00%	0.00%
COOTIE	1	0	100.00%	0.00%
GIHI	1	0	100.00%	0.00%
FXL	1	0	100.00%	0.00%
FALCON	1	0	100.00%	0.00%
FAUVEL	1	0	100.00%	0.00%
FG	1	3	25.00%	75.00%
FXS	1	2	33.33%	66.67%
GIHIB	1	0	100.00%	0.00%
ESA	1	0	100.00%	0.00%
EC	1	0	100.00%	0.00%
WHITCOMB	1	0	100.00%	0.00%
ULTIMATE	1	0	100.00%	0.00%
UI	1	0	100.00%	0.00%
UAG	1	0	100.00%	0.00%
SM	1	0	100.00%	0.00%

SARATOV	1	0	100.00%	0.00%
SOKOLOV	1	0	100.00%	0.00%
STE	1	1	50.00%	50.00%
MUE	1	0	100.00%	0.00%
HOBIE	1	0	100.00%	0.00%
HH	1	0	100.00%	0.00%
GS	1	0	100.00%	0.00%
HOBIESM	1	0	100.00%	0.00%
LDS	1	0	100.00%	0.00%
MARSDEN	1	0	100.00%	0.00%
GRIFFITH	1	0	100.00%	0.00%
GIHJ	1	0	100.00%	0.00%
GIHK	1	0	100.00%	0.00%
GIHC	1	0	100.00%	0.00%
GIHD	1	0	100.00%	0.00%
GIHE	1	0	100.00%	0.00%
GIHF	1	0	100.00%	0.00%
GIHG	1	0	100.00%	0.00%
GIHN	1	0	100.00%	0.00%
GIHM	1	0	100.00%	0.00%
GLENNMARTIN	1	2	33.33%	66.67%
GIHL	1	0	100.00%	0.00%
KENMAR	1	0	100.00%	0.00%
L	1	3	25.00%	75.00%
JOUKOWSK	1	0	100.00%	0.00%
GM	1	0	100.00%	0.00%
USNPS	1	0	100.00%	0.00%
MB	1	0	100.00%	0.00%
MILEY	1	0	100.00%	0.00%
MRC	1	1	50.00%	50.00%
NN	1	0	100.00%	0.00%
PT	1	0	100.00%	0.00%
PMC	1	0	100.00%	0.00%
TRAINER	1	0	100.00%	0.00%
DRGNFLY	0	1	0.00%	100.00%
DORMOY	0	1	0.00%	100.00%
CH	0	1	0.00%	100.00%
CHEN	0	1	0.00%	100.00%
BACNLF	0	1	0.00%	100.00%
APEX	0	1	0.00%	100.00%
FXLV	0	1	0.00%	100.00%
LVN	0	1	0.00%	100.00%
LG	0	1	0.00%	100.00%
J	0	1	0.00%	100.00%
JN	0	1	0.00%	100.00%
GU	0	1	0.00%	100.00%



HSNLF	0	1	0.00%	100.00%
EBAMBINO	0	1	0.00%	100.00%
GIIH	0	1	0.00%	100.00%
FXM	0	1	0.00%	100.00%
MS	0	2	0.00%	100.00%
NCAMBRE	0	1	0.00%	100.00%
NASASC	0	1	0.00%	100.00%
NPLX	0	1	0.00%	100.00%
STRAND	0	1	0.00%	100.00%
US	0	1	0.00%	100.00%
WASPSM	0	1	0.00%	100.00%

Table 3: Train-Test Split

---

## Appendix 2

The core components of this project, including the full source code of all functions used throughout, preprocessing, model training, and evaluation scripts, are available in the public GitHub repository:

[https://github.com/bengawith/curve\\_gen/](https://github.com/bengawith/curve_gen/)

to clone this repository, use the following command:

```
git clone https://github.com/bengawith/curve_gen.git
```

For further details on running the scripts and reproducing results, please refer to the README.md file in the repository.