ABSTRACT

Title of Dissertation: A META-LEARNING BASED
AERODYNAMIC ANALYSIS FRAMEWORK
FOR WIND TURBINE DESIGN
APPLICATIONS

Koushik Marepally
Doctor of Philosophy, 2023

Dissertation Directed by: Professor James D Baeder
Department of Aerospace Engineering

Design testing and analysis is a major bottleneck in the design process of wind turbine applications, mainly due to the computational cost of analysis tools like computational fluid dynamics (CFD). Furthermore, the accuracy of the state-of-the-art turbulence models is low in flows with high adverse pressure gradients such as airfoils operating at high angles of attack. This study aims to develop an aerodynamic analysis framework for wind turbine airfoils with both improved cost and improved accuracy to use in design applications.

An artificial neural network-based data-driven surrogate model is developed to predict the aerodynamic performance quantities of lift coefficient, lift-to-drag ratio, and pitching moment coefficient for wind turbine airfoils. An efficient geometric space exploration strategy is used to generate a representative database of wind turbine airfoils and their corresponding performance quantities. The developed surrogate model shows a uniform accuracy across a wide range of wind turbine airfoil geometries, with an L_2 error estimates of 0.03 in lift coefficient, 0.4 in lift-to-drag ratio, and 0.003 in pitching moment coefficient. These errors correspond to less than 2%

magnitudes of the corresponding performance quantities at the design point. With a benefit of more than six orders of magnitude in computational cost compared to CFD, the surrogate model has the capabilities to be embedded in uncertainty quantification (UQ) and multidisciplinary design analysis and optimization (MDAO) frameworks.

To reduce the model development cost, various parameter space exploration and reduction strategies are tested to benchmark the impact of reducing the training data on the accuracy of the surrogate model. With uniform data puncturing style, the accuracy level of the surrogate model is maintained even with up to a 50% reduction in the training data.

The propagation of uncertainty from the geometric parameters of the airfoils to the airfoil performance quantities is quantified using the surrogate model coupled with a Monte-Carlo-based UQ framework. The performance quantities show an uncertainty of about 3% of their magnitude for a 5% geometric uncertainty near the operational angle of attack and more than 10% magnitude of uncertainty near the stall angle of attack.

Secondly, field inversion machine learning (FIML) methodology is applied on multiple airfoils to arrive at a model consistent correction to the turbulence model for improved airfoil stall predictions. The corrected turbulence model shows a consistent improvement of the stall lift predictions with an improvement in stall angle of attack by more than 35% and stall lift coefficient by more than 40%. Besides the lift coefficient, the corrected turbulence model predicts the surface pressure and flow separation point more accurately.

A meta-learning model is developed using the corrected turbulence model on the database of wind turbine airfoils, which is both computationally inexpensive and closer to the experimental data. The model is integrated with an evolutionary optimization framework and tested on various airfoil design problems, including airfoil drag minimization by 5% and stall delay by 1°.

A META-LEARNING BASED AERODYNAMIC ANALYSIS FRAMEWORK FOR WIND TURBINE DESIGN APPLICATIONS

by

Koushik Marepally

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
- Professor James D Baeder, Chair/Advisor
- Associate Professor Anubhav Datta
- Professor Inderjit Chopra
- Professor Pino Martin
- Professor Ricardo H Nochetto
- Assistant Professor Deep Ray

Dedicated to

my parents, Mr and Mrs Jayasree Ravi Kumar*, for all their sacrifices and love,*
my brother, Varun Kumar*, for his unmatching support,*
my grand father, Shri. B Venkatacharya*, for inspiring me to become an engineer,*
Professor Gopal R Shevare*, for igniting interdisciplinary thought process in me.*

Acknowledgments

I would like to take this opportunity to thank each and everyone whose contributions have either directly or indirectly shaped this dissertation. While I am extremely grateful to all of them, it is impossible to name each person here. Nonetheless, I would like to mention a few people who have made my research journey very enjoyable and an intellectually stimulating experience.

Firstly, I would like to thank my doctoral advisor, Dr James Baeder. I was always told that a PhD journey is greatly dependent of the advisor, and thanks to my advisor, my journey has been nothing less than extraordinary. I was fortunate to have a lot of discussions with him, both academic and life related. I admire his research spirit and ideas, and the fact that he always respects his students' ideas. The diverse research in the lab group is undeniably due to his expertise and enthusiasm to wet his foot in interdisciplinary research, and I'm fortunate to be a part of the applied machine learning research here. I greatly value his patience and support during the days of unfruitful research in my PhD timeline. I mainly enjoy my talks with him especially the ones where we discuss the big picture and implications of various research ideas. In short, I can say that he's the best advisor I could ask for. I aim to inculcate several of his abilities to shape me up as a good mentor.

Further, I would like to thank Dr Ganesh Vijayakumar, my mentor at NREL. I was fortunate to have participated in an internship for ten months with NREL, pursuing some aspects of this dissertation research. Ganesh was very helpful, especially to tackle the software implementation and development aspects.

Next, I would thank my doctoral committee members for taking their time to review my research and for their inputs during different research milestones. I would specifically mention Dr Nochetto for his advises on general research practices and time keeping, and for his participation despite his busy schedule. I also thank Dr Ray for an extremely detailed review of my research and dissertation. Based on his feedback, I was able to make this dissertation more clear and mathematically sound.

I would also thank Dr V T Nagaraj for his extremely valuable and practical advises. His ideology and research insights are practical and relevant to design engineering, and shaped up some aspects of my research.

I also would like to thank the staff of NREL's HPC resource (Eagle) and UMD's HPC resources (Deepthought2 and Zaratan). These resources were an integral part of my research, especially for generating the massively large database of CFD solutions.

I am indebted to my research colleagues Dr Bumseok Lee, Dr Yong Su, Dilhara, Paulo, Miranda, Apurva and Muneeb. Bumseok was extremely helpful in various aspects of my research and helped me to get familiarized to the lab's in-house CFD solution frameworks. He also helped me with automation process, which was valuable for timely progress in my research. I have collaborated with Apurva and Muneeb and learnt certain modern machine learning frameworks. I also had some interesting discussions with other members of the the rotorcraft center.

Aerospace engineering staff has been very helpful to me and I could knock on their door for any administrative work or a friendly discussion. Gabbie, Dana, Leslie and Renee made sure that I had a timely response for any of my requests and to support traveling to conferences for presenting my research.

I would also thank a long list of my roommates and friends at UMD: Mrinalgouda Patil, Nikhil Subraveti, Lavan Teja Burra, Abhishek Shastry, Shashank Maurya, Hema Choudhary, Manidhar Vutla, Mrugandha Namjoshi, Nikhil Bharadwaj and Lohith Madem. Mrinal was my friend from undergraduate, and he along with Nikhil and Abhishek made sure I felt UMD like my home. Badminton with Nikhil and Shashank was my go to for balancing work pressure and maintaining inner peace. I enjoyed my discussions with Manidhar and Mrugandha on several topics related to technology, spirituality and movies. Lavan, Nikhil Bharadwaj and Lohith were extremely bearing with me, especially during the last phase of my research. Nikhil Bharadwaj was also my swimming and billiards mate, which was my stress buster during the dissertation writing phase.

Vinod has been a great source of support to me despite the long distance between India and the US. He was a moral support at times of trouble in research or adjusting in the US. Same goes for Sivaji, Dileep and Manoj. With the COVID pandemic occupying nearly half of my PhD duration, continuous communication and online games with them helped me to be productive in the working times.

Dr G R Shevare, Dr Guruprasad, Dr Sandip Singh (from IIT Bombay) and Dr Vinod Kaushik (from ISRO) greatly helped me during the start of my research journey. Discussions with Dr Shevare are what brought me into the world of design engineering, which was further strengthened by Dr Nagaraj.

My acknowledgments will be definitely incomplete without mentioning my family. It is with the immense support of my parents and my brother, I am what I am today, and no words can do justice to recognizing their contributions. My mother (Jayasree, a science teacher) and my brother (Varun Kumar, an engineer) actively participated in stimulating discussions with me, and contributed to shaping up my dissertation. My brother helped me with software implementation and development aspects which were integral part of my research work. My dad (Ravi Kumar, an advocate) has inculcated the practice of debating with reasonable argument in me. It is all their sacrifices and the values they instilled in me, that made me take this step in my life. I would also mention the latest member of my family, Vaikhaan (my nephew) who was born during the last months of my PhD. Video conferencing with him was an absolute pleasure booster during the peak time of my dissertation writing, and he must receive a special mention in this dissertation.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction and Overview

1.1: Motivation

Design engineering is formally defined as “the series of steps used in creating functional products and processes”. The series of steps involved in a design process are establishing the design requirements, defining a design concept, analysis and testing of designs, and refinement of designs using optimization. Aerospace design engineering involves multiple disciplines of physics and engineering like aerodynamics, structural mechanics, control systems and often other disciplines related to the particular design concept. For example, design of wind turbines essentially involves these disciplines of aerospace engineering, along with the fields of power generation, climate studies etc. A major bottle-neck of the design process is the analysis of the design concepts, which usually requires either experiments or mathematical models that can predict the performance of the designs and aid in the refinement process.

Advancements of modeling and simulation tools have greatly revolutionized the design analysis process, with no exception to the field of aerodynamics. With the developments in high performance computing architectures, usage of the simulation tools like computational fluid dynamics (CFD) has drastically reduced the wind tunnel testing overhead [1, 2]. Additionally, CFD is also used to enhance the understandings of critical flow phenomena that can limit the performance of aerospace vehicles or components, opening new frontiers in the aerospace design engineering [3].

Usage of CFD tools to model the aerodynamics of a wind turbine has been applied in many strategies from the analysis of 2D blades with blade element methods to the simulation of 3D rotor blades, from design conditions to extreme weather conditions etc [4–8]. Despite the success in demonstrating the usage of the high-fidelity simulation techniques to model the aerodynamics of wind turbines, it is impractical to use these tools actively in the design optimization loop due to their computational costs. Several factors contribute to increasing the cost of analysis step in the design process, and are discussed in the next section.

1.1.1: Grand Challenges in Aerospace Design Engineering

Design analysis and optimization loop is an exhaustive process involving analysis of a wide range of designs. The cost of this step in the design process is governed by several factors such as the expanse of the design space, cost of the analysis tool for each design evaluation, requirements of multi-disciplinary focus etc. Each of these factors independently affect the overall design analysis timeline, and are required to be addressed for an efficient, cost-effective and timely design process.

1.1.1.1: Curse of Dimensionality

"Curse of Dimensionality" in the context of design engineering is associated with the exploding size of design space due to increasing number of design variables. For example, the number of variables required to define a wind turbine is greater than that required to define an airfoil cross section. As a result, the number of wind turbine designs to be analysed becomes far more greater than the number of airfoil designs.

As the number of design variables or the design complexity increases, the size of the design space grows exponentially. This makes the design space exploration difficult, often leading to sub-optimal design decisions.

Design of Experiments (DoE) techniques can be used to efficiently explore the design space with relatively less number of designs. However, the total number of design evaluations still increases with the number of design variables, increasing the total cost of the analysis and optimization loop.

1.1.1.2: Complexity and Fidelity of the Physics Model

A wide variety of analysis tools are used to perform the design analysis steps. In CFD, these tools range from simple linear models and potential flow solutions codes to high fidelity scale resolving tools like large-eddy simulation (LES) and direct numerical simulations (DNS).

While high fidelity tools are powerful and accurate models to evaluate the designs, they are extremely expensive. As a result, usage of high-fidelity tools for optimization significantly limits the number of designs that can be explored. On the contrary, low fidelity models are computationally less expensive, and can be used on a large design space.

Often, the design process involves the usage of both low fidelity and high fidelity models at different stages of the optimization process. Low fidelity tools are used to exhaustively explore the feasible design space and high fidelity tools are used in a limited region that has higher probability of containing an optimal design.

While the hybrid approach is efficient, the decision of when to switch the analysis model is fuzzy and depends on the particular design problem.

1.1.1.3: Multidisciplinary Analysis

Over the recent years, a major shift of focus is seen into multidisciplinary analysis and optimization (MDAO). Since the design requirements and constraints are governed by multiple disciplines, separate design analysis of one discipline may result in counter-indicative design decisions compared to other disciplines. For example, for a wind turbine, aerodynamic analysis may indicate that power generation efficiency improves with a higher wind speed while the structural considerations limit the blade loads. Thus, MDAO becomes necessary for almost all the design problems.

A major problem associated with MDAO is the integration of different analysis tools. The level of tightness in the coupling of the physics governs the integration process. Further, the computational complexity of each design evaluation becomes higher due to an integrated cost of analysis.

1.1.1.4: Uncertainty Quantification

An important aspect of the design process is robustness of the final design, i.e., a robust design with sub-optimal performance is preferred over a highly uncertain design with better performance. The uncertainties in the performance are ubiquitous and can occur in various forms such as variability in design manufacture, operating conditions, model parameters etc. The analysis tools must account for the propagation of the sources of uncertainty into the performance.

Analysis with uncertainty quantification (UQ) adds more cost to the design evaluation steps. The cost depends on the uncertainty model incorporated into the analysis toolchain.

Summary of the Grand Challenges

All the grand challenges are qualitatively illustrated together in Figure 1.1. The vertical axis represents cost of the design evaluation process, and the horizontal axis is the complexity of the physics model. The three curves denoted as CFD, MDAO and MDAO+UQ correspond to single discipline analysis, multidisciplinary analysis and multidisciplinary analysis with uncertainty quantification. Along each curve is the design complexity, defined by the arcs showing an airfoil and a wind turbine.

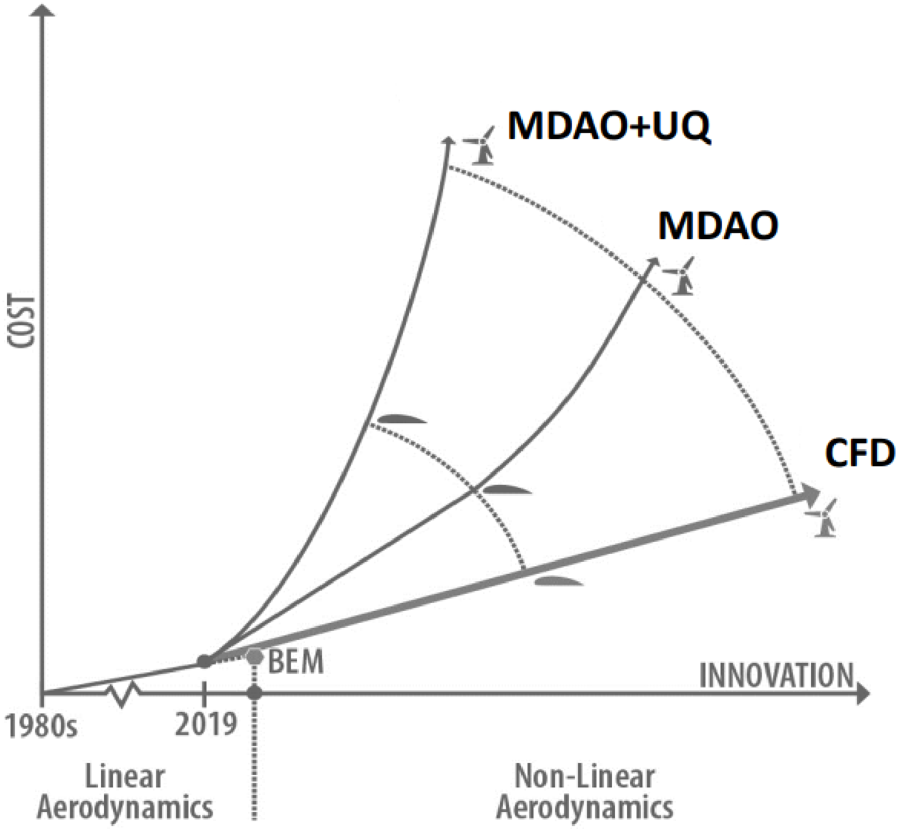

Figure 1.1: Illustration of the grand challenges in aerospace design engineering

In summary, the cost of design analysis and optimization increases with increasing design complexity or model fidelity. It further increases with multidisciplinary focus or uncertainty capabilities, and the rate of increase is also higher.

1.1.2: Potential for Machine Learning: “The Big Picture”

Machine learning is a branch of data-driven techniques that can be used to learn from and make predictions based on data. There has been increased usage of machine learning in the fields of engineering for predictive modeling of various phenomena of engineering importance.

In the context of the grand challenges, machine learning can be applied to address multiple challenges by reducing the cost of analysis as well as increasing the fidelity of the physical model. The potential application of machine learning lies in the usage of limited high fidelity data to build a accurate and less expensive design analysis model that can further support multidisciplinary analysis and uncertainty quantification capability.

This task can be broken down into two sub-domains of machine learning, each with its own techniques to make an overall cost efficient aerodynamic analysis model.

1.1.2.1: Data-driven Surrogate Modeling

High fidelity CFD methods are powerful tools to approximate the flow behavior accurately. However, from a design point of view, only certain quantities of interest are used to make design and optimization decisions. The information intensive CFD computations can be replaced with lower order models that can estimate the quantities of interest. These models only estimate the quantities of interest and do not compute any additional information, making them cost effective.

Surrogate models are formally defined as cost efficient engineering models that estimate certain quantities of interest, which are otherwise extremely expensive or difficult to measure or compute. These models are built based on the data of the quantities of interest rather than on the physical models. The benefit of using data-driven methods is that the developed surrogate model is true to the data used for learning, and hence the accuracy of high fidelity models can be achieved.

Surrogate modeling is extensively used in various engineering applications ranging from simple empirical relations to modern machine learning architectures.

This sub-domain involves exploring efficient design of experiments strategies and response surface modeling techniques.

1.1.2.2: Physics-informed Modeling

This is the other end of machine learning applications, where high-fidelity or experimental data is used to improve the physics model using machine learning. In the case of CFD, all the RANS turbulence closure models are calibrated using some physics modeling and some empirical model constants. These constants are determined based on the experimental data available, which is a conventional learning approach limited by the ease of the human modeler. With the development of modern machine learning techniques, this process can be used to develop model consistent improvements that can enhance the quality of CFD predictions.

This sub-domain involves development and formulation of model consistent functional corrections, techniques on how to use the limited high fidelity data to translate into learnable functional corrections and feature space engineering.

1.1.2.3: “The Big Picture”

In summary, the potential for machine learning to address the grand challenges can be visualized in Figure 1.2.

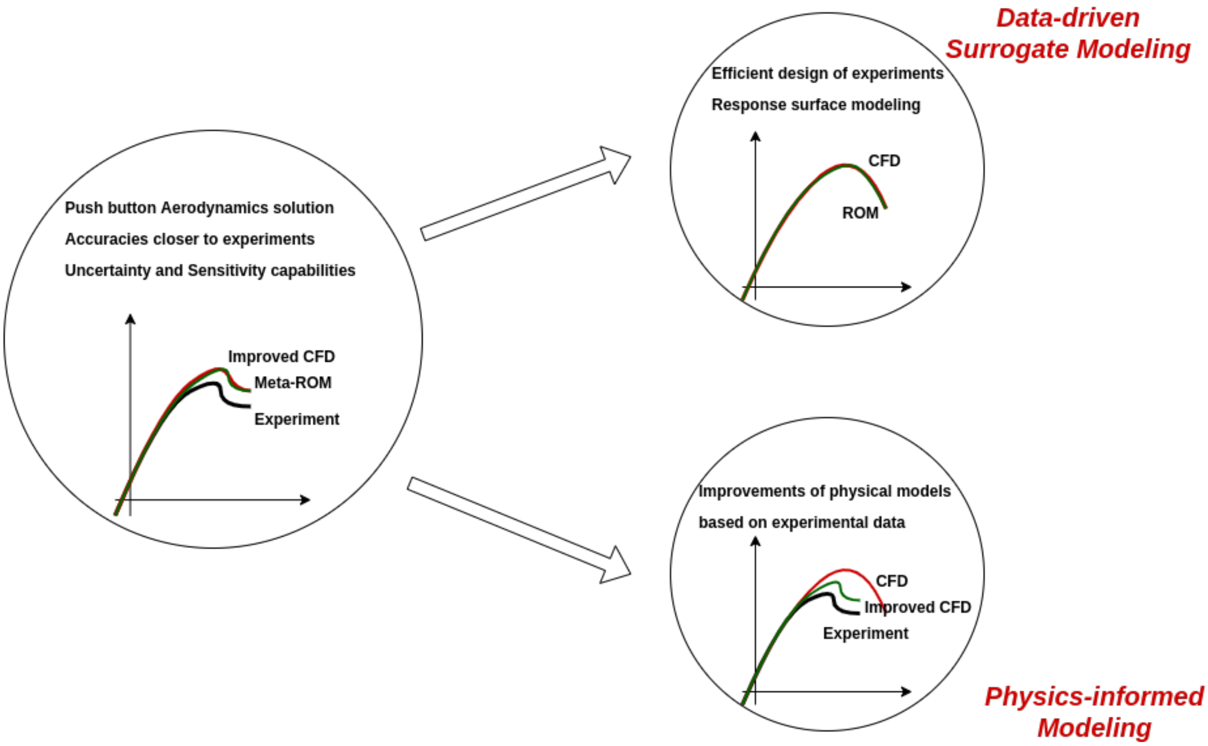

Figure 1.2: Illustration showing the potential for machine learning to address the grand challenges

The main goal of machine learning in this context is to achieve a low cost aerodynamics solution tool that has accuracy closer to experiments and can be used together with uncertainty quantification frameworks. This can be achieved by integrating the outcomes of the two sub tasks of data-driven surrogate modeling and physics-informed modeling.

1.2: Background - Data-driven Surrogate Modeling

Data-driven surrogate modeling is extensively used in a variety of engineering applications from the simplistic empirical relations to the modern machine learning architectures. Expensive simulations are replaced with surrogate models for various applications like design optimization, sensitivity analysis, real-time control, digital twin engineering etc.

1.2.1: Uncertainty Quantification and Sensitivity Analysis

Uncertainty quantification is one of the major application areas of surrogate modeling. With uncertainties inherently present in the operating conditions and design manufacture for aerospace applications, various studies have extensively focused on quantifying the sensitivity of the aerodynamic performance characteristics with uncertainties in various parameters such as freestream quantities, shapes, component erosions etc.

Lockwood et al. [9, 10] demonstrated the usage of surrogate models for quantifying the uncertainties of hypersonic viscous flows and compared the sensitivities with gradient based methods. Usage of surrogate models with Monte-Carlo simulations showed similar predictions of sensitivity as gradient based estimates, but with significantly lower computational cost. Kwon [11] developed an uncertainty quantification framework to measure the uncertainty in energy production of wind turbines due to variability in wind resources using Monte-Carlo simulations with a surrogate model. Zhang [12] developed a domain segmentation framework based on uncertainties in surrogate models to characterize the uncertainties in wind farms and improved the confidence of the developed surrogate models. Atif [13] used a radial basis function network based surrogate model to quantify the effects of varying freestream turbulence intensity on airfoil performance. Kinshuk et al. [14] used multifidelity methods to quantify the uncertainty of windfarms.

DeGennaro et al. [15] and Tabatabaei et al. [16] studied the effects of icing on the airfoil performance characteristics and the latter quantified the losses in wind turbine performance. Xiaojing et al. [17, 18] studied the effects of geometric uncertainty on the airfoil performance, identifying that the uncertainty region moves downstream as the flow changes from subsonic to transonic. Francesco et al. [19] studied the effects of damages on leading and trailing surfaces of wind turbine blades on the performance.

The majority of the studies focused on developing a local surrogate model around the design point of interest and quantified the uncertainties of the respective quantities of interest. Moin et al. [20] developed a surrogate model for airfoils representing them using control points, and trained on the data with NACA airfoils. Their model accurately predicted the lift and drag characteristics of a wide range of airfoils.

1.2.2: Design Optimization

Surrogate models can significantly accelerate the design optimization process by reducing the computational costs associated with the design analysis phase. Various studies have developed surrogate models to perform aerodynamic design optimizations.

Mack [21] developed a framework for surrogate based design optimization as one of the primary proof of concepts in aerospace design engineering and validated the framework for single objective and multi-objective optimizations. Aircraft design engineeering was pioneered by Alonso in several of his works [22–25], with the latest works focusing on a reliable design process using surrogate models. Hu et al. [26] performed reliability based optimization of wind turbine blades under uncertain dynamic wind loads and designed a 5MW wind turbine blade with target probability of failure of 2.275%. Lin et al. [27] developed a multi-fidelity surrogate modeling based optimization framework and showed the approach to be more efficient and robust than one shot sampling methods for design optimization problems.

In summary, various studies used a wide variety of surrogate modeling techniques from simple response surface models to modern machine learning architectures for design applications, from airfoil shape optimization to multi-objective multi-component designs.

1.2.3: Real-time Control

Another major application that requires a push-button model computation is real-time control and simulation. These applications require very simplistic surrogate models to assist the need for extremely low computational cost. Real-time control is applied in multiple scenarios that require quick decisions such as gust response control, aircraft traffic control etc.

Matsuno [28] developed a near-optimal control strategy to mitigate aircraft traffic conflict using polynomial chaos kriging based surrogate modeling. This is a relatively new surrogate modeling technique that uses non-intrusive polynomial chaos expansion methods together with surrogate modeling. Bird et al. [29] used dimensionality reduction based surrogate models for compressor flows. The dimensionality reduction techniques aided in achieving very simple architectures for the surrogate model.

Techniques applied in surrogates for real-time control achieve the least model complexity and can be used to significantly lower the computational cost of the design process.

1.2.4: Digital Twin Engineering

Digital twin engineering is a technique to create a digital replica of a physical system or process. It can be considered a bi-directional learning where the digital model actively learns about the state of the physical system and adjusts the model, as well as the physical system is optimized using the predictions of the digital twin. This has major applications in life-cycle engineering.

Kraft [30] developed a digital twin for life cycle integration where the techniques to combine experimental data with computational tools are investigated. Renganathan [31] developed a data fusion strategy to extract more information from different aerodynamic datasets generated using different types of computational or experimental methods. Bayesian frameworks are used to maximize the likelihood of the true data from the fusion of the data components. Berri [32] used surrogate models combined with real-time health monitoring of aerospace systems to dynamically adapt the fault propagation models.

Techniques applied in digital twin engineering can be used in conjunction with the surrogate model for active design optimization or life cycle assessment.

1.2.5: Takeaways

Several literature sources employing surrogate models for a variety of applications are studied. While each of the studies achieved computational benefits from using the surrogate models, efforts to develop global surrogate models that can act as a general aerodynamics solution suffered from low accuracies compared to their true model counterparts.

Efficient design space exploration strategies together with modern machine learning methods has the potential to overcome this drawback, and can aid in the development of a general aerodynamics solution for wind turbine design components. As a proof of concept, the objective of this study is to develop a fast surrogate model to predict the aerodynamic performance quantities of general wind turbine airfoils.

1.3: Background - Physics-informed Modeling

Physics-informed modeling techniques are a class of methods that use physics based models and high-fidelity data to enhance the model by recalibration. These techniques are extremely useful in case of certain physical models with inherent empiricism. For example, Reynolds Averaged Navier-Stokes (RANS) equations require turbulence closure models, which must contain data-driven models to account for unresolved length and time scales. All the turbulence closure models fall into the category of physics informed modeling, since the data is governed by an underlying physical model and calibrated with experimental data.

Spalart and Allmaras [33, 34] developed a one equation model for aerodynamic flows at high Reynolds numbers, and calibrated with flows over airfoils. Lohse and Axel [35] developed anisotropy corrections for the turbulence models. These methods are data-driven, but do not employ modern machine learning techniques for calibration of the models.

1.3.1: Machine Learning based improvements in Turbulence Models

With the developments of applied machine learning in engineering disciplines, a lot of studies focused on predictive modeling for turbulent flows. Cheung [36] developed a Bayesian uncertainty analysis framework and applied it to turbulence modeling. This was the first study on exploring model inadequacy in turbulence with machine learning techniques. A new methodology of field inversion machine learning (FIML) was developed and pioneered by Duraisamy et al. [37–39]. Different forms of model correction are applied to SA model, SST model and achieved improved predictions for flows over airfoils and in channels. Holland et al. [40, 41] improved the FIML approach to embed neural networks directly into the inversion approach. The methodology is tested on S809 airfoil and validated with S809 and S814 airfoils.

1.3.2: Takeaways

All the data-driven machine leraning based turbulence model improvements show a consistent improvement in the predictions for flows similar to the training data. However, they worsen the accuracies for different types of flows [42].

State-of-the-art turbulence models perform poorly for flows over airfoils at high angles of attack. Exclusive improvements to the turbulence model can be trained to enhance these flow predictions using the FIML methodology. The objective of this study is to develop a FIML based correction trained on the flows over airfoils with adverse pressure gradient and flow separation.

1.4: Objectives

The main objective of this study is to develop a computationally inexpensive aerodynamic solution tool to predict the performance characteristics of wind turbine airfoils, with improved predictions near stall. This objective can be broken down into three tasks as listed below.

(I) Develop a robust and cost-efficient surrogate modeling framework for wind turbine airfoils.

(II) Use physics-informed modeling techniques to achieve consistent improvements in CFD predictions for airfoils at high angles of attack.

(III) Integrate the surrogate modeling framework with improved physics model to arrive at a "Meta-learnt" data-driven physics-informed surrogate model applicable for wind turbine airfoil design problems.

For task (I), efficient low-cost design space exploration strategies must be explored to generate a representative database of wind turbine airfoil geometries. The surrogate model must be able to accurately replicate the CFD results for a wide domain of wind turbine airfoils.

For task (II), the field-inversion machine learning methodology should be implemented for a model consistent correction of baseline turbulence model. The one equation Spalart-Allmaras (SA) model is used as the baseline turbulence model for this study. Robust learning methods must be used to achieve a minimal bias of the correction towards training data, so that the model can be generalizable to the best possible range of flow scenarios.

For task (III), the developments in both tasks (I) and (II) must be integrated to develop a "Meta-learnt" surrogate model that is both cost efficient and with predictions reliably closer to the experimental data. The developed surrogate must be tested for compatibility with a design optimization framework.

1.5: Organization of the Dissertation

(1) Chapter 1 discusses the motivation for this study, a literature review, and the research objective of the present work.

(2) Chapter 2 presents a detailed account of the development of a surrogate model to predict the airfoil aerodynamic performance quantities for wind turbine airfoils. The sequential

process of design space exploration, data generation, machine learning model, training and validation are systematically explained.

(3) Chapter 3 explores data puncturing strategy, and investigates the effect of reducing the amount of training data on the accuracy and performance of the surrogate model. Different styles of data puncturing are discussed and modifications to the training process to accommodate the punctured datasets are developed and presented in this chapter.

(4) Chapter 4 discusses the quantification of uncertainty using the develped surrogate model. The propagation of uncertainty into the performance quantities is investigated for uncertainties in airfoil geometries.

(5) Chapter 5 discusses the field inversion machine learning approach applied to wind turbine airfoils at high angles of attack. A consistent improvement in the prediction of the stall angle of attack and stall lift coefficient is achieved and compared to the baseline SA turbulence model.

(6) Chapter 6 contains the details on the development of a combined surrogate model that uses the surrogate modeling techniques from Chapters 2 and 3 and combines with the improved physics model in Chapter 5. The developed model is integrated with a design optimization framework and tested on sevaral design problems.

(7) The final chapter presents the overall summary of the present work and the key conclusions obtained, with some recommendations for future extensions.

Chapter 2: Surrogate Model for Airfoil Aerodynamics

The state-of-the-practice wind turbine blade design process involves designing the blade using pre-selected airfoil shapes, with Blade Element Momentum theory, to predict the generated power and blade loads under steady-state conditions [43]. Results from the studies of Barrett and Ning [44] show as much as 6.7% increase in energy production of the wind turbines if the design process involved optimization of airfoil shapes together with blade profile. While aerodynamic design optimization can be performed using adjoint methods with RANS based CFD solvers [45], the explosion of the computational cost of these high-fidelity methods creates a bottle-neck to the design process. Data-driven surrogate modeling offers a cost-efficient alternative to address this problem, without largely compromising on the accuracy of the estimates of the performance quantities of interest. This chapter discusses the various steps involved in the development of a surrogate model to predict the aerodynamic performance measures of airfoils. This surrogate model can be used together with blade analysis tools for the optimization of the wind turbine blade.

2.1: Description of the Problem

All wind turbine blades are made up of airfoils sections, which are lofted along its elastic axis. Thus, the main aerodynamic performance measures that drive the design process include the performance indicators of these two dimensional airfoil aerodynamics and the three dimensional effects of axial induction, swirl and other nonlinear effects. The surrogate model for airfoil aerodynamics should predict these airfoil performance indicators for a given input of airfoil geometry. The relevant performance indicators are the static aerodynamic loads. The conventional non-dimensional indicators of these loads are the lift coefficient (c_l), the drag coefficient (c_d) and the pitching moment coefficient (c_m). Aerodynamic efficiency or the lift-to-drag ratio (L/D) is defined as the ratio of the lift and drag coefficients, and is an important parameter in wind turbine design and operation to achieve maximum power with minimum blade loads from thrust [46]. Moreover, from the perspectives of uncertainty, off-design and multi-point optimization, these performance quantities become important for a range of incident angles of attack and Reynolds numbers. Thus, the surrogate model should predict the performance quantities (lift coefficient, lift-to-drag ratio and pitching moment coefficient) for a given geometry of the airfoil for a range of angles of attack and Reynolds numbers.

2.1.1: Mathematical Representation

The surrogate model falls under the class of predictive modeling in supervised machine learning theory. A modern definition of machine learning is given by Tom Mitchell [47] which states “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.” In the current context of the surrogate model for airfoil aerodynamics, the task T is to learn a functional mapping from the airfoil geometries to the aerodynamic performance quantities. The experience E corresponds to the available and known data of aerodynamic performance quantities for a representative population of airfoil geometries. The performance measure P should quantify the extent of the accuracy of the surrogate model compared to the known true data.

The framework describing the learning process for the surrogate model development is shown in Figure 2.1. The task T component of the framework is fixed and the hypothesis, experience and learning algorithm components can be suitably designed to cater the needs of the modeler. The efficiency and accuracy of the developed surrogate model depends on the design and choice of each of these components.

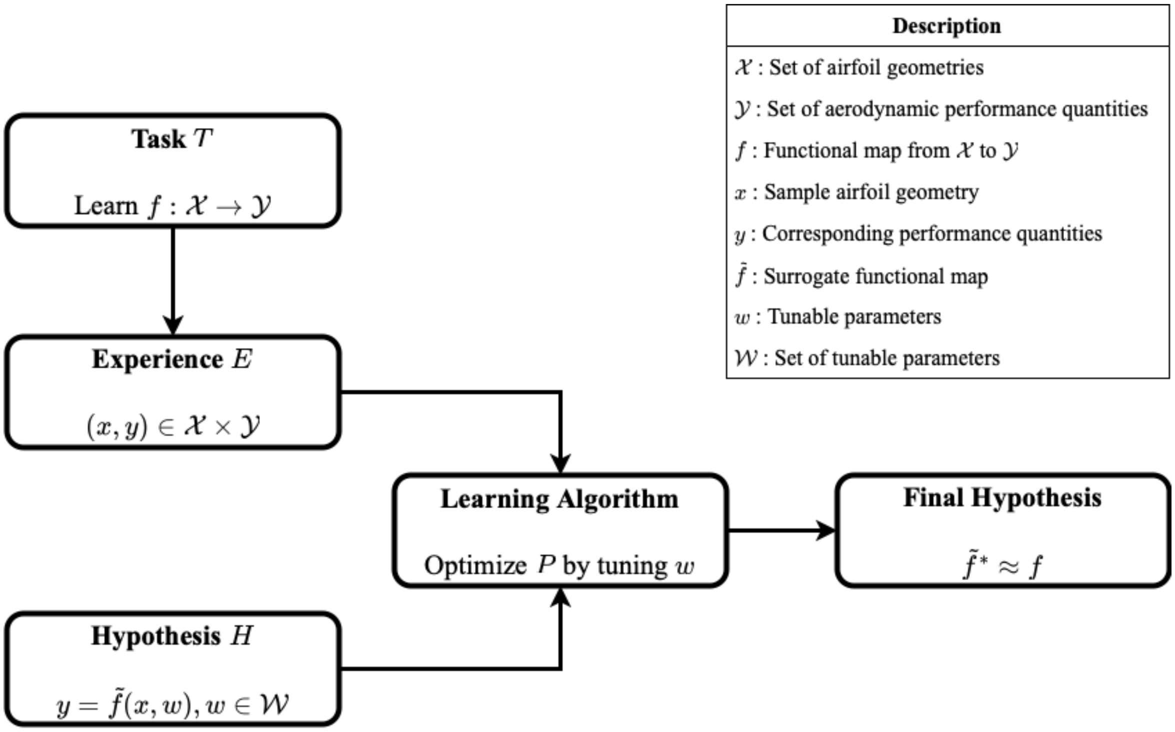

Figure 2.1: The framework of the learning process for the development of the surrogate model

The experience block E deals with obtaining the appropriate instances of airfoil geometries (x) and their corresponding performance quantities (y) that collectively represent the space of wind turbine airfoils $(\mathcal{X})$ and the aerodynamic performance quantities $(\mathcal{Y})$. This involves efficient design of experiments (DoE) techniques and true aerodynamic data generation using high-fidelity methods like CFD.

The hypothesis block H involves design of suitable learning models or architectures $(\tilde{f})$ that contain an approximation of the true functional map between $\mathcal{X}$ and $\mathcal{Y}$. This is usually tuned by a set of hyperparameters that define the learning model or architecture.

The learning algorithm uses the data from E and obtains an optimal function that satisfies H, where the optimality is defined by a performance measure P defined on E that quantifies the accuracy of the surrogate model.

Mathematically, the entire learning framework can be represented using Equation 2.1, where f is the true functional relation between the airfoil geometries and their corresponding aerodynamic performance quantities and $\tilde{f}^*$ is the learnt surrogate model that approximates f with optimal performance measure.

$$\tilde{f}^* = \underset{\tilde{f} \in H}{\operatorname{argmin}} \left\{ P_E \left(y - \tilde{f}(x) \right) \right\} \tag{2.1}$$

2.2: Generation of a Wind-Turbine Airfoil Database

This section gives a detailed account of the steps involved in generating a representative database of wind turbine airfoil geometries and their corresponding aerodynamic performance quantities. Firstly, each airfoil geometry is defined using a set of geometry parameters and then a design of experiments is performed on this parametric space of airfoils. CFD is performed on these airfoil geometries to arrive at the corresponding aerodynamic performance quantities.

2.2.1: Airfoil Geometry Parametrization

Airfoil parametrization is required to reduce the number of degrees of freedom to represent the geometries while simultaneously representing the range of airfoil geometries that define the cross sections of wind turbine blades. This becomes important to reduce the impact of "curse of dimensionality" for increasing number of degrees of freedom.

Early methods of airfoil geometry parametrization involved representing the geometries by leading edge radius, maximum camber and maximum thickness and their corresponding chord-wise locations. While these parameters represent a wide range of subsonic airfoil geometries, they restrict the camber and thickness profiles to very few degrees of freedom. Additional notations mentioning different characteristics such as reflex camber are introduced.

Other class of parametrization methods use control points to represent airfoil surface using blending functions such as B-spline, Bezier curves. These methods pose difficulty in obtaining and modifying the control points as they can be out of the airfoil surface. Hicks and Henne [48] used local analytic functionals to represent deformations and shape variations from baseline airfoils. While these functionals are extremely powerful in representing the airfoils accurately, they are localized to a baseline geometry and cannot be used for a global parametrization of the geometric space. PARSEC method by Sobieczky [49] uses 11 variables such as leading edge radius, upper and lower crest curvatures and their locations, trailing edge thickness, direction angle and offset. This method provides a refinement of the early variables of importance, but however doesn't accurately represent complex distributions of thickness and camber.

Class function - Shape function Transformation (CST) by Kulfan [50] is a parametrization method to represent a wide variety of aerospace geometries ranging from airfoils to fuselages.

Class function determines the class of geometries for which the parametrization is used. When applied to wind turbine airfoils, both the top and bottom surface of the airfoil are expressed as a combination of class function and shape function. The mathematical representation of the top and bottom surface of the airfoil shape is given in terms of the normalized surface coordinate (ζ) and chord-wise coordinate (ψ) in Equations 2.2 - 2.5.

$$\zeta(\psi) = C_{N_2}^{N_1}(\psi)S(\psi) + \psi\zeta_T \qquad \text{(Class} - \text{Shape function)} \tag{2.2}$$

$$C_{N_2}^{N_1}(\psi) = \psi^{N_1}(1-\psi)^{N_2} \qquad \text{(Class function)} \tag{2.3}$$

$$S(\psi) = \sum_{k=0}^{n} A_k S_k \qquad \text{(Shape function)} \tag{2.4}$$

$$S_k(\psi) = {}^{n}C_k \psi^k (1-\psi)^{n-k} \qquad \text{(Bernstein polynomials of order } n) \tag{2.5}$$

The class function is frozen for all airfoils and represents the characteristic of rounded leading edge and sharp trailing edge by using $N_1 = 0.5$ and $N_2 = 1$. An additional term to account for finite trailing edge thickness is introduced using ζ_T. The shape function is allowed to be from the space of 8th degree polynomials, to allow for larger gradients of the thick wind turbine airfoils.

In the original CST method, the shape function is further represented as a combination of Bernstein polynomials in Equation 2.5, where the coefficients of these shapes are the design variables. This parametrization becomes ill-conditioned with higher order of polynomials, i.e., two largely different airfoils in the parametric space approximately give the same airfoil in physical space. To address this problem, throughout this study, the basis polynomials are replaced with Chebyshev polynomials (Equation 2.6), with an affine-transformation of the domain to $[0, 1]$, to allow for well-behaved parametrization of the geometries (Refer Appendix A).

$$S_k(\psi) = \cos(k \arccos(2\psi - 1)) \qquad \text{(Chebyshev polynomial of order } k\text{)} \tag{2.6}$$

The total number of design variables in this parametrization is 20, viz., 9 coefficients of Chebyshev polynomials each for the upper and lower surfaces, and 1 variable each to define the trailing edge parameter for upper and lower surfaces. Figure 2.2 shows the first 5 class-shape functions for the airfoils and the accuracy of the parametrization compared to the true airfoil surface for DU93-W-210 airfoil using 8^{th} order polynomial shape functions.

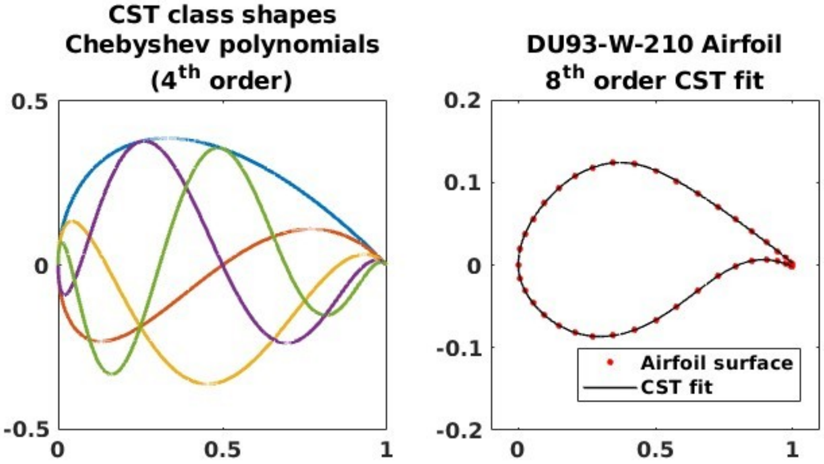

Figure 2.2: Airfoil representation using CST with Chebyshev polynomials. Left: First 5 class-shape functions, Right: Parametrization of DU93-W-210 airfoil using 8^{th} order polynomial shape function

2.2.2: Design Space Exploration

To generate a representative database of wind turbine airfoils, various existing wind turbine designs are surveyed and airfoil geometries defining the NREL-5MW [51], DTU-10MW [52], and IEA-15MW [53] wind turbines are chosen. Figure 2.3 shows the variation of local relative thickness (thickness-to-chord ratio) of the airfoil cross-sections along the blade span for the three reference turbines.

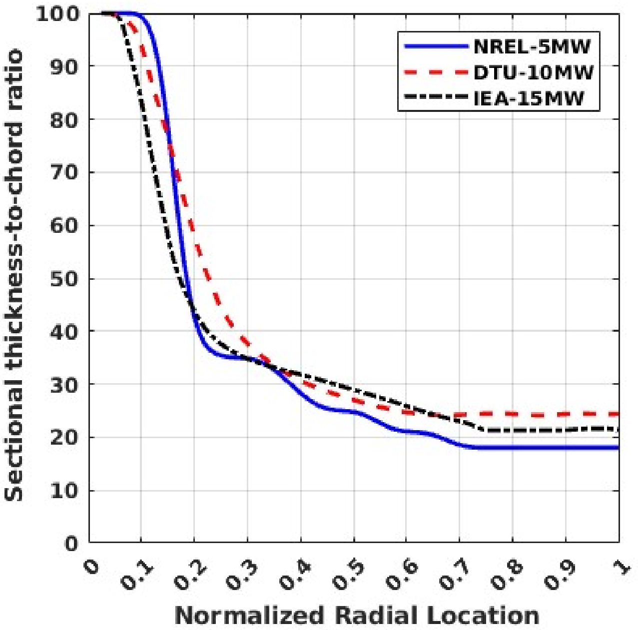

Figure 2.3: Variation of airfoil cross-section thickness-to-chord ratio for the reference turbines

2.2.2.1: DU and NACA Airfoils

The NREL-5MW reference turbine blade is defined using airfoils from DU series and NACA series. 5 different airfoils defining the turbine blade, ranging from 17% thickness to 35% thickness are chosen as a representative baseline set of wind turbine airfoils. Figure 2.4 shows the 5 baseline airfoils profiles from the NREL-5MW blade, and their thicknesses are given in Table 2.1.

Table 2.1: List of airfoils from NREL-5MW blade

Airfoil	t/c	Notes
DU99-W-350	35%	DU35_A17
DU97-W-300	30%	DU30_A17
DU91-W2-250	25%	DU25_A17
DU93-W-210	21%	DU21_A17
NACA64-618	18%	NACA64_A17

2.2.2.2: FFA Airfoils

The DTU-10MW and IEA-15MW reference turbine blades are defined using airfoils from FFA series. 4 different airfoils defining the turbine blade, ranging from 21% thickness to 36% thickness are chosen as a representative baseline set of wind turbine airfoils. Along with these, 2 cross-sections in the intermediate thickness regions are extracted and shown in Table 2.2 as blended airfoils. Additionally, the inboard airfoil profile with Gurney flap is also added to the list. Figure 2.5 shows the 7 baseline airfoils profiles chosen from the family of FFA airfoils, and their thicknesses are given in Table 2.2.

Table 2.2: List of airfoils from DTU-10MW and IEA-15MW blade

Airfoil	t/c	Notes
FFA-W3-360	36%	
FFA-W3-301	30%	
FFA-W3-241	24%	
FFA-W3-211	21%	
FFA-W3-360-GF	36%	FFA-W3-360 airfoil with Gurney flap
FFA-W3-330	33%	Blended airfoil
FFA-W3-270	27%	Blended airfoil

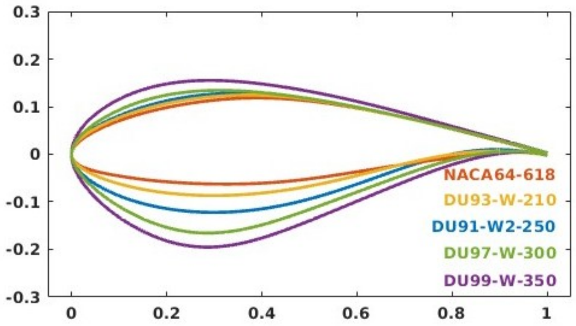

Figure 2.4: 5 baseline airfoil profiles from NREL-5MW turbine

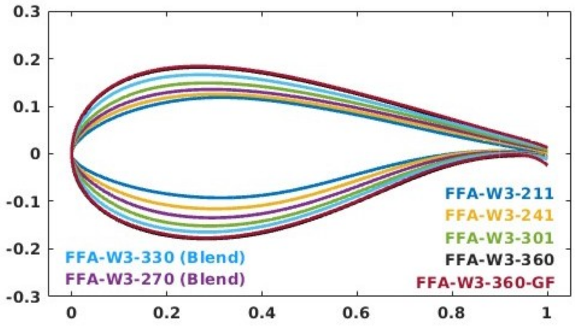

Figure 2.5: 7 baseline airfoil profiles from DTU-10MW and IEA-15MW turbines

2.2.2.3: Design of Experiments

To explore the design space of airfoils, new airfoil geometries are generated by perturbing the design parameters of each baseline airfoil upto 20% of their magnitudes. Latin hypercube sampling is used to appropriately choose a subset of 80 airfoil geometries around each baseline airfoil, by also maximising the distance among the perturbations around all 12 airfoils. This was done to ensure exploring a larger space of wind turbine airfoils, generating 972 unique airfoil shapes in the database. Figure 2.6 shows the magnitudes of the pairwise sample correlation coefficient matrix for the design variables for the baseline set of airfoils and the entire database of 972 airfoils. While the correlations of the variables corresponding to the upper surface are high for the entire database, this is a result of the correlated variables defining the geometries of the baseline airfoils, hence making a representative database of wind turbine airfoils.

Figure 2.7 shows the perturbation geometries around FFA-W3-241 airfoil and Figure 2.8 shows the range of the airfoil geometries used in this study. This database covers a large range of airfoils spanning from 15% thick airfoils to 40% thick airfoils.

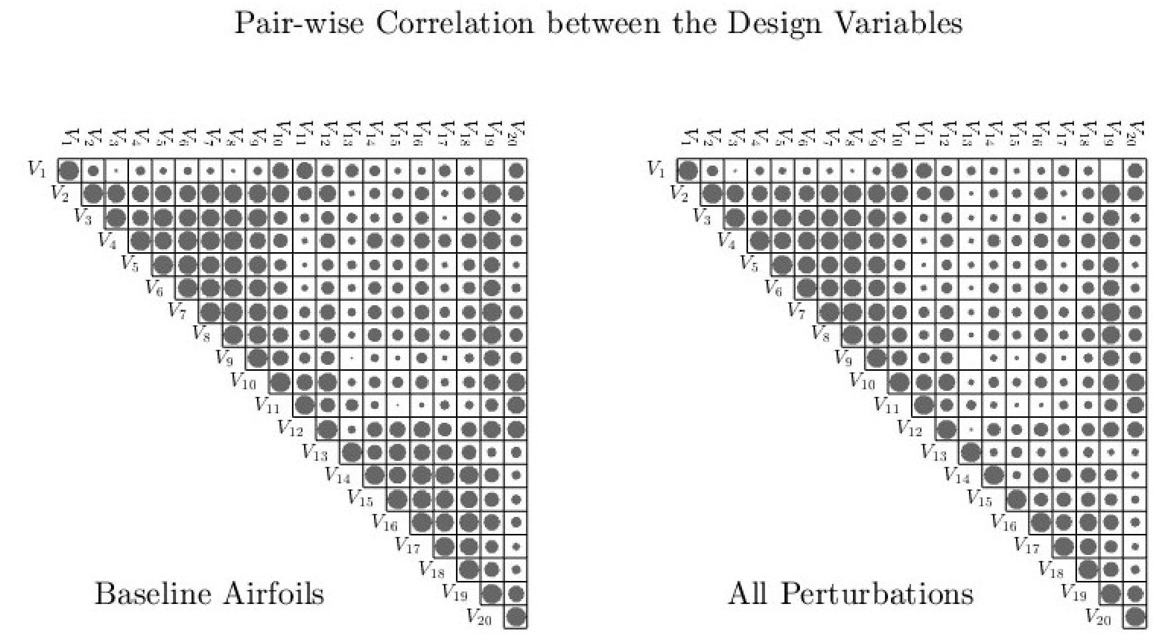

Figure 2.6: Correlation coefficient between design variables in the airfoil database - size of the marker represents the magnitude

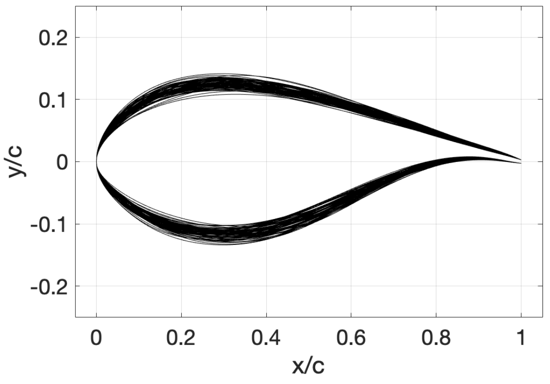

Figure 2.7: 80 perturbation geometries around FFA-W3-241 airfoil

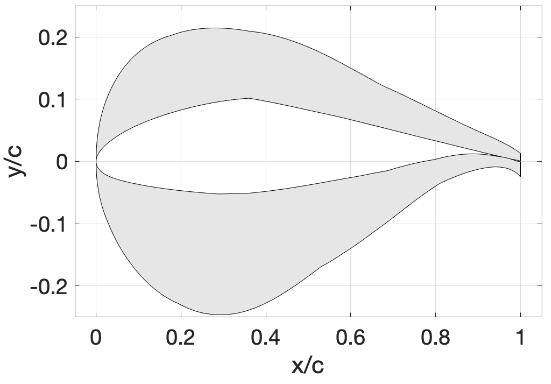

Figure 2.8: Range of perturbation geometries around all baseline airfoils

2.2.3: Generation of Performance Data

The quality of a surrogate model as an analysis tool is directly dependent on the quality of its training data. Thus, the true performance quantities of each of the airfoils in the database are estimated using high-fidelity methods like CFD.

2.2.3.1: Test Matrix of Freestream Parameters

Figure 2.9 shows the variation of the local chord based Reynolds number and angle of attack along the blade span for the reference wind turbines. The effective angle of attack profiles are obtained using blade element momentum theory computations for wind turbine blades, with axial and tangential flow induction, and Glauret correction [54].

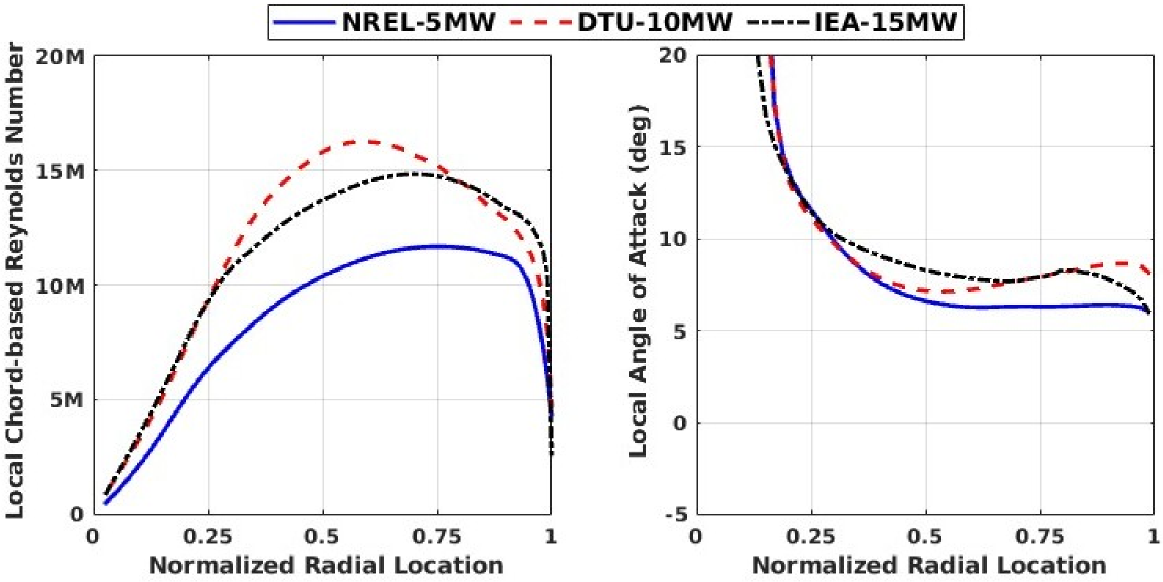

Figure 2.9: Distribution of local chord based Reynolds number (left) and local angle of attack (right) along the blade span for the reference wind turbines

Based on the reference values and the general operational range of wind turbines, the relevant range of Reynolds numbers (Re) and angles of attack (α) for generating the training data are chosen to be $\mathrm{Re} \in \{3\mathrm{M}, 6\mathrm{M}, 9\mathrm{M}, 12\mathrm{M}\}$ and $-4^\circ \leq \alpha \leq 20^\circ$.

2.2.3.2: Description of the CFD Solver

To obtain the performance profiles of the database of airfoils, an in-house solver HAM2D is used to perform CFD simulations. This solver is built on a Hamiltonian path based flow solution methodology [55] for fully quadrilateral unstructured grids.

Governing Equations

Navier-Stokes Equations

The governing conservation laws of mass, momentum and energy for a continuum fluid are collectively known as the Navier-Stokes equations. These equations describe the general physical behavior of fluids. In their conservation form, these equations are shown in Equations 2.7-2.9, where the variables $\rho, \rho\vec{u}, \rho E$ correspond to the mass, momentum and total energy of the fluid in a unit volume respectively, p, T correspond to the static pressure and static temperature, τ is the viscous stress vector.

$$\frac{\partial}{\partial t}(\rho) \quad +\nabla \cdot (\rho\vec{u}) \quad = 0 \qquad \text{(Mass Conservation)} \tag{2.7}$$

$$\frac{\partial}{\partial t}(\rho\vec{u}) \quad +\nabla \cdot (\rho\vec{u} \otimes \vec{u}) \quad = -\nabla p + \nabla \cdot \tau \qquad \text{(Momentum Conservation)} \tag{2.8}$$

$$\frac{\partial}{\partial t}(\rho E) \quad +\nabla \cdot ((\rho E + p)\vec{u}) \quad = k\nabla^2 T + \nabla \cdot (\tau \cdot \vec{u}) \qquad \text{(Energy Conservation)} \tag{2.9}$$

The viscous stress is formulated based on Stokes' hypothesis and shown in Einstein's indicial notation in Equation 2.10, with the molecular viscosity of the fluid as μ.

$$\tau_{ij} = \mu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right] \tag{2.10}$$

No general analytical solution is found for these equations, and numerical methods are in practice to understand the physical behavior of fluids.

Reynolds-Averaged Navier-Stokes (RANS) Equations

Numerical simulation of the time dependent Navier-Stokes equations is called as Direct Numerical Simulation (DNS), and requires exorbitant computational power to resolve the physics of turbulent flow across various length and time scales. The Reynolds-Averaged Navier-Stokes (RANS) equations offer a practical reduction of the computational effort, by resolving only statistically steady flow features. In this approach, flow variables are decomposed into mean and fluctuation components, and the flow equations describing the time averaged mean flow quantities are obtained. Most of the fluctuation terms get nullified due to the time averaging. However, the cross correlation terms of the fluctuation quantities do not average to zero. These terms contain the information of the statistics of flow behavior in the unresolved scales, and act in the momentum and energy equations. The additional term in the momentum equation is called the Reynold's stress tensor, given by Equation 2.11, with $\bar{f}$ describing the mean quantity and f' describing the fluctuation of f.

$$(\bar{\tau_{ij}})_{\text{Turb}} = -\rho \langle u_i' u_j' \rangle \tag{2.11}$$

The additional terms involved in the energy equation are ignored for this study, as all scenarios considered will not have any heat transfer in them.

Spalart-Allmaras (SA) Turbulence Closure Model

While the RANS equations provide a computationally cheaper description of the fluid behavior, additional assumptions are required to compute the extra terms (Reynolds stress). This is called the turbulence closure problem. Boussinesq hypothesized that the momentum transfer caused by the turbulent eddies can be modeled with eddy viscosity (μ_t), which forms the basis for various turbulence closure models, and is given in Equation 2.12, with k corresponding to the turbulent kinetic energy.

$$(\bar{\tau_{ij}})_{\text{Turb}} = 2\mu_t \left(\bar{S_{ij}} - \frac{1}{3}\frac{\partial \bar{u}_k}{\partial x_k}\delta_{ij} \right) - \frac{2}{3}\bar{\rho} k \delta_{ij} \tag{2.12}$$

The closure model for eddy viscosity is given in the form of a one equation turbulence model by Spalart and Allmaras [33]. The model describes the transport of eddy viscosity through a working variable $\tilde{\nu}$ as in Equation 2.13. The model has several empirical constants, and is calibrated with wall-bounded flows and maintaining consistency with the generalized law of the wall. The terms P, D, E correspond to the production, destruction and dissipation of the eddy viscosity variable.

$$\frac{D\tilde{\nu}}{Dt} = P(\tilde{\nu}, U) - D(\tilde{\nu}, U) + E(\tilde{\nu}, U) \tag{2.13}$$

Numerical Algorithm

The HAM2D flow solver uses a fifth-order WENO scheme for spatial reconstruction, with Roe's Riemann solver for inviscid flux, second-order central differencing for viscous flux, and GMRES with preconditioning for implicit time integration. HAM2D is extensively tested and validated with various wind turbine airfoil data available in the literature [56].

The computational grid generation for each airfoil is automated [57] to generate a pure quadrilateral mesh system, with 400 points on the airfoil surface in the wrap-around direction and an initial wall-normal spacing of $y^+ = 1$. The outer boundary is 300 chord lengths away from the airfoil with a far-field boundary condition. Figure 2.10 shows the grid around a DU21 airfoil generated using the automatic meshing framework.

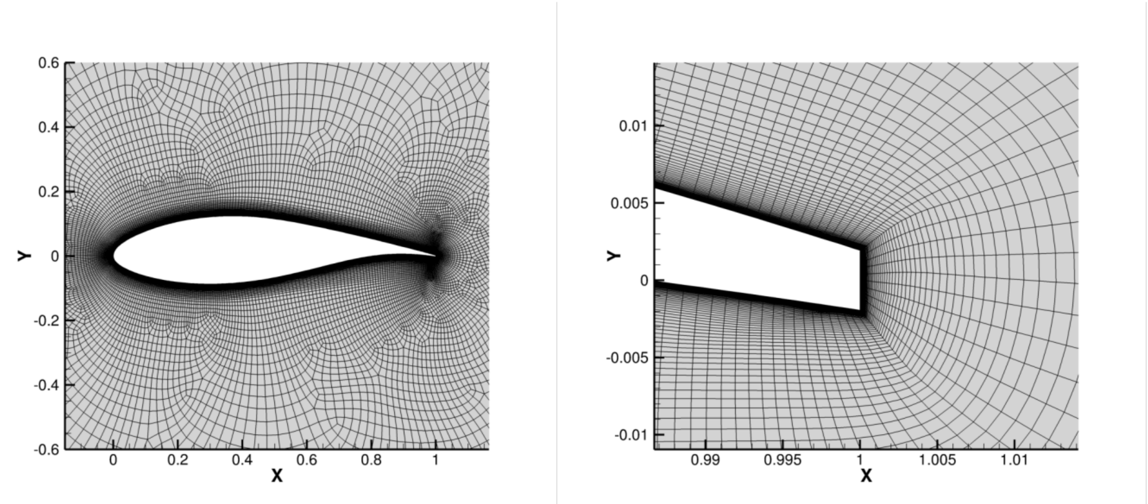

Figure 2.10: Spatial discretization for the field around DU21 airfoil: zoomed view near the surface (left) and zoomed view at the trailing edge (right)

2.2.3.3: Database of Performance Quantities

The solver is used to simulate the flow around the airfoil geometries for four different Reynolds numbers viz., $\{3, 6, 9, 12\} \times 10^6$ and 25 different angles of attack, equally spaced between $-4°$ and $20°$. A total of 3888 airfoil performance profiles formed the database of training and validation data for the surrogate model.

Figure 2.11 shows the range of the lift, pitching moment coefficients and lift-to-drag ratio of all the airfoils in the database. Both the geometries and the performance parameters span a large range, indicating the representativeness of the database for the purpose of surrogate modeling.

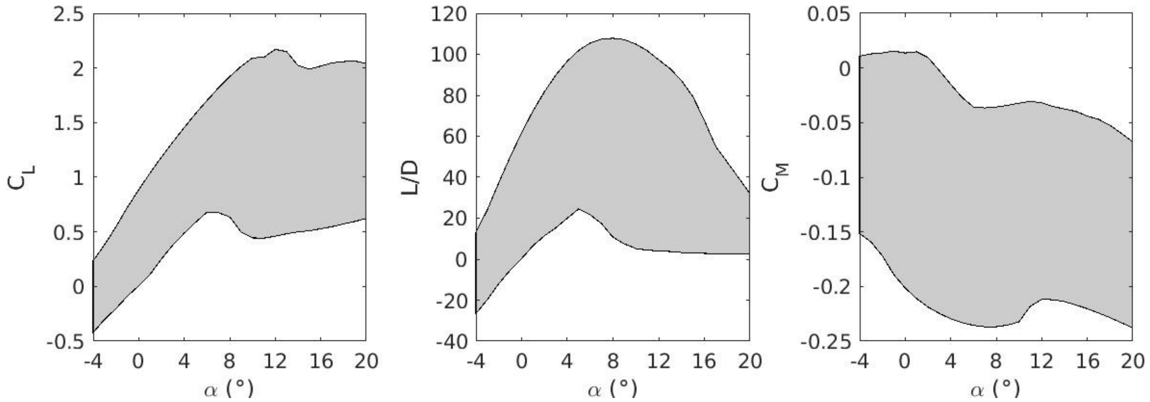

Figure 2.11: Range of the data from the CFD solutions - left: lift coefficient, center: lift-to-drag ratio, right: pitching moment coefficient

2.3: Training the Surrogate Model

This section describes different aspects of training the surrogate model using the geometry and aerodynamic quantities in the database. Firstly, the machine learning model is described, followed by the training method and hyperparameter tuning.

2.3.1: Machine Learning Model and Architecture

The main idea of a surrogate model is to predict the airfoil performance characteristics for a given geometry, without having to perform a computationally expensive CFD calculation. Artificial neural networks (ANNs) are a class of machine learning architectures that have robust non-linear function approximation capabilities. While other architectures such as SVM, Gaussian processes etc. also have strong modeling capabilities, the scope of this dissertation does not investigate the effect of these models on the wind turbine data.

Figure 2.12 shows the schematic of the surrogate model with the artificial neural network. Here, the surrogate model must predict entire performance quantity distribution (at multiple angles of attack). The choice of the architecture is made to ensure that the inputs and outputs correspond to the parametrization and aerodynamic polars respectively.

The sequence of computations can be broken down into three broad steps as shown in Figure 2.12 and represented mathematically in Equations 2.14 - 2.16 respectively. Marepally et al. [58] demonstrate an improved accuracy of the surrogate models with very simple architectures by using orthogonal input features using Proper Orthogonal Decomposition (POD). The first step in the surrogate model is the input processing step to ensure the representation in an orthogonal basis. Similar accuracy is achived using Chebyshev polynomials, indicating and reaffirming the need for well-conditioned parametrization of the geometries.

The second step contains the neural network computations, where the outputs are calculated through a sequence of computations propagating across hidden layers of neurons. Nonlinearity is introduced in the model through the use of activation functions at each layer. For the surrogate model studied in this work, the architecture consists of two hidden layers with hyperbolic tangent

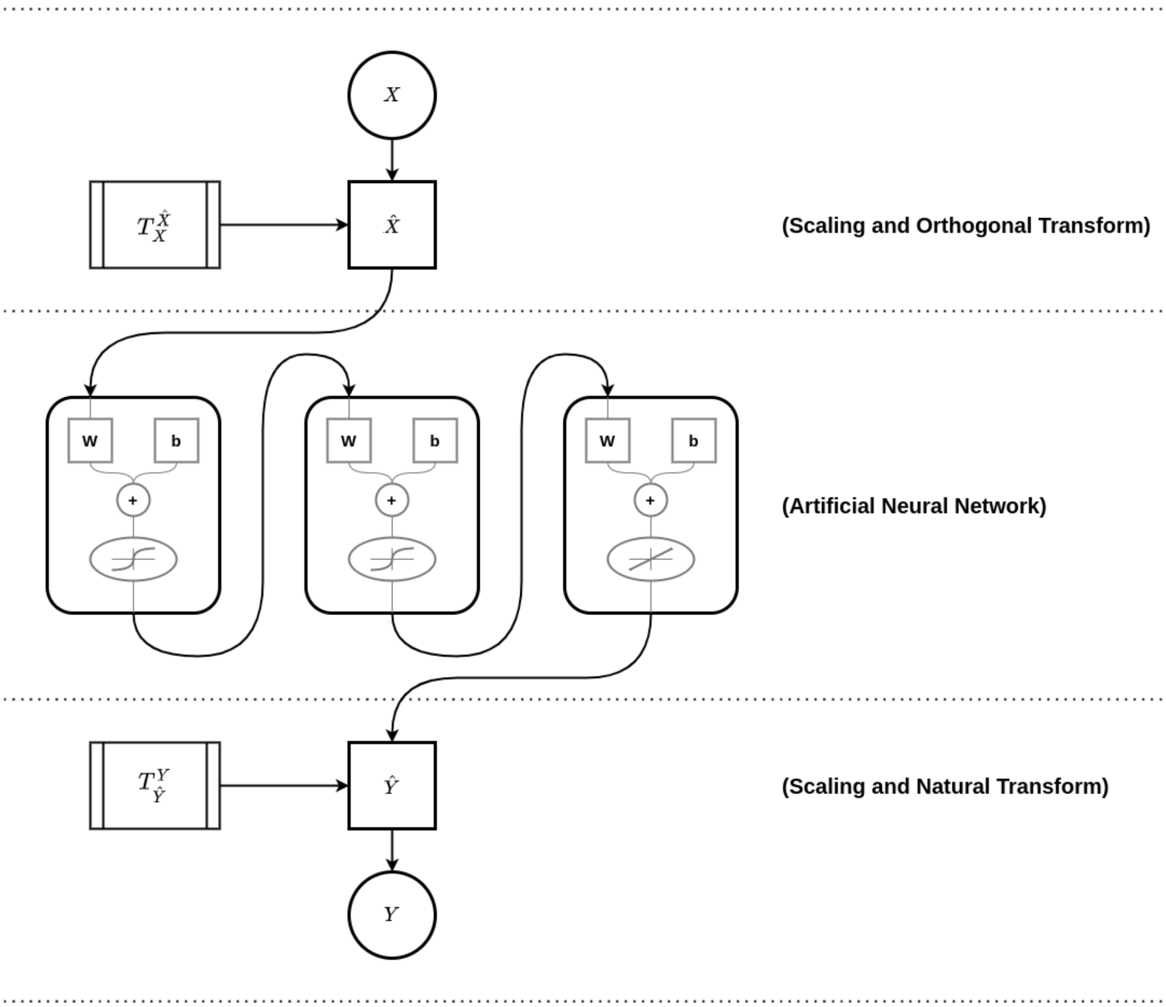

Figure 2.12: Schematic of surrogate model describing the sequence of computations and data flow

activation function in each layer. The outputs from this step are scaled and transformed values of the performance parameters. The surrogate model is adjusted by changing the free parameters w and b which represent the neural network weights and bias values.

The final step transforms the outputs of the neural network into aerodynamics performance polars. This computation is similar to the inverse of the first step, but performed on the output quantities, i.e., the output performance polars can be represented as a combination of certain general mode functions and the output of the neural network in the second step is the coefficient vector of the airfoil performance represented in this transformed space of mode functions. The added advantage of using such a transformation is that the modes can be chosen as continuous functions and the performance polar information is now available at any angle of attack in the range. In this study, the transformation function and mode shapes for the output performance polars are obtained using POD of the output polars of the baseline airfoil set.

$$\hat{X} = T_X^{\hat{X}}(X) \qquad \text{(Scaling and Orthogonal transform)} \tag{2.14}$$

$$\hat{Y} = g(\hat{X}, w, b) \qquad \text{(Neural network computations)} \tag{2.15}$$

$$Y = T_{\hat{Y}}^{Y}(\hat{Y}) \qquad \text{(Scaling and Natural transform)} \tag{2.16}$$

2.3.2: Training Algorithm

2.3.2.1: Performance Measure (Training Objective)

The surrogate model is developed to predict the aerodynamic performance quantities of any new wind turbine airfoil geometry. Hence, the measure of performance should quantify the accuracy in predictive modeling. Standard machine learning practice is to hold-out some data for validation and the performance is measured on the hold-out data.

The accuracy can be measured using normalized square of the L_2 norm of the error (mean square error - MSE) between true data and the predicted aerodynamic quantities for the validation set of geometries. However, this gives large contribution to outliers and the model tries to compensate by lowering the errors due to outliers with better performing points. Another measure is the normalized L_1 norm of the error (mean absolute error - MAE). While this relatively eases on the outliers, the measure is non-differentiable and could cause delayed convergence. A combination of these two measures is the Huber loss defined by Equation 2.17, where the error behaves as MSE at low values and as MAE at high values. A comparison of the Huber loss function with MSE and MAE can be seen in Figure 2.13.

$$L_\delta(\epsilon) = \begin{cases} \frac{1}{2}\epsilon^2 & |\epsilon| \le \delta \\ \delta\left(|\epsilon| - \frac{1}{2}\delta\right) & |\epsilon| > \delta \end{cases} \tag{2.17}$$

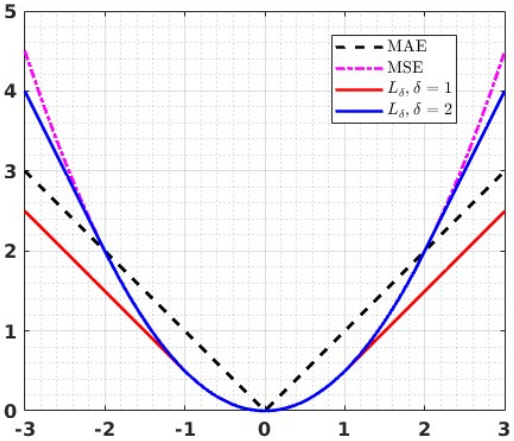

Figure 2.13: Comparision of Huber loss function with MSE and MAE

2.3.2.2: Training

Gradient based optimization techniques are used to optimize the free parameters (weights and bias values) of the neural network. The gradient of the performance measure is computed using backpropagation algorithm, gradient descent (Equation 2.18 is used to update the weights and bias of the neural network.

$$w_j^{(i+1)} = w_j^{(i+1)} - \lambda \nabla_j \langle P(X, Y) \rangle \tag{2.18}$$

Here, $P(X, Y)$ is a representative performance measure of the surrogate model. P can be computed on the entire dataset or representative subsets of the dataset to reduce the cost of computation. In stochastic gradient descent, for each iteration, the gradient is computed on the performance measure on a randomly chosen data point from (X, Y) and is used as an estimate of the true gradient in Equation 2.18. A good compromise between the cost and accuracy of the gradient is to perform mini-batch stochastic gradient descent, where the gradient is computed on a randomly chosen subset of (X, Y).

2.3.2.3: Cross-Validation

In order to test the predictive modeling of the surrogate model, i.e., testing the model performance on unseen data, a standard practice is to hold out parts of the dataset from training and test on the held out data. This process is called cross-validation.

Different variants of cross-validation are used for standard machine learning problems. A basic variant of cross-validation is to split the data into training (X_*, Y_*) and validation (X_{**}, Y_{**})

subsets, and validate the model performance on X_{**}. For this variant, the performance measure becomes a representative estimate only if the size of the datasets is very large.

Another variant of cross validation is called k-fold cross-validation. Here, data is divided into k partitions (folds) after randomly shuffling the dataset. Performance is measured by making each group as validation subset and the rest of the groups as training subset. The validation error estimate is given as the mean of the individual estimates. Algorithm 1 shows the sequence of steps involved in k-fold cross validation. The inner loop corresponds to the test-train split cross validation. A standard practice in the machine learning community is to use $k = 5$ or 10.

Algorithm 1 k-fold Cross-Validation Algorithm

```
procedure k-FOLD-CV($(X, Y)$)    ▷ Model training and cross validation with datasets $X, Y$
    Partition (randomly) the datasets $(X, Y)$ into $k$-folds, $(X, Y) = \bigcup_{i=1}^{k}(X_i : Y_i)$
    for $i = 1, 2, \ldots, k$ do
        $(X_{**}, Y_{**}) = (X_i, Y_i)$    ▷ Validation subset
        $(X_*, Y_*) = (X - X_i, Y - Y_i)$    ▷ Training subset
        Train the model on $(X_*, Y_*)$
        Compute the performance $P_i$ on $(X_{**}, Y_{**})$    ▷ $P_i$ is an error norm
    end for
    Performance of the model $P = \frac{1}{k}\sum_{i=1}^{k} P_i$
end procedure
```

An extension of k-fold cross validation is a repeated k-fold variant, where the k-fold cross validation is repeated on the dataset by reshuffling it. The performance measure in this variant is the average of the k-fold performances on the repititions. An illustration showing the comparison of the validation data in these three approaches can be seen in Figure 2.14.

A major benefit of performing cross-validation is that it ensures good predictive modeling and no overfitting of the data.

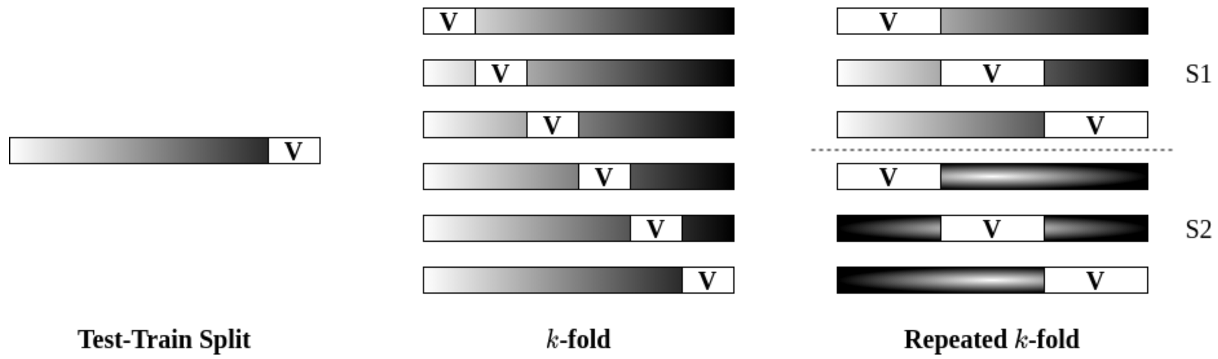

Figure 2.14: Illustration of different types of cross validation - Left: Test-train split, Center: k-fold cross validation (with $k = 6$), Right: Repeated k-fold cross validation with $k = 3$ and 2 repititions

2.3.2.4: Regularization

Another technique to ensure no overfitting of the data is to introduce a penalty parameter into the training loss function to restrict the usage of large number of free parameters. This penalty is introduced by adding the norm of the free parameters into the loss function as shown in Equation 2.19.

$$L_{\text{train}} = P(X, Y) + \lambda \|w\|_* \tag{2.19}$$

If the choice of the penalty norm is L_2, then the regularization is called Ridge regression. This term tries to move the free parameters close to zero. Another choice is to use L_1 norm, called Lasso regression. Here the inactive free parameters are dropped off due to the nature of L_1 norm.

2.3.2.5: Hyperparameter Tuning

While the training and cross-validation achieve optimal performance measure by tuning the free parameters of a model, the base architecture of the model often saturates the extent to which the performance can be optimized. The model architecture is defined by a set of hyperparameters. For instance, a neural network architecture is governed by the number of layers and number of neurons in each layer. These hyperparameters are not altered during the training process, and hence additional considerations need to be made to fix on the hyperparameters. Hyperparameter tuning is associated with the modification of these parameters to change the architecture of the surrogate model so that an optimal performance can be achieved.

Algorithm 2 shows the methodology of tuning the model architecture nested with cross-validation.

Algorithm 2 Hyperparameter Tuning Algorithm

```
procedure HYPERPARAMETER-TUNING                                   ▷ Tuning the architecture
    Choose the possible domain of the hyperparameters ($\lambda$) that define the model architecture
    for each $\lambda$ do
        Perform $k$-fold-CV$(X, Y)$ to compute $P_\lambda$
    end for
    Best architecture corresponds to $\lambda$ for which $P_\lambda$ is the best.
end procedure
```

Model tuning can be performed by computing the performance exhaustively on the entire possible set of hyperparameters, or using random search and bayesian methods. In the context of the current work, exhaustive grid search is performed as the number of hyperparameters is limited to 2 viz., number of neurons in each hidden layer.

2.4: Performance of the Surrogate Model

2.4.1: Hyperparameter Tuning Results

The surrogate model is trained using mini-batch stochastic gradient descent with Huber loss function as a performance measure on the scaled output quantities. Lasso regression is used, and repeated k-fold cross validation is applied for this study with $k = 5$ and 3 repititions. Hyperparameters for the surrogate model are the number of neurons in the two hidden layers of the neural network.

The cross-validation error for different set of hyperparameters can be seen in Figure 2.15, with $H1$ and $H2$ being the number of neurons in the first and second hidden layer respectively. Observe that the cross-validation error is significantly high for less neurons in the first layer, and is nearly constant with the second layer after a minimum. As the number of neurons increases, the error drops down, showing best performance around $H1 = 32$ and $H2 = 16$.

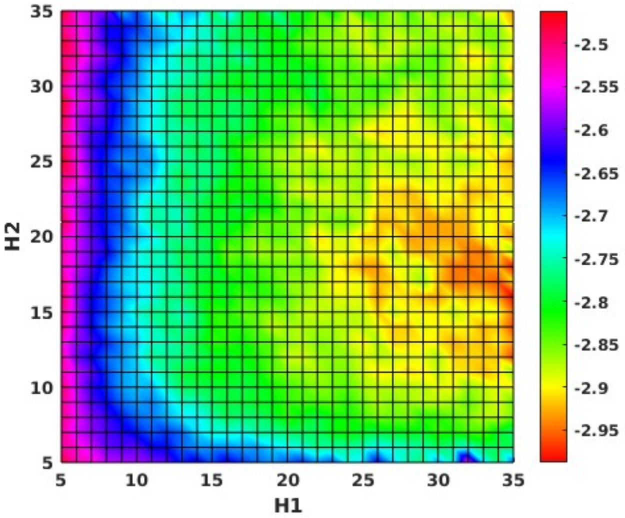

Figure 2.15: Cross-validation error for different hyperparameters (logarithmic scale)

Observe that the cross-validation error is average performance of the multiple repetitions of the validation dataset. In order to truly understand the variability of the model with respect to the change in training data, the variation of the error is also computed for the multiple folds and shown in Figure 2.16. While the cross-validation error corresponds to the Bias estimate of the model, this term estimates the variance. The variance estimate is observed to become higher with the number of free parameters, i.e., in both the directions of increasing $H1$ and $H2$.

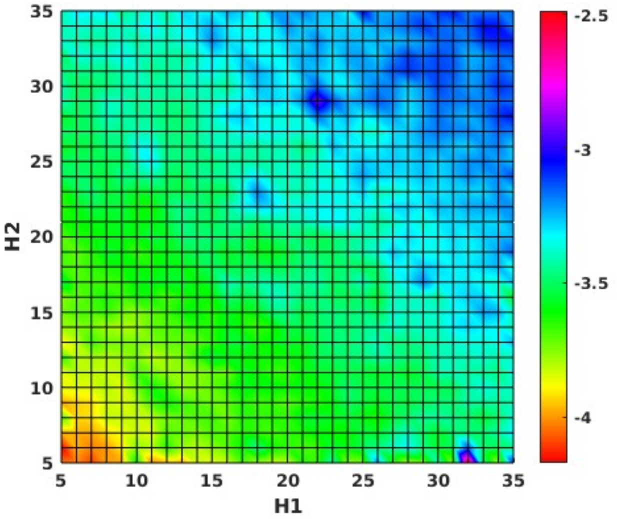

Figure 2.16: Variablilty of the error for different hyperparameters (logarithmic scale)

A composite performance metric is calculated as the sum of these two quantities, shown in Figure 2.17. An optimal choice of the network based on this composite score is $H1 = 25$ and $H2 = 20$.

While the number of layers of the neural network can also be varied, no improvement was observed in the cross-validation error with higher number of layers. Furthermore, as the network becomes deeper (more layers), the complexity increases and the variance estimates also increase. Hence, for this dataset, 2 hidden layers is the optimal choice of architecture.

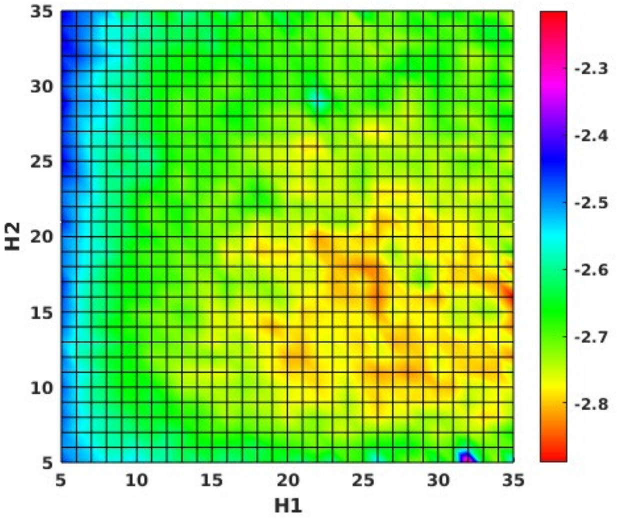

Figure 2.17: Composite performance statistic for different hyperparameters (logarithmic scale)

2.4.2: Cross-Validation Results

The comparison of the predicted quantities to the actual CFD predictions for the 5 folds of the data are shown in Figure 2.18. These results correspond to the optimal neural network architecture of 2 hidden layers $H1 = 25$ and $H2 = 20$. It can be seen that all the 5 folds show similar range of error compared to CFD.

2.4.3: Validity of the Surrogate model

Once the model is tuned using the composite performance statistic to yield the required level of optimality in hyperparameter space, the surrogate model is retrained on the entire dataset without any hold out data. This is the final surrogate model that is used to test the performance in predicting the airfoil aerodynamic characteristics.

Figure 2.19 shows the lift, lift-to-drag ratio and pitching moment coefficients for NACA64-618 (18% thickness), DU91-W2-250 (25% thickness), FFA-W3-301 (30% thickness) and FFA-

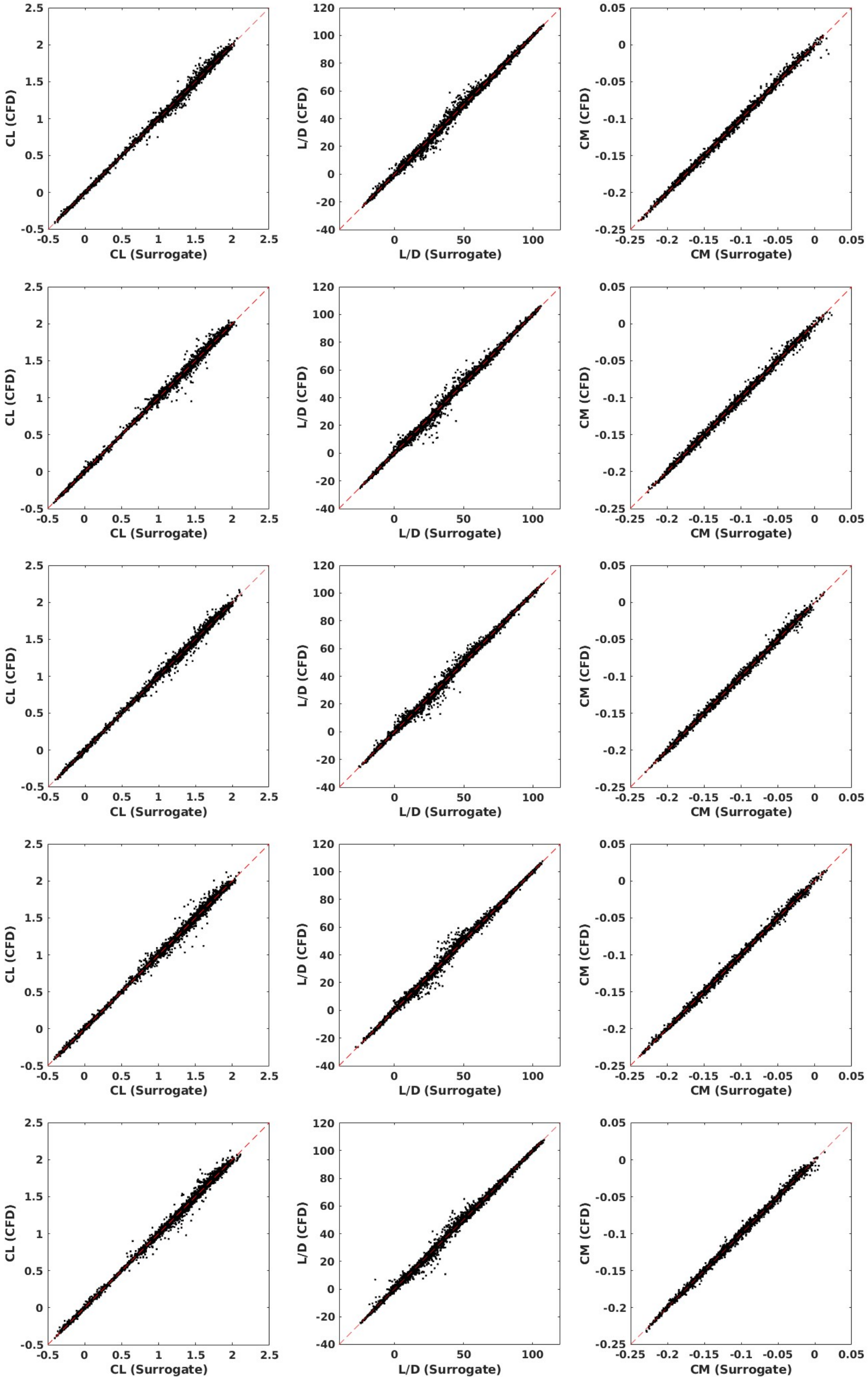

Figure 2.18: Performance of the surrogate model on $k = 5$-folds

W3-360 (36% thickness) airfoils. It can be seen that the predictions are uniformly accurate over the large range of thickness of wind turbine airfoils.

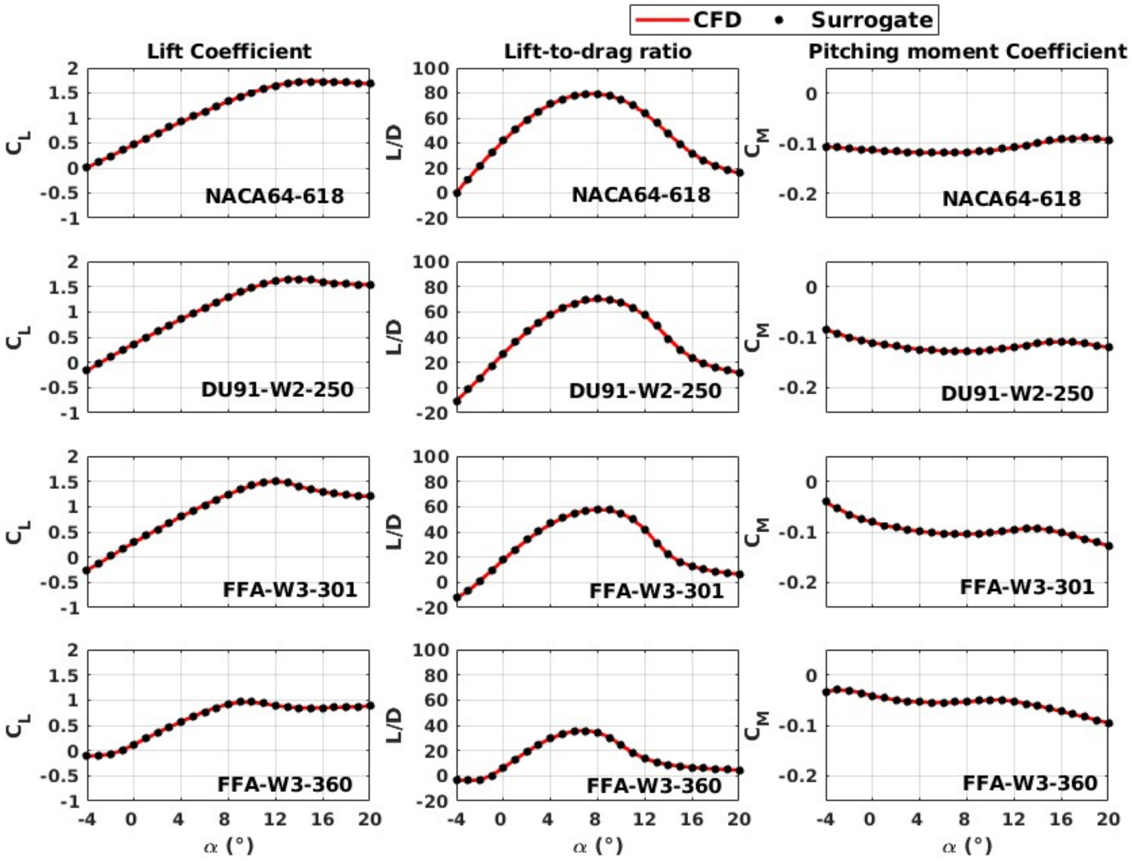

Figure 2.19: Performance of the surrogate model on representative airfoils in various thickness categories - columns (left to right): lift coefficient, lift-to-drag ratio, pitching moment coefficient respectively

Quantitative measures of the error associated with the surrogate model is calculated for all the airfoils in the database and shown in Table 2.3. The L_2 error norm describes the overall fit of the profile and L_∞ norm indicates the maximum error. The 99th percentile value of L_∞ norm is also indicated. This is a better measure of the performance, as it excludes the effect of outliers.

The observed L_2 and L_∞ (99th percentile) errors of the surrogate model shown in Table 2.3 correspond to less than 2% of the magnitudes of the corresponding lift coefficient, lift-to-drag ratio and pitching moment coefficient of a nominal airfoil in the database at the operational angle of attack.

Table 2.3: Quantitative error estimates of the surrogate predictions for all airfoil geometries

Error norm	c_l	L/D	c_m
L_2	3.4×10^{-2}	3.9×10^{-1}	2.8×10^{-3}
L_∞	1.8×10^{-1}	$1.2 \times 10^{+1}$	1.6×10^{-2}
L_∞ (99th percentile)	3.2×10^{-2}	2.0×10^{0}	2.1×10^{-3}

2.5: Summary

A surrogate model exclusively trained using wind turbine airfoil geometries is developed. Some key takeaways are

(1) Efficient design space exploration strategies are explored and a representative database of wind turbine airfoil geometries is generated.

(2) The generated database of airfoil geometries and performance curves is the first of its kind with around $100,000$ CFD simulations, and is a benchmark dataset of wind turbine airfoils.

(3) Very simple architectures of shallow neural networks could achieve high accuracy, with efficient processing of the data into orthogonal features.

(4) The surrogate model predictions are uniformly accurate for airfoils spanning over a large thickness range, with very low 99th percentile maximum error norms of 0.03, 2 and 0.002 in the lift coefficient, lift-to-drag ratio and pitching moment coefficient respectively when measured on the entire database. These values correspond to less than 2% of the respective magnitudes at the operational angle of attack of the airfoils.

Chapter 3: Data Puncturing

While the accuracy of the developed surrogate model to predict the aerodynamics of airfoil geometries is high, generation of the training data is the costliest step in its surrogate model development. This becomes a bottle-neck in most of the physical systems modeling due to the high costs of each simulation.

In addition to the geometric design variables for airfoils, the freestream conditions such as angle of attack and Reynolds number each in turn magnify the number of simulations required to generate the training data exponentially. Other freestream parameters such as the effect of turbulence intensity, Mach number etc become important to the designers based on the area of application. Owing to the curse-of-dimensionality, it is impractical to use either experiments or computational methods to generate the required amount of data at each of these freestream conditions for a large set of geometries.

3.1: Cost of Surrogate Model Development

The cost associated with the development of the surrogate model is given by Equation 3.1. This includes the cost of data generation and the cost of training the model.

$$T = N_{\text{train}} \times T_{\text{data}} + T_{\text{training}} \tag{3.1}$$

For engineering applications depending on numerical simulations, the cost of training is negligible in comparison to the cost of data generation. For the RANS based CFD of airfoils, the estimated time per CFD case at one angle of attack is 3 hours, while the time for training is 15 minutes. Since the generation of data is physical model based, the only free parameter to reduce the cost of surrogate model is N_{train}, i.e., required amount of training data. Thus, it becomes important to understand the effect of the amount of training data on the accuracy of the surrogate model.

3.2: Data Synergy and Redundancy

In a dataset, every data point carries certain inherent information. However, when multiple data points are studied together, various scenarios arise regarding the collective information obtainable with these data points. Data is said to be "synergistic" if the collective information of the data points is more than the sum of information carried by each data point separately, and "redundant" if the collective information is less than the sum of individual information.

The idea of synergy and redundancy can be explained with a simple example. Consider 3 data points to be the lift predictions of a particular airfoil at three different angles of attack. Independently, each data point has the information of lift coefficient at that particular angle of attack and nothing else. However, when two data points are studied together, apart from the lift coefficient at those two angles of attack, they also give information about the trend of the lift coefficient. If these are within attached flow angles, this extra information can be an estimate of the lift curve slope. Here the data is synergistic. Now, if the third data point also falls within the linear trend, no additional information is added. Moreover, this prediction can be achieved even without the third data point. This is the case of data redundancy, that is, at least one of the three points is not necessary. On the other hand, if the third point gives information about onset of stall or beyond stall characteristics, all the data is synergistic. However, it must be noted that even when the third data point is redundant, it adds confidence to the estimate of lift curve slope. Hence, synergy of a dataset is a measure of the information that the dataset explains while redundancy is a measure of the confidence of the inferences made on the dataset.

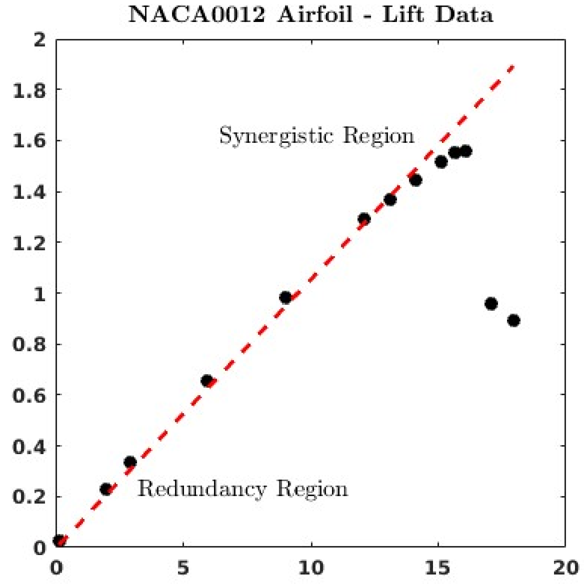

Figure 3.1: Illustration of synergy and redundancy - NACA0012 lift data

Figure 3.1 shows the regions of synergy and redundancy on sample NACA0012 airfoil data. It can be noted that the same best-fit line will be predicted even with more measurements at low angle of attack. Also, it must be noted that the redundancy and synergy using linear lift region explained here is just an illustration, and is based on the fact that airfoils exhibit linear lift region at low angles of attack. Hence, any known behavior of data adds more information at our disposal, resulting in redundancy of data points.

There is a huge potential for improvements in the efficiency of training data generation for the development of surrogate models, by carefully adjusting data synergy and redundancy to ensure a suitable balance between model cost and confidence.

3.3: Data Puncturing

“Data puncturing” is a new term introduced in this study, and is defined in the context of data points that require multiple measurements or evaluations. For example, if each data point is considered as an airfoil lift curve, to formally define the curve, lift measurement is required at multiple angles of attack within the range of interest. For such data points, punctured data is formally defined as a data point that is not complete, i.e., lift curve with measurements only at few angles of attack.

As explained in the previous section, knowledge of the aerodynamic behavior of airfoils can reduce the amount of data required to characterize and predict the airfoil aerodynamics. This becomes crucial in terms of reducing the cost of model generation, by reducing the associated data generation costs.

In the developed surrogate model, the aerodynamic performance polars of the airfoils are expressed in terms of their POD modes. Thus, an information of the general behavior of the airfoils is embedded into these mode functions, and thereby evaluation of these polars at all combinations of freestream parameters for all the geometries becomes redundant.

We test different styles of data reduction to see the impact on the accuracy of the surrogate model.

3.3.1: Naive Data Puncturing

Naive data puncturing strategy is the most basic reduction in the data. Here, data is removed at equal intervals for all the geometries. In the context of aerodynamic polars, the data computation can be visualized as not available at specific angles of attack for all the geometries.

Figure 3.2 shows the data availability plots of the punctured dataset for different levels of naive puncturing. Each row corresponds to a diferent geometry, and each column corresponds to a different angle of attack.

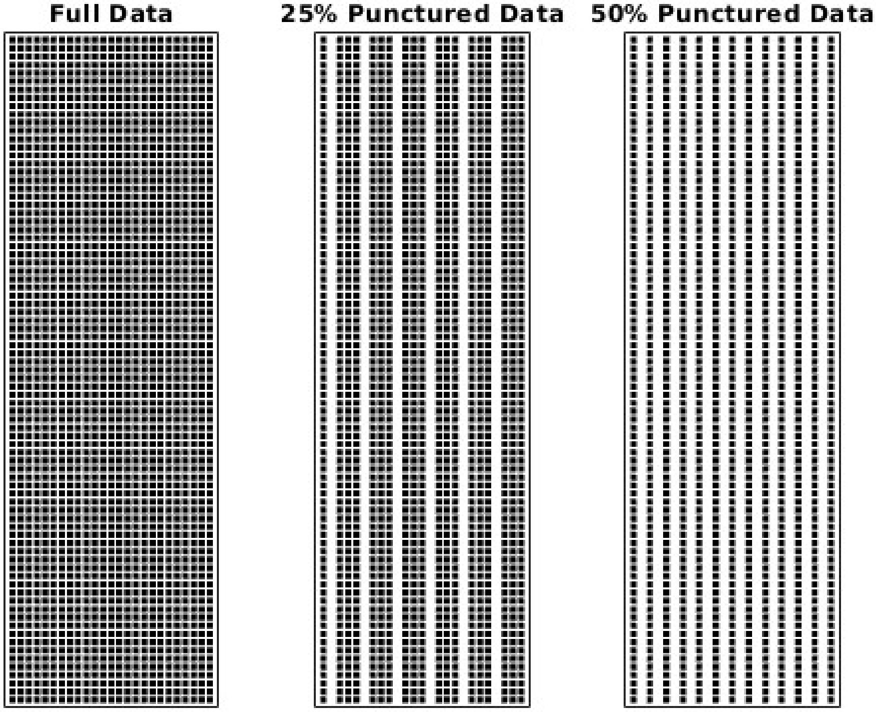

Figure 3.2: Examples of naive data puncturing

It can be observed that while the total amount of data is reduced, the distributions of synergy and redundancy remain nearly the same, making it a less robust data puncturing approach.

3.3.2: Uniform Data Puncturing

In this approach, the data is generated at a uniformly distributed random sampling of angles of attack. This approach differs from the naive puncturing in the fact that each geometry doesn't evaluate the performance at the same subset of airfoils.

Figure 3.3 shows the data availability plots of the punctured datasets for different levels of uniform puncturing. Each row corresponds to a diferent geometry, and each column corresponds to a different angle of attack.

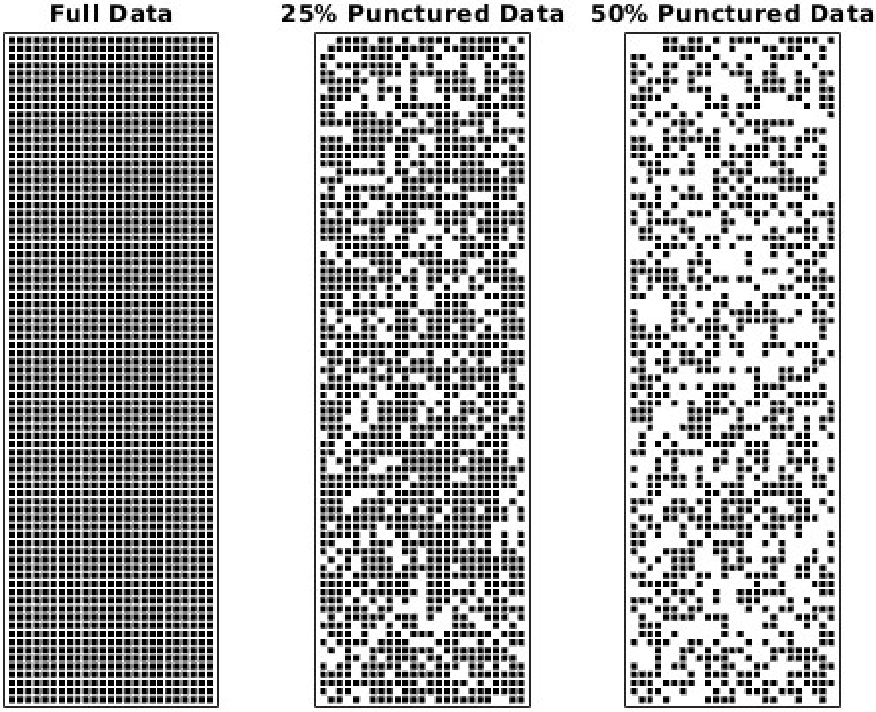

Figure 3.3: Examples of uniformly punctured datasets

Observe that, while the variation with angle of attack is not completely inferable in each geometry, there are different geometries that contain this information. That is, some data points carry information of the variation of performance with angle of attack, while a few points together carry the information about the variations with geometry at a given angle of attack. In a way, the collective information of all the geometries is nearly similar to that of unpunctured data, indicating that the synergy of the data is sustained and redundancy is reduced in this approach.

In comparison, naive approach doesn't have more information on variation with angle of attack for a fixed geometry, and has redundant information on variations with geometry.

3.3.3: Selective Data Puncturing

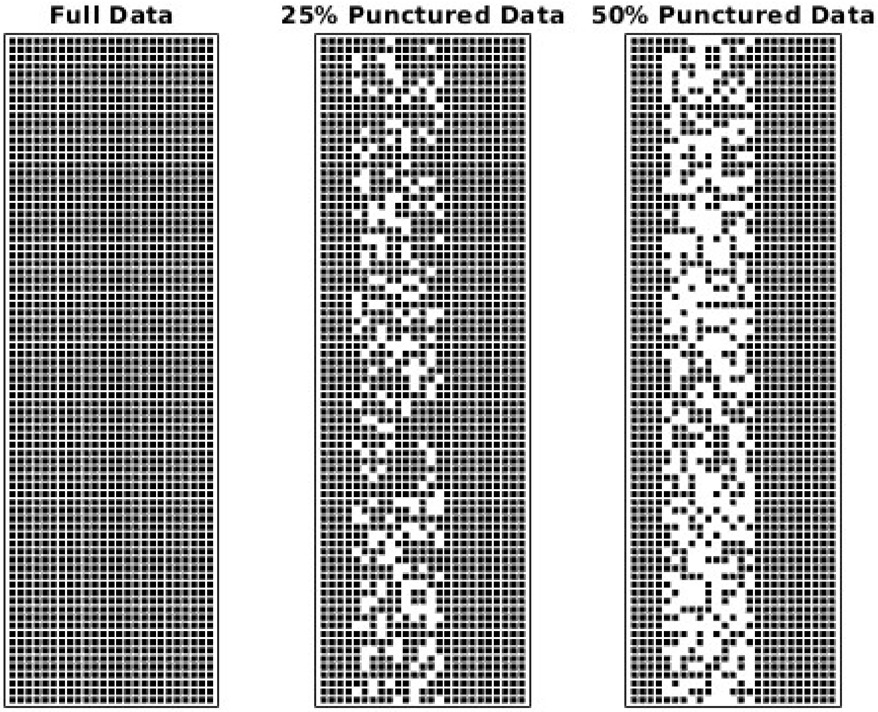

Figure 3.4: Examples of selectively punctured datasets

A third approach of data puncturing is to selectively bias the puncturing strategy for certain regions of the data. For example, the lift curve is linear for almost all airfoils in the database at low positive angles of attack. Here, redundancy minimization can be done by selectively increasing the puncturing proportion for that range of angles of attack.

Figure 3.4 shows the data availability plots of the punctured datasets, where the puncturing is selectively performed at a certain range of angles of attack. Each row in the figure corresponds to a different geometry, and each column corresponds to a different angle of attack.

In this approach, since the puncturing is biased towards the regions of high redundancy, the proportion of synergy to redundancy in fact increases. However, this requires knowledge of the regions of high redundancy in the data.

3.4: Modifications to the Training Approach

While the data puncturing aims to generate a dataset with high synergy, the training of surrogates with punctured data becomes a little tricky. This is mainly because the surrogate model predicts performance at all angles of attack, while the data is available only at a few angles. The training objective should be affected only by the available data. This can be achieved by introducing a window function that can selectively activate when data is available and deactivate otherwise. The modified training objective function is shown in Equation 3.2, where the square of the error function is convoluted with the window function represented by a summation of Dirac delta distributions (δ). This convolution operation transforms the error of the surrogate model in functional space to a scalar value. The second term in the objective function is a regularization function, that prevents overfitting of the surrogate model.

$$J = \frac{1}{2}\frac{1}{\#(\text{Train})}\sum_{X\in\text{Train}}\left(Y(X) - Y_{\text{true}}(X)\right)^2 * \left(\frac{1}{\#(\text{Train(X)})}\sum_{\alpha\in\text{Train}(X)}\delta(\alpha)\right) + \frac{\lambda}{2}\left\|w\right\|_2^2 \tag{3.2}$$

The cardinality operator (#) is used in the window function so that the integral of the window function over the range of angles of attack is unity. This ensures that the relative contribution of each airfoil to the overall objective function is same across all the geometries in the training set.

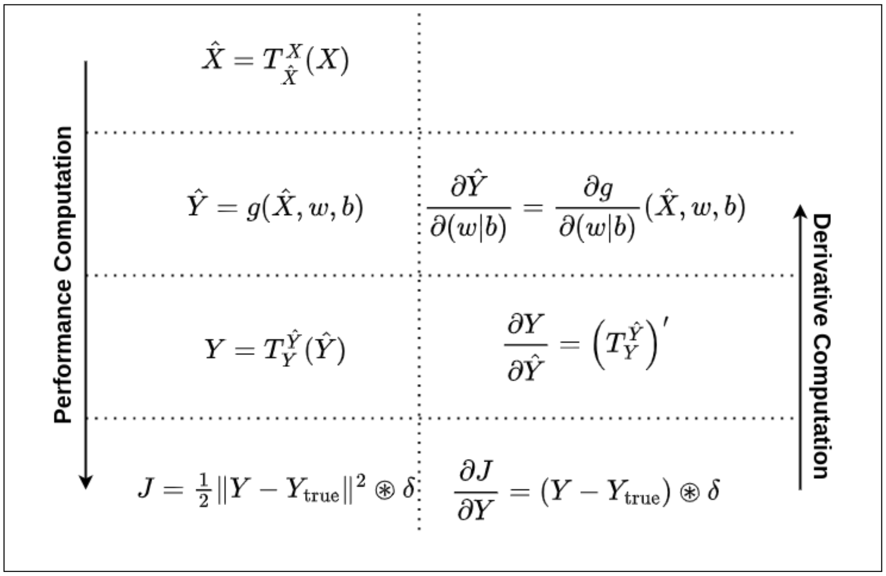

Figure 3.5: Schematic of the data flow for forward and reverse computations with modifications to accommodate data puncturing methodology

Figure 3.5 shows the changed sequence of computations for forward mode (estimation of aerodynamic polars) and reverse mode (derivative calculation), with the addition of the window operation to the cost function. The computed derivatives of the cost function is used to perform a gradient based optimization to achieve optimal weights and bias values for the hidden layers in ANN.

3.5: Performance of the Puncturing Approach

Throughout this study, the transformation operators and the mode shape functions for both first and last step of the surrogate model (Equations 2.14, 2.16) are fixed and not changed with any style of data puncturing. Figure 3.6 shows the different styles and levels of puncturing tested and compared.

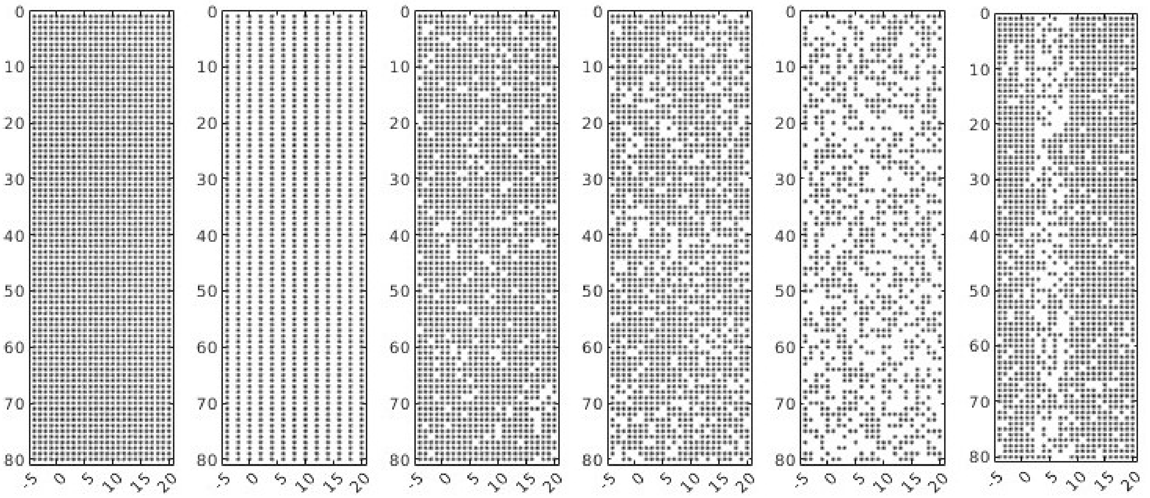

Figure 3.6: Test matrix of puncturing styles - (left to right): No puncturing (Full data), 50% Naive puncturing, 10% Uniform puncturing, 20% Uniform puncturing, 50% Uniform puncturing, 20% Selective puncturing

All styles and levels of data puncturing are performed on the airfoil dataset after removing the baseline airfoil set. The baseline airfoil set is chosen as a standard benchmark to compare the performance of each surrogate model trained using these punctured sets. k-fold cross-validation is applied with $k = 5$. Each of these surrogate models have two hidden layers with 25 neurons each, and are initialized with Xavier weights initialization, and trained using batch-stochastic gradient descent algorithm. The architecture of the surrogate model is held constant across all the tested styles of puncturing.

Table 3.1: Quantitative error estimates of different styles and levels of data puncturing

Dataset	Error norm	c_l	L/D	c_m
0% Data puncturing	$\langle L_2 \rangle$	1.39×10^{-2}	8.49×10^{-2}	1.04×10^{-3}
	$\langle L_\infty \rangle$	3.41×10^{-2}	2.12×10^{-1}	2.20×10^{-3}
50% Naive puncturing	$\langle L_2 \rangle$	2.48×10^{-1}	$1.52 \times 10^{+0}$	1.86×10^{-2}
	$\langle L_\infty \rangle$	8.34×10^{-1}	$5.23 \times 10^{+0}$	5.47×10^{-2}
10% Uniform puncturing	$\langle L_2 \rangle$	1.45×10^{-2}	8.79×10^{-2}	1.08×10^{-3}
	$\langle L_\infty \rangle$	3.62×10^{-2}	2.26×10^{-1}	2.37×10^{-3}
20% Uniform puncturing	$\langle L_2 \rangle$	1.42×10^{-2}	8.70×10^{-2}	1.08×10^{-3}
	$\langle L_\infty \rangle$	3.31×10^{-2}	2.08×10^{-1}	2.18×10^{-3}
50% Uniform puncturing	$\langle L_2 \rangle$	1.42×10^{-2}	8.68×10^{-2}	1.07×10^{-3}
	$\langle L_\infty \rangle$	3.37×10^{-2}	2.12×10^{-1}	2.22×10^{-3}
20% Selective puncturing	$\langle L_2 \rangle$	1.39×10^{-2}	8.46×10^{-2}	1.04×10^{-3}
	$\langle L_\infty \rangle$	3.31×10^{-2}	2.06×10^{-1}	2.13×10^{-3}

Each surrogate model is tested on the set of 12 baseline airfoils and Table 3.1 and Figure 3.7 show the averages of L_2 and L_∞ errors obtained by comparing with true data. Figure 3.8 shows the scatter plots of expected output vs true output of each surrogate model for the lift coefficient for all the baseline airfoils. Similar trends are observed for lift-to-drag ratio and moment coefficient. Surrogate models trained on both uniform puncturing and selective puncturing datasets show similar level of accuracy as model with no puncturing. However, the model with naive puncturing shows significantly higher errors compared to other models. This is also confirmed in the regression plots of the expected output vs true output. Figure 3.9 shows the obtained performance outputs for a subset of the baseline set showing the accuracy over a range of geometries. While no noticeable difference can be seen for most of the surrogate models, the naive approach results are significantly different from the others.

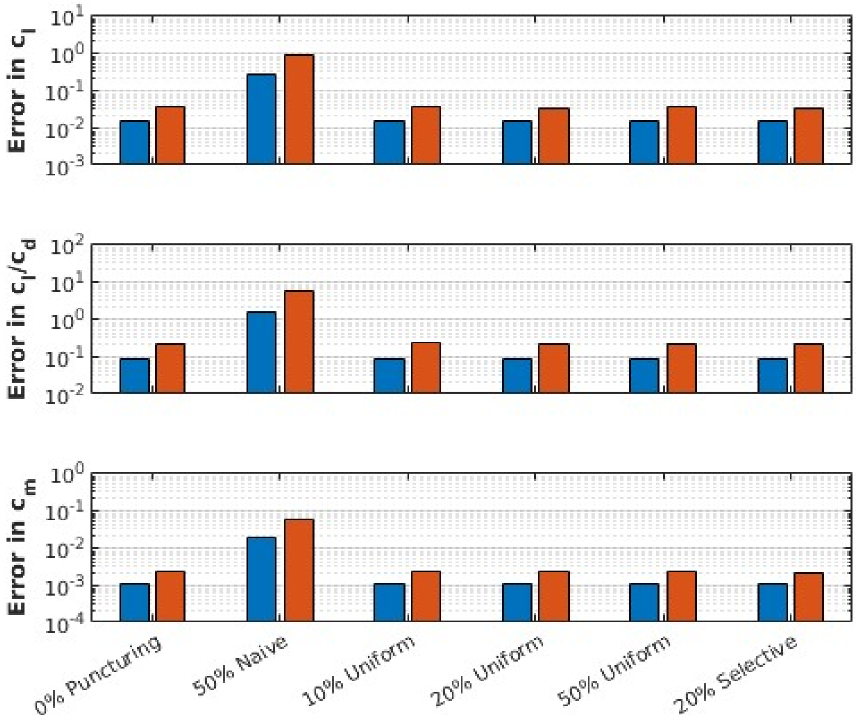

Figure 3.7: Error estimates of each surrogate model for the baseline airfoils - Blue: $\langle \mathbf{L_2} \rangle$ Error, Red: $\langle \mathbf{L_\infty} \rangle$ Error - Columns (left to right): No puncturing (Full data), 50% Naive puncturing, 10% Uniform puncturing, 20% Uniform puncturing, 50% Uniform puncturing, 20% Selective puncturing - Rows (top to bottom): lift coefficient, lift-to-drag ratio, pitching moment coefficient

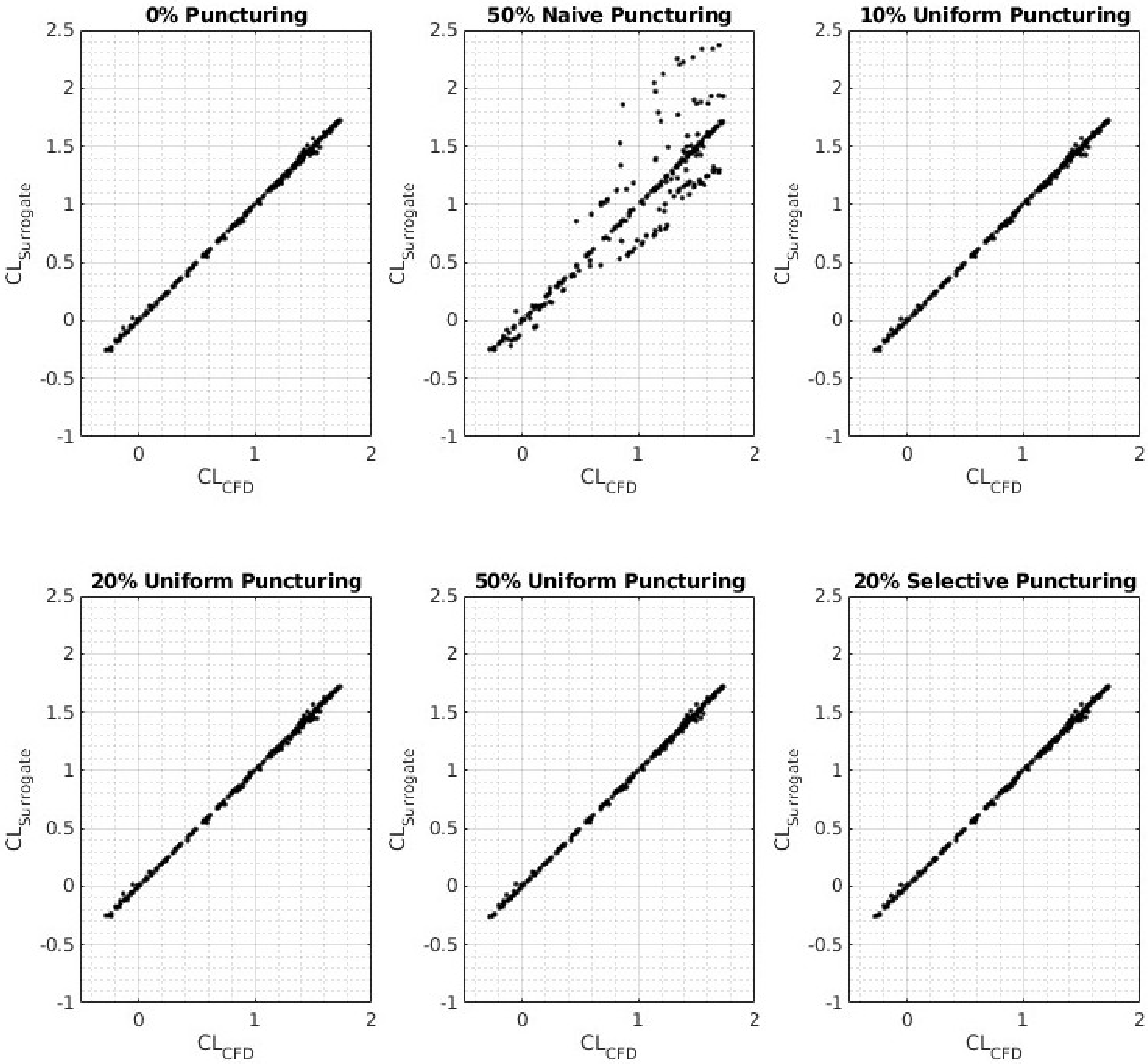

Figure 3.8: Comparison of the lift coefficient predicted by the surrogate models with the true data

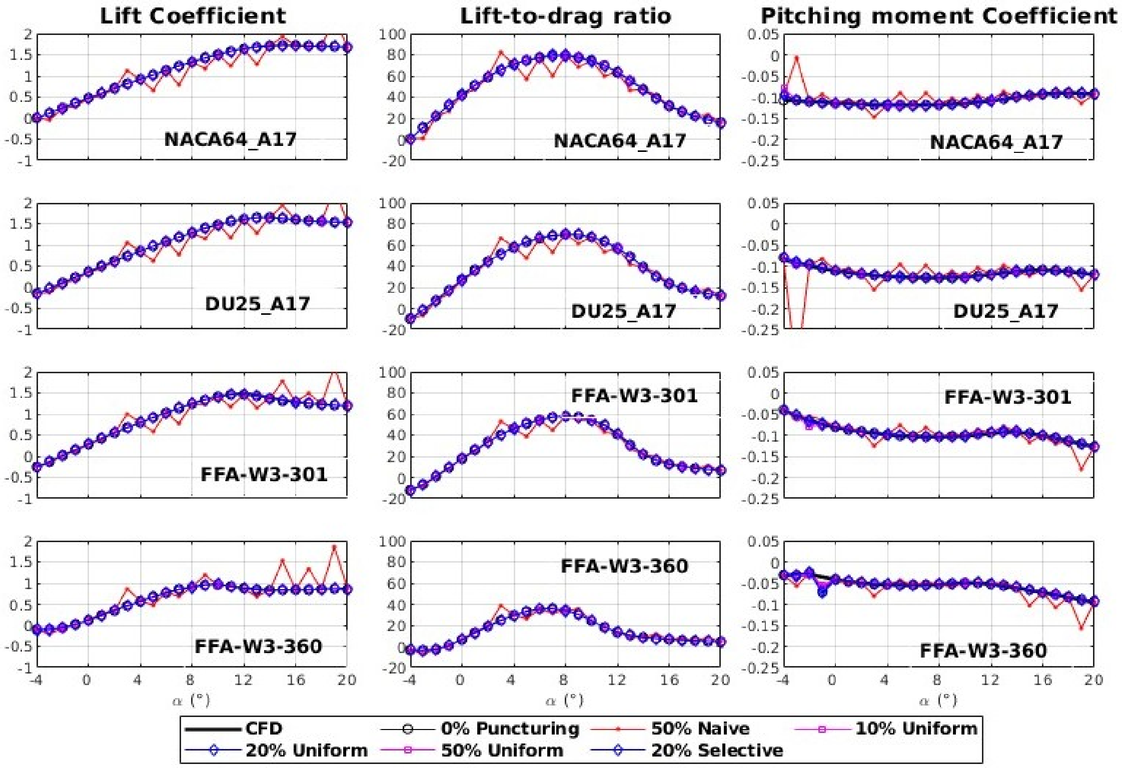

Figure 3.9: Performance of the surrogates for representative airfoils in various thickness categories - Rows (top to bottom): NACA64_A17, DU25_A17, FFA-W3-301 and FFA-W3-360 airfoils - Columns (left to right): lift coefficient, lift-to-drag ratio, pitching moment coefficient

3.5.1: Naive Puncturing

The surrogate model trained on the naive punctured dataset was the least accurate model compared to other styles of puncturing. The dataset is punctured at the same angle of attack for all the airfoils. In the backpropagation and gradient descent for this surrogate model, certain weights of the ANN do not get updated because of tight coupling with the missed angles of attack. This results in significantly poor prediction at the missing angles of attack. This is confirmed from the results in Figure 3.9. Naive puncturing is thus a poor strategy to reduce the training data generation costs, as it significantly lowers data synergy.

3.5.2: Uniform Puncturing

Uniform puncturing model results are similar to the full data models, with insignificant variation in accuracy over different levels of puncturing. Experiments with further levels of puncturing show increase in error norms. In terms of synergy and redundancy, as the level of puncturing slowly increases from 0, redundancy of the data continuously decreases with very minimal loss in synergy. Further increase in the level of puncturing decreases the synergy and we see rise in error norms. This is confirmed by the error trends, where the error first grows around and beyond stall angles of attack for lift coefficient, and around the maximum point for the lift-to-drag ratio and slowly globalize on the entire range of angles of attack. Figure 3.10 shows the observed increase in error norms for lift coefficient at higher levels of puncturing. Similar growth is observed in the other performance parameters. Since the puncturing is performed at random, some airfoils may have data at very few angles of attack. This could cause some variability in the results, but the trend line of error growth is expected to be similar.

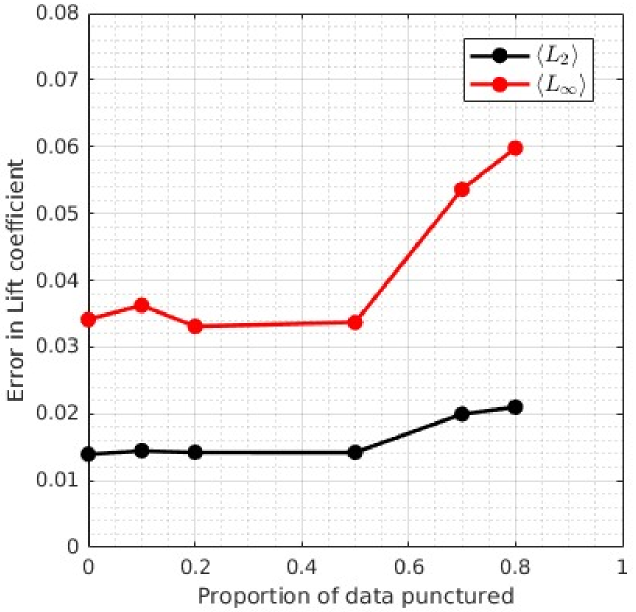

Figure 3.10: Growth of lift coefficient error norms for different levels of uniform puncturing

3.5.3: Selective Puncturing

The selective puncturing can be performed using any suitable non-uniform distribution that increases puncturing level at redundant domains and lowers it at synergistic domains. The 20% selective puncturing dataset used in this study is generated by performing a 10% uniform puncturing between $-4^\circ \leq \alpha \leq 2^\circ$ and $9^\circ \leq \alpha \leq 20^\circ$, and 50% uniform puncturing between $3^\circ \leq \alpha \leq 8^\circ$. This dataset performed well, with slightly better accuracy than its constituent 10% and 50% uniformly punctured sets as well as a similar level 20% uniformly punctured set. While the observed difference in accuracy is very small, this dataset has a slightly better distribution of redundancy in the domain, which ensures uniform confidence in the model across all range of angles of attack when compared with 20% uniformly punctured dataset.

3.6: Summary

Methods to reduce the data generation costs for developing fast surrogate models are investigated. Data puncturing methods allow to reduce the training data and with a modified training strategy, these datasets can be used to develop surrogate models. Three different styles of data puncturing viz., naive, uniform and selective, are developed and compared with fully populated datasets in terms of their synergy and redundancy, as exhibited by the accuracy of the surrogate models generated on these sets. Some key conclusions are

(1) Naive puncturing is a poor strategy to reduce data size and maintain synergy.

(2) Training with uniform puncturing consistently achieves accuracies similar to that of full dataset, over a wide range of puncturing levels.

(3) As the proportion of puncturing increases, first redundancy decreases and slowly synergy decreases with a lag, causing a plateau of stationary errors followed by a drop in accuracy.

(4) Selective puncturing can be leveraged to achieve a nearly uniform distribution of synergy to redundancy proportion, but requires the knowledge of distribution of redundancy in the dataset

Chapter 4: Uncertainty and Sensitivity Analysis

With the usage of modeling and simulation tools like CFD, and the development of high performance computing frameworks and architectures, CFD simulations have become an integral part in understanding the operational behavior of wind turbines as well as to make design and control decisions for optimal performance. While the estimation of fluid behavior using CFD can be performed for a given set of operating conditions, there exists an inherent uncertainty in all these conditions, which propagate into an uncertainty in performance. Uncertainties in wind turbine performance can occur due to operating conditions (freestream flow properties and relative orientations) as well as blade geometry uncertainties. Using the simulation tools repetitively for all possible operating conditions is impossible, and there is a need to quantify the uncertainty associated with the estimates with a reasonable cost.

Several uncertainty quantification (UQ) methods are developed and used in CFD based applications, including Monte Carlo simulations (MCS) [59], moment methods like polynomial chaos expansion (PCE) [60]. MCS involve generating the output performance for a distribution of inputs, and thereby approximate the probability distribution of the output for uncertain inputs. PCE on the other hand expresses the uncertainty in outputs as an expansion of polynomial terms, and the magnitudes of these terms can be obtained using relatively small number of simulations. While PCE is computationally less expensive than MCS, the accuracy of PCE based

UQ estimates are limited by the assumption of how accurately the polynomial expansion captures the uncertainty.

A major reduction in cost of MCS can be achieved by using surrogate models to replace the expensive full-order simulation. This chapter uses the surrogate model developed in the previous chapters to build an uncertainty quantification framework for the airfoil performance quantities.

4.1: Inherent Model Uncertainty

Relation between a surrogate and its training data is given in Equation 4.1, where ϵ is a component of the variation of true data that is not learnt by the surrogate.

$$\hat{y} = \hat{f}(x) = f(x) + \epsilon \tag{4.1}$$

That is, if the true relation of the output y variation with the input x is f, the surrogate model $\hat{f}$ has an inherent uncertainty with a distribution given by ϵ. Assuming that ϵ follows a normal distribution, an estimate of the model error variance is given as an expectation value operation on the error function across the entire domain of inputs x, as shown in Equation 4.2. Since the database of airfoils is a representative of the domain of wind turbine airfoil geometries, the expectation value can be approximated using the computation on this subset of airfoils.

$$\sigma_M^2 = E(\epsilon^2) = E\left[\left(f(x) - \hat{f}(x)\right)^2\right] \approx \text{MSE} \tag{4.2}$$

Since the training dataset is a representative subset of the population, the estimate of the inherent model uncertainty is given by the mean squared error of the surrogate on the training data. The L_2 error in Table 2.3 is an estimator of the standard deviation, and the corresponding 3σ confidence bounds give a 99.7% confidence interval of the surrogate model estimate.

4.2: Propagation of Geometric Uncertainty

Geometric uncertainty in the airfoil is modeled by varying each of the design variables using an independently distributed normal random variable. A nominal value of 5% variation in the magnitude of the CST coefficients is chosen as the standard deviation of the corresponding normal distribution of each parameter. A Monte-Carlo simulation is performed using the surrogate model for $10,000$ sample designs generated using the random distribution. The associated uncertainty in the outputs due to the geometric uncertainty is computed using the standard deviation σ (Equation 4.3) of the predicted $10,000$ sample outputs, as an expectation value of the deviations from the base geometry $(\bar{x})$.

$$\sigma^2 = E\left[\left(\tilde{f}(x) - \tilde{f}(\bar{x})\right)^2\right] \tag{4.3}$$

Observe that this standard deviation is the actual uncertainty due to geometric variations, if the surrogate model is the true model. However, the surrogate model itself has an inherent uncertainty. Hence, the overall uncertainty estimates due to geometric uncertainty are given by Equation 4.4. The contributions of the inherent uncertainty and the geometric uncertainty estimates can be decoupled in the computation as they are independent. In Equation 4.4, σ^2 corresponds to the Monte-Carlo estimate of the uncertainty due to geometric variations, and σ_M^2 is the additional uncertainty added due to the model error propagated from the surrogate model.

$$\sigma_x^2 = E\left[(f(x) - f(\bar{x}))^2\right] \approx \sigma^2 + \sigma_M^2 \tag{4.4}$$

Figure 4.1 shows the uncertainty estimates of the performance curves associated with a 5% uncertainty in geometry, for NACA64-618 (18% thickness), DU91-W2-250 (25% thickness), FFA-W3-301 (30% thickness) and FFA-W3-360 (36% thickness) airfoils. The confidence bounds are computed using the model adjusted estimate in Equation 4.4.

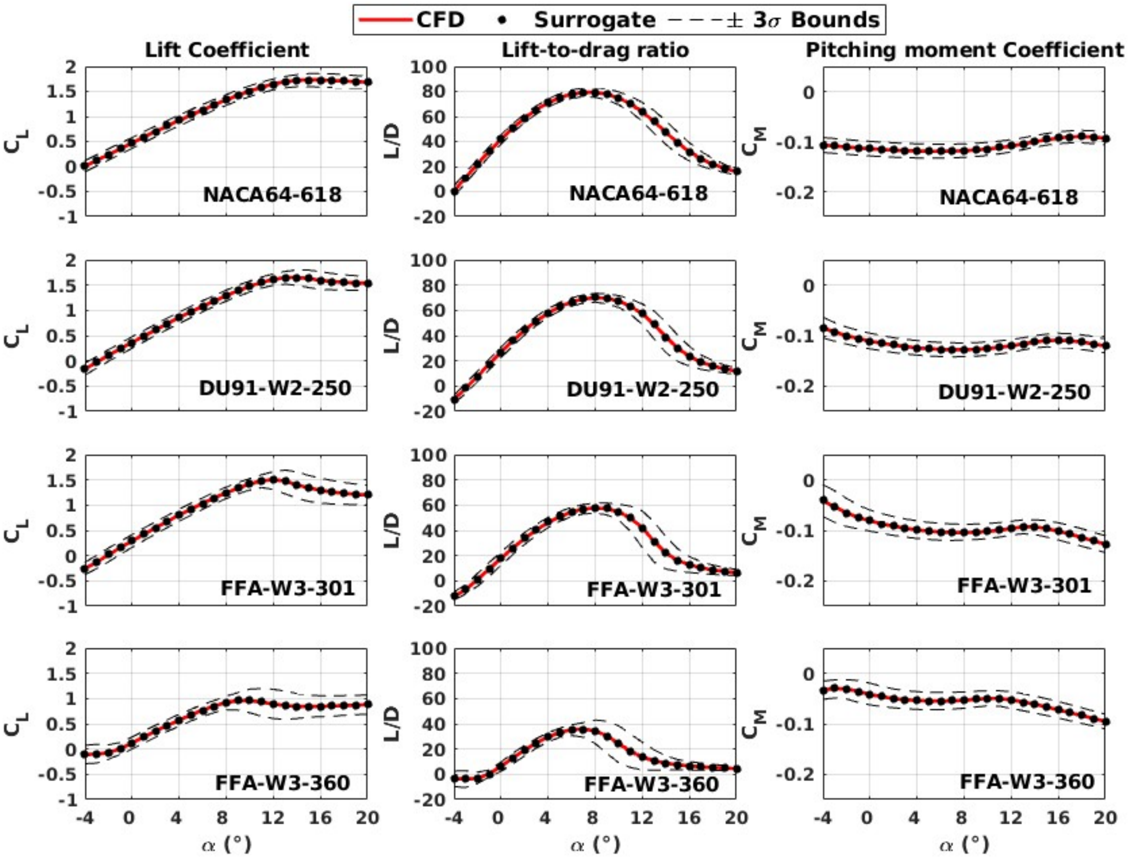

Figure 4.1: Uncertainty quantification of the surrogate model on representative airfoils in various thickness categories - columns (left to right): lift coefficient, lift-to-drag ratio, pitching moment coefficient respectively

The uncertainties associated with lift coefficient and lift-to-drag ratio are higher near the stall angle of attack. This is expected because the stall and boundary layer characteristics are highly sensitive to geometry of the airfoil. Near the design operating point (at max. lift-to-drag ratio), the corresponding standard deviations in lift coefficient, lift-to-drag ratio and pitching moment are in the ranges of 0.01-0.02, 1.1-3 and 0.003-0.005 respectively. Near the stall region, the corresponding ranges are 0.04-0.1, 4-7 and 0.003-0.005 respectively. For lift coefficient and lift-to-drag ratio, these values correspond to approximately 2-3% near the operating point and 5-10% near the stall region, with higher numbers being observed for thicker airfoils.

4.3: Global Sensitivity Analysis

Sobol's analysis [61], also known as Global Sensitivity Analysis (GSA), is a powerful tool to quantify the relative importance and contribution of each input in the distribution of the output of a multiple input system. The main idea of this analysis is to quantify the contribution of each input variable and their interactions in the variance of the output.

In the context of the airfoil performance as a function of geometric design variables, the influence of each design variable is quantified by means of two indices, viz., main-effect index and total-effect index. If X_i are the input variables and Y is the output, the indices are given by Equation (4.5), (4.6), with $X_{\sim i}$ representing all input variables except X_i. Var is the variance operator and E is the expectation value operator.

$$S_i = \frac{\mathrm{Var}_{X_i}(E_{X_{\sim i}}(Y|X_i))}{\mathrm{Var}(Y)} \tag{4.5}$$

$$S_{Ti} = \frac{E_{X_{\sim i}}(\mathrm{Var}_{X_i}(Y|X_i))}{\mathrm{Var}(Y)} \tag{4.6}$$

The main-effect index S_i measures the effect of varying X_i alone on the output, and the variation is estimated on an averaged quantity. The total-effect index S_{Ti} includes the effect of varying X_i, also accounting for the interaction effects with other variables $X_{\sim i}$. These indices are estimated using a quasi-Monte Carlo method.

Both the main effect and total effect indices are used to rank the importance of input variables and their interactions, to identify the design variables that can influence the performance quantities the most.

Observe that while the CST parametrization with Chebyshev polynomials has mathematical advantage in developing the surrogate model, the Chebyshev coefficients do not give a intuitive information to the designer compared to Bernstein coefficients. Figure 4.2 shows the CST class shapes corresponding to 4th order Bernstein polynomials and Chebyshev polynomials. The Bernstein coefficient number approximately governs the local region of influence of the shape function, i.e., for 4th order polynomials, 0th shape function majorly affects the leading edge region, 1st one near quarter-chord point etc. Thus, sensitivity analysis with respect to Bernstein polynomials gives more insight to the designer than Chebyshev polynomials.

The conversion between the coefficients in both the representations is a linear algebraic transformation, and there is no loss in representation accuracy if the order of the polynomials is maintained the same.

A quasi-Monte Carlo method is employed to estimate the influence indices for the coefficients of Bernstein polynomials.

Figure 4.3 shows the main-effect indices of each of the geometric parameters corresponding the uncertainty in lift, pitching moment and lift-to-drag ratio. All these quantities are calculated for the same set of airfoils at an angle of attack of $5°$. This angle of attack falls within the range

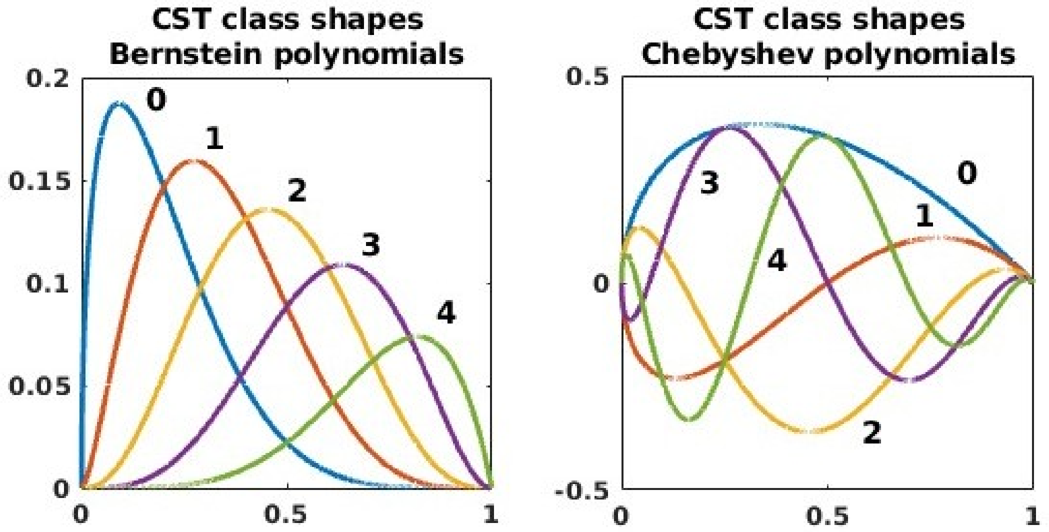

Figure 4.2: Comparison of class shapes using Bernstein and Chebyshev polynomials

of attached flow conditions for all these airfoils. It can be seen that the indices are more uniform across all design variables in case of lift and pitching moment coefficients. However, the bottom surface is more influencing the lift-to-drag ratio compared to top surface.

It was also observed that the total-effect index estimates are both qualitatively and quantitatively similar to the main-effect indices. This is an indication that the cross influence indices are insignificant and the variance or uncertainty of the airfoil performance can be majorly explained by exclusive variations in each parameter.

A similar analysis is performed at a post-stall angle of attack of $15°$. Figure 4.4 shows the main-effect indices of each of the design parameters. Clearly, the top surface is found to be the most dominant variable, with the Sobol's index close to unity. Also, the trends of the indices is similar for both lift and lift-to-drag ratio. The cross influence indices are negligible even in this case, indicating very weak coupling between design variables.

Also, the most influential polynomial number is moving leftwards from NACA64_A17

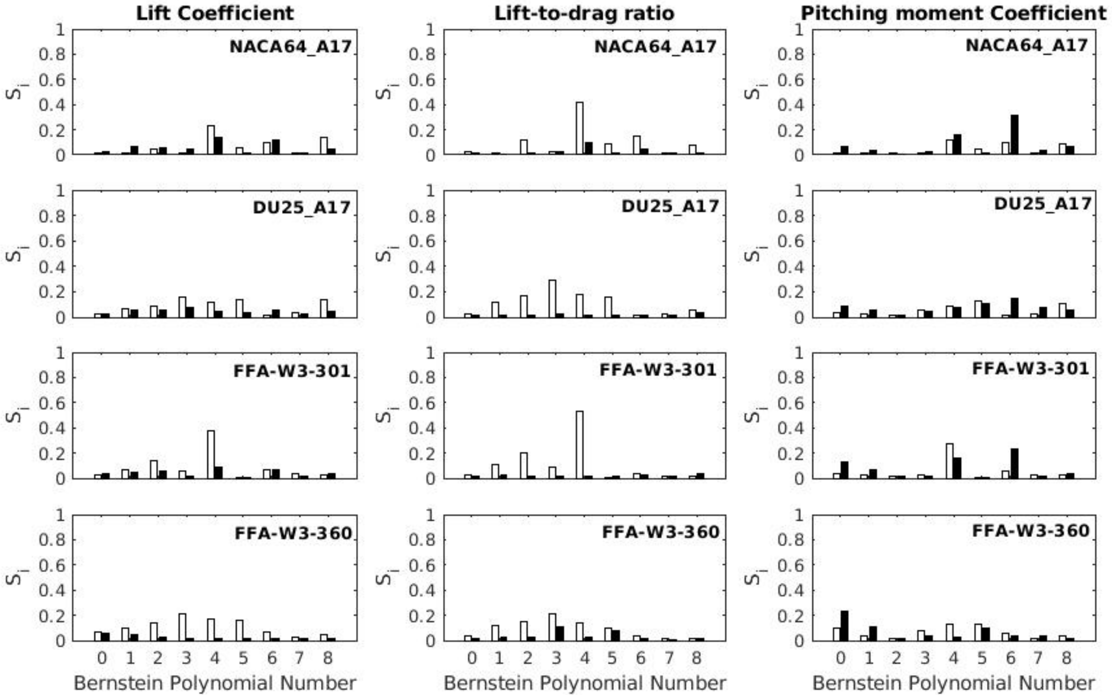

Figure 4.3: Main effect indices for Bernstein coefficients at a pre-stall angle of attack for representative airfoils in various thickness categories - columns (left to right): lift coefficient, lift-to-drag ratio, pitching moment coefficient respectively - hollow bars correspond to the bottom surface and filled bars are for top surface

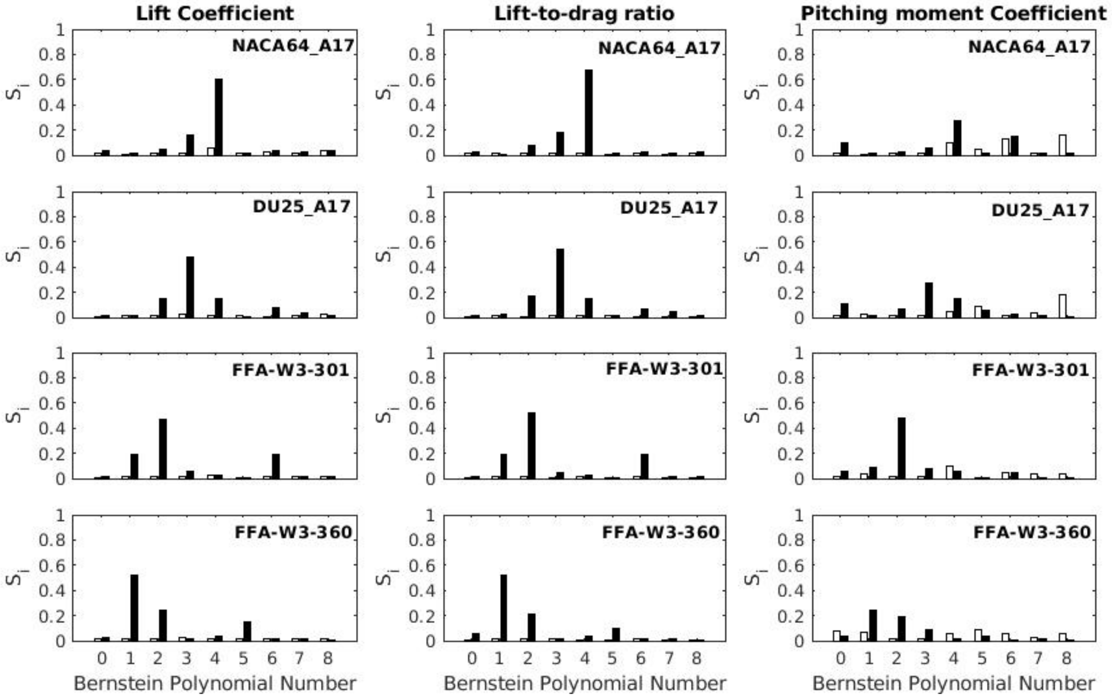

Figure 4.4: Main effect indices for Bernstein coefficients at a post-stall angle of attack for representative airfoils in various thickness categories - columns (left to right): lift coefficient, lift-to-drag ratio, pitching moment coefficient respectively - hollow bars correspond to the bottom surface and filled bars are for top surface

airfoil to FFA-W3-360 airfoil. By observing the geometries of these 4 airfoils, it can be clearly seen that these dominant polynomials correspond to the location of maximum thickness on the top surface. Thus, clearly, the location of maximum thickness will be a key influencing geometric variable for lift and lift-to-drag ratio of an airfoil operating at high angles of attack in separated flows.

4.4: Summary

The developed surrogate model is embedded into a Monte-Carlo based uncertainty quantification and sensitivity analysis framework. Some key takeaways are

(1) The uncertainty propagated through the uncertainties in geometric parameters are quantified using a Monte-Carlo simulation. A 5% uncertainty in the magnitudes of geometric parameters corresponds to 2-3% uncertainty in performance parameters at the operating point of the airfoils. This uncertainty increase as high as 10% for the lift and lift-to-drag ratio when the airfoil operates closer to stall region.

(2) Thicker airfoils show higher levels of uncertainty due to geometry at stall angle of attack compared to thin airfoils.

(3) Sobol's analysis predicted the bottom surface to be influential at the operating point and top surface is influential when operating in stall.

(4) The coupling between design variables for the airfoils is weak, i.e., no significant cross-correlations between the geometric parameters in terms of uncertainty propagation.

Chapter 5: Improvements to the Turbulence Model

5.1: Need for better Turbulence Models

In the previous chapters, machine learning techniques were employed to develop surrogate models for predicting the static aerodynamic polars of airfoils. It is observed that the accuracy of the surrogate model compared to the CFD data is very high when compared to the accuracy of the CFD models with experiments. Figure 5.1 compares the lift coefficients from experiments [62, 63] on different wind turbine airfoils with the corresponding CFD predictions using the Spalart-Allmaras (SA) turbulence model.

For all the airfoils shown, the accuracy of CFD when compared with experiments is high in the linear lift region. However, the onset of stall is predicted with high delays in CFD and occurs at significantly lower angles of attack in reality.

Figure 5.2 shows the comparison of experimental lift coefficient polars [64] at various Reynolds numbers with the corresponding CFD estimates. The comparison shows similar trends of low accuracy in predicting the onset of stall.

Stall phenomena on an airfoil occurs due to separation of flow from the surface of the airfoil. As the angle of attack increases, an adverse pressure gradient continuously builds on the suction side of the airfoil. Beyond certain angles of attack, the boundary layer can not support the resulting strong adverse pressure gradient and detaches from the surface. This causes a sudden

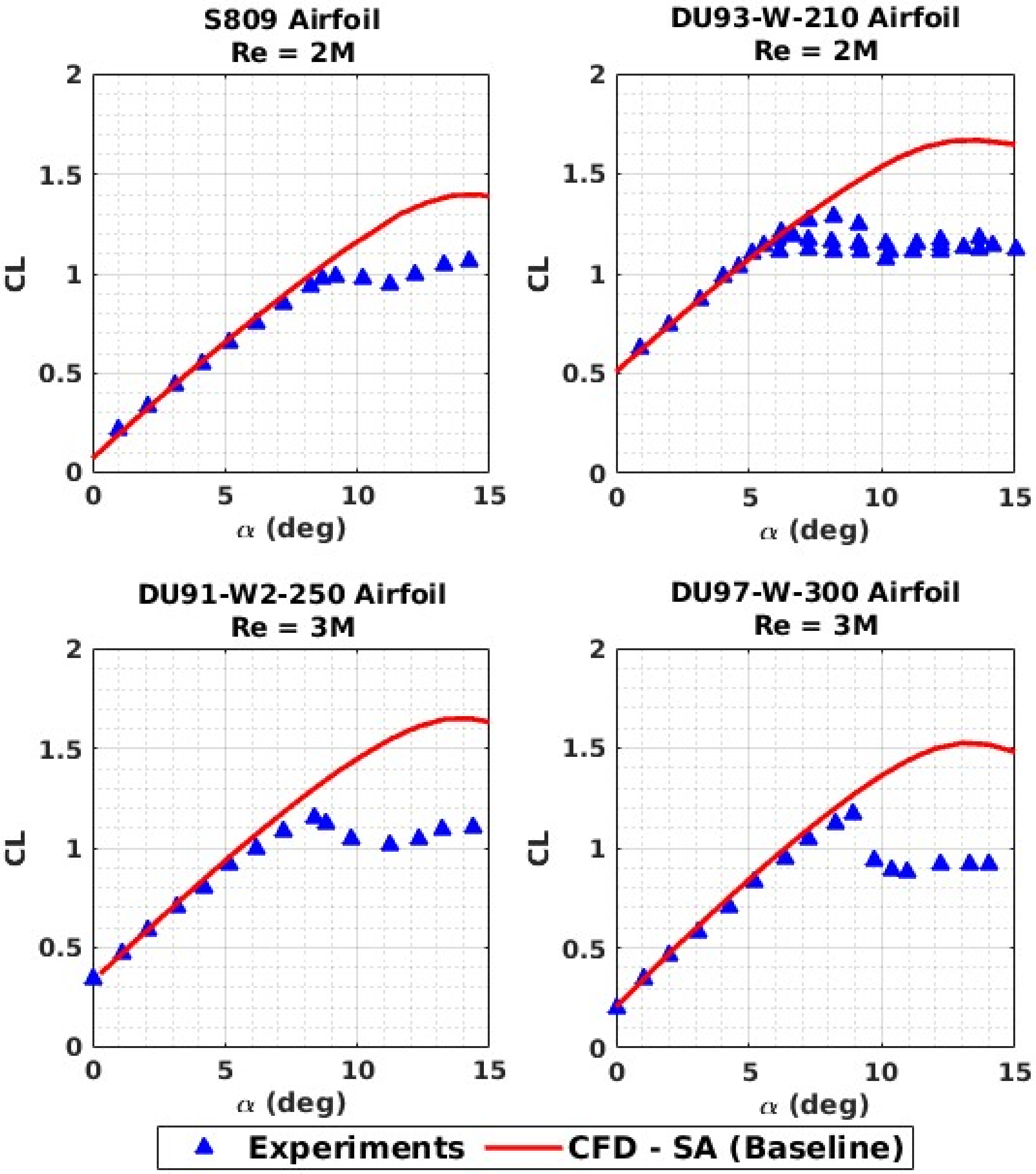

Figure 5.1: Comparison of experiments with CFD for different wind turbine airfoils

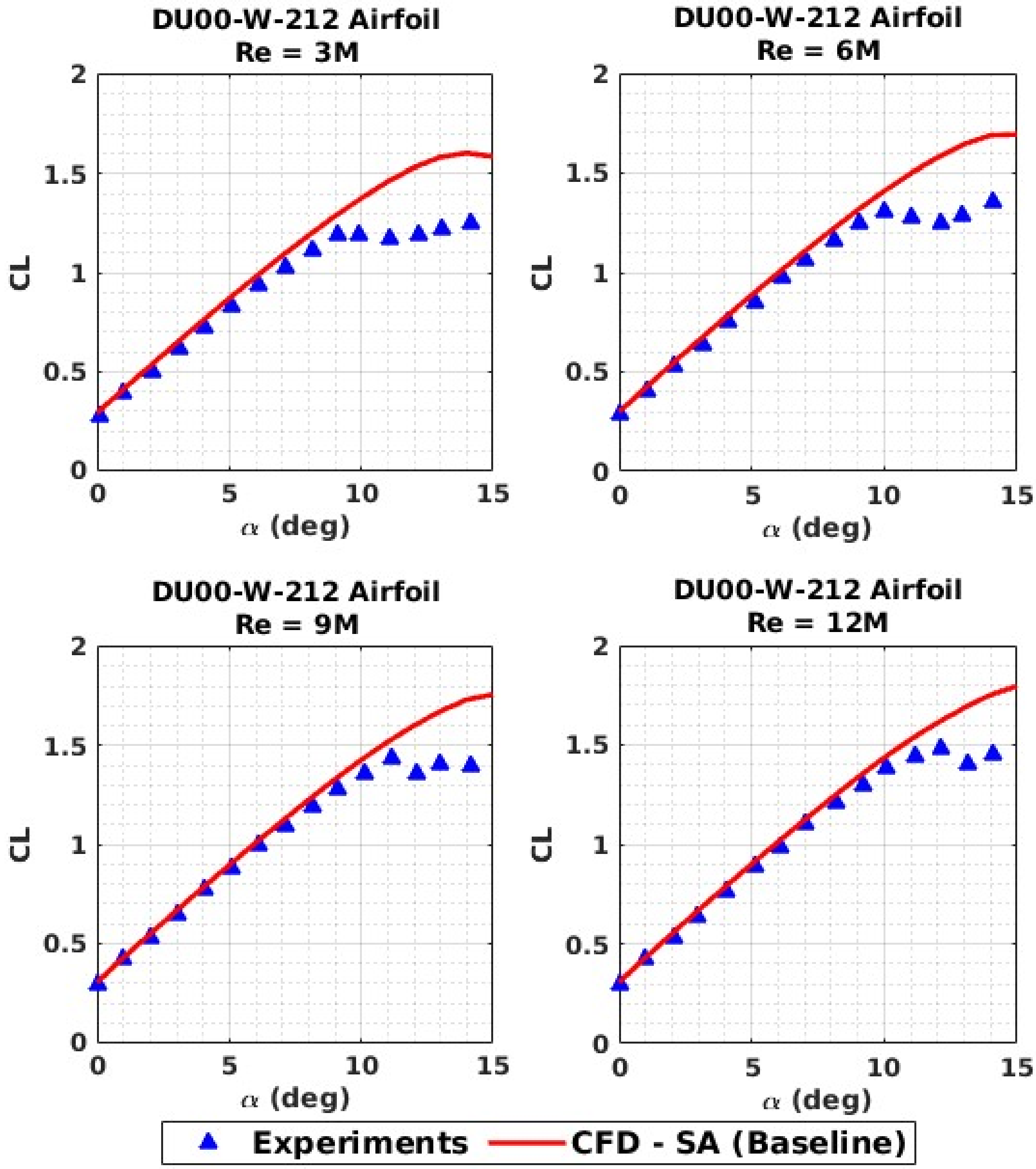

Figure 5.2: Comparison of experiments with CFD for DU00-W-212 airfoil at different Reynolds numbers

drop in lift and increase in drag forces on the airfoil. This phenomena is not accurately resolved by the current state-of-the-art RANS turbulence models.

The stall angle of attack is an important parameter for the operation of wind turbines. The operating conditions are ensured to produce maximum power output, at the angles of maximum lift-to-drag ratio [46] and when the airfoil operates in the stall region, the forces on the wind turbine blade suddenly change causing significant drop in power output. Furthermore, in stall, the blades experience high loading which can result in structural damages.

Additionally, a “pitch-to-stall” shutdown strategy is employed in certain wind turbines [65] to maintain the wind turbine within operating limits, in the events of increased wind speeds. In this strategy, as the wind speed increases beyond a certain threshold, the forces on the blades increase and leads to the turbine reaching its maximum rated power output. The pitch of the blade is increased to reduce the angle of attack and maintain the power generation of the turbine. If the wind speed continues to increase beyond, the blades are further pitched to operate in stall. This reduces the power output of the turbine to near zero and ensures safe operation of the turbine from reaching ultimate loads.

Thus, it becomes important to improve the CFD models to predict the onset of stall more accurately. This chapter discusses the SA turbulence model and employs the Field Inversion Machine Learning (FIML) methodology to arrive at a model correction that can improve the predictions of airfoil performance.

5.2: SA Turbulence Model

The SA turbulence model is based on the Boussinesq assumption, and is a one equation turbulence model describing the transport of eddy viscosity. The governing equation for the SA working variable ($\tilde{\nu}$) is given by Equation 5.1. The development of turbulent eddies in the flow is governed by the production, destruction and dissipation terms in the equation, represented by P, D and E respectively. The model is developed in consistency with the law of the wall, and calibrated with attached flows over airfoils.

$$\frac{D\tilde{\nu}}{Dt} = P(\tilde{\nu}, U) - D(\tilde{\nu}, U) + E(\tilde{\nu}, U) \tag{5.1}$$

The working variable ($\tilde{\nu}$) is defined to maintain consistency with the law of the wall, so that the eddy viscosity equals to the working variable away from the wall and follows ideal log-layer behavior very close to the wall.

The production term P represents the rate at which the eddy viscosity is generated in the flow. This term is proportional to the magnitude of the vorticity adjusted to maintain the log-layer behavior near the wall and free shear behavior away from the wall. Equation 5.2 shows the production term, where $\tilde{S}$ is the adjusted magnitude of vorticity and c_{b1} is an empirical constant.

$$P = c_{b1}\tilde{S}\tilde{\nu} \tag{5.2}$$

The destruction term (D) captures the effect of destruction of Reynolds stress terms due to the effect of the wall. The destruction term is inversely proportional to the square of the distance to the closest wall. Additionally a damping functional dependent on mixing length is introduced.

Equation 5.3 shows the destruction term, where d is the distance from the wall function, f_w is the damping function and c_{w1} is an empirical constant.

$$D = c_{w1} f_w \left(\frac{\tilde{\nu}}{d} \right)^2 \tag{5.3}$$

The dissipation term (E) is a non-conservative diffusion term consistent with the dissipation of the turbulent kinetic energy. Equation 5.4 shows the dissipation term with model constants c_{b2} and σ.

$$E = \frac{1}{\sigma} \left[\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu}) + c_{b2} (\nabla \tilde{\nu})^2 \right] \tag{5.4}$$

The model consists of several empirical constants defining each of these terms and the damping functions.

With the inherent empiricism in the model, the SA model is calibrated with wide range of turbulent flows, with extensive emphasis on the flows over airfoils. However, a primary limitation of the model is it's lack of accuracy in predicting flows operating in strong adverse pressure gradients and flow separation.

5.3: Field Inversion Machine Learning

The Field Inversion and Machine Learning (FIML) approach [66] is a predictive modeling framework used to obtain a data-driven correction to an underlying physical model. FIML methodology can be employed to obtain a correction to the SA turbulence model that can better predict the onset of stall in flows over wind turbine airfoils. The FIML methodology can be described as two sub tasks as shown in the flow chart in Figure 5.3. The first step is the field inversion, which involves obtaining an optimal case-specific correction to the turbulence model. The next step uses machine learning techniques to learn a functional correction to the turbulence model, consistent with the original model.

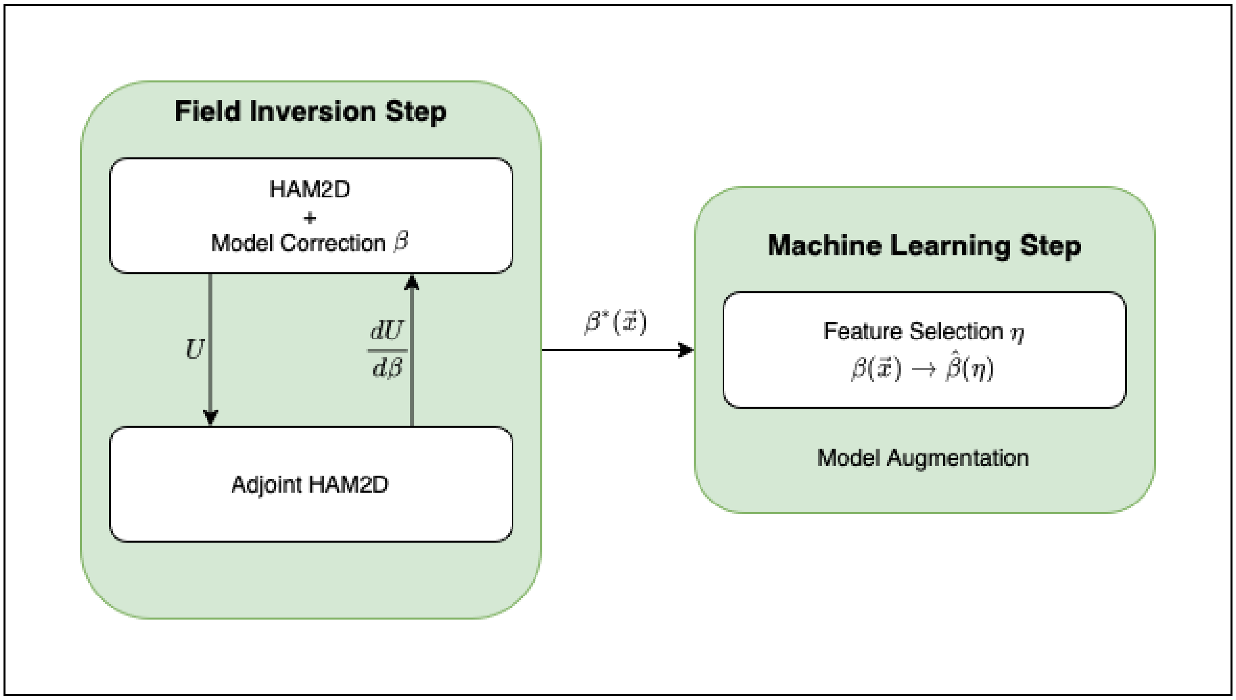

Figure 5.3: Flow chart describing the model correction strategy in FIML methodology

5.3.1: Model Correction

A correction field can be introduced into the SA model by means of a source term in the transport equation defined in Equation 5.1. The modified turbulence model is given by Equation 5.5.

$$\frac{D\tilde{\nu}}{Dt} = P(\tilde{\nu}, U) - D(\tilde{\nu}, U) + E(\tilde{\nu}, U) + \Delta(\vec{x}) \tag{5.5}$$

A suitable non-dimensionalization can be made to this correction by normalizing with either of the terms in the SA model. The modified turbulence model with the non-dimensional correction field is given by Equation 5.6.

$$\frac{D\tilde{\nu}}{Dt} = \beta(\vec{x})P(\tilde{\nu}, U) - D(\tilde{\nu}, U) + E(\tilde{\nu}, U) \tag{5.6}$$

The non-dimensional parameter β is a field variable and it represents the baseline SA model when $\beta = 1$ everywhere in the flow field.

5.3.2: Field Inversion

The Field Inversion step (FI) aims at obtaining a suitable correction field β for a specific flow scenario that matches the CFD predictions with experimental data. This can be represented as an optimization problem, with an objective function that quantifies the error between the numerical predictions and experimental data.

Let k_{exp} be the experimental data and k_{model} be the corresponding quantities predicted by CFD, then an L_2 norm of the error between these two quantities is the objective function. It must be noted that the experimental data need not be a flow field measurement, and can be just the integrated loads on the airfoil.

Since the free variable in optimization is a field variable, there may exist multiple field corrections that can minimize the objective function. However, all these fields may not be physical or consistent with the model. To introduce a well-posed optimization problem and to maintain model consistency, a regularization term is introduced into the objective function, that penalizes large deviations from the baseline model. The regularized objective function is given by Equation 5.7.

$$J(\beta) = \|k_{\text{exp}} - k_{\text{model}}(\beta)\|_2^2 + \lambda\|\beta - 1\|_2^2 \tag{5.7}$$

This data can represent a combination of multiple experimental quantities such as surface pressure coefficients, skin friction coefficients, integrated lift, drag etc. The majority of the experimental data for airfoils is available as the integrated quantities (lift, drag and moment coefficients). FIML implementations by Singh [38] showed similar improvements by using only the lift coefficient compared to using the surface pressure measurements. Hence, throughout this work, only the lift coefficient of the airfoil is used to drive the inverse problem for optimal correction. The field inversion problem for this work is thus mathematically represented by Equation 5.8.

$$\beta^* = \underset{\beta}{\operatorname{argmin}} \left((c_{l,\text{exp}} - c_{l,\text{CFD}}(\beta)) + \lambda\|\beta - 1\|^2 \right) \tag{5.8}$$

The optimization is performed using a gradient descent algorithm, and the gradients of the objective function with respect to the correction parameter β are estimated using the discrete adjoint approach [67].

The field inversion step is performed independently for various airfoils as shown in Table 5.1, and the corresponding sources of experimental measurements are referenced in the table. These airfoils range from 12% thickness to 30% thickness.

Field inversion for all these airfoils is performed at a representative range of angles of attack set of $\alpha = \{6°, 10°, 12°, 13°, 14°\}$. These angles of attack include flow scenarios with attached flow, onset of a strong adverse pressure gradient and separated flows.

Table 5.1: List of airfoil geometries used for Field inversion based on experimental data availability

Airfoil	Reynolds numbers	Data source
NACA0012	6M	[68]
S809	2M	[62]
DU93-W-210	2M	[63]
DU91-W2-250	3M	[63]
DU97-W-300	3M	[63]
DU00-W-212	3M, 6M, 9M, 12M	[64]

Figure 5.4 shows the sample optimal correction field results for S809 airfoil and DU97-W-300 airfoil at 6° angle of attack. These scenarios correspond to attached flow regions, where the SA baseline model predicts the lift coefficient quite accurately. Hence the magnitude of the correction field is small, as confirmed by the contours showing β close to 1.

Figure 5.5 shows the sample optimal correction field results for NACA0012 airfoil and DU93-W-210 airfoil at 10° angle of attack. These scenarios correspond to onset of stall, where the SA baseline model starts to deviate from the experimental lift coefficient. Hence the magnitude of the correction field is seen to increase as confirmed by the contours.

It must be noted that, in each of these cases, the correction field is $\beta = 1$ as we move away from the airfoil. This was ensured by the regularization term in the objective function.

Figure 5.6 compares the lift coefficients predicted using the optimal β field obtained from the field inversion for various airfoils at the testing set of angles of attack. Clear improvement of the lift coefficient is seen, with the FI predicted values close to the experimental data. The small differences arise due to the regularization term in the optimization.

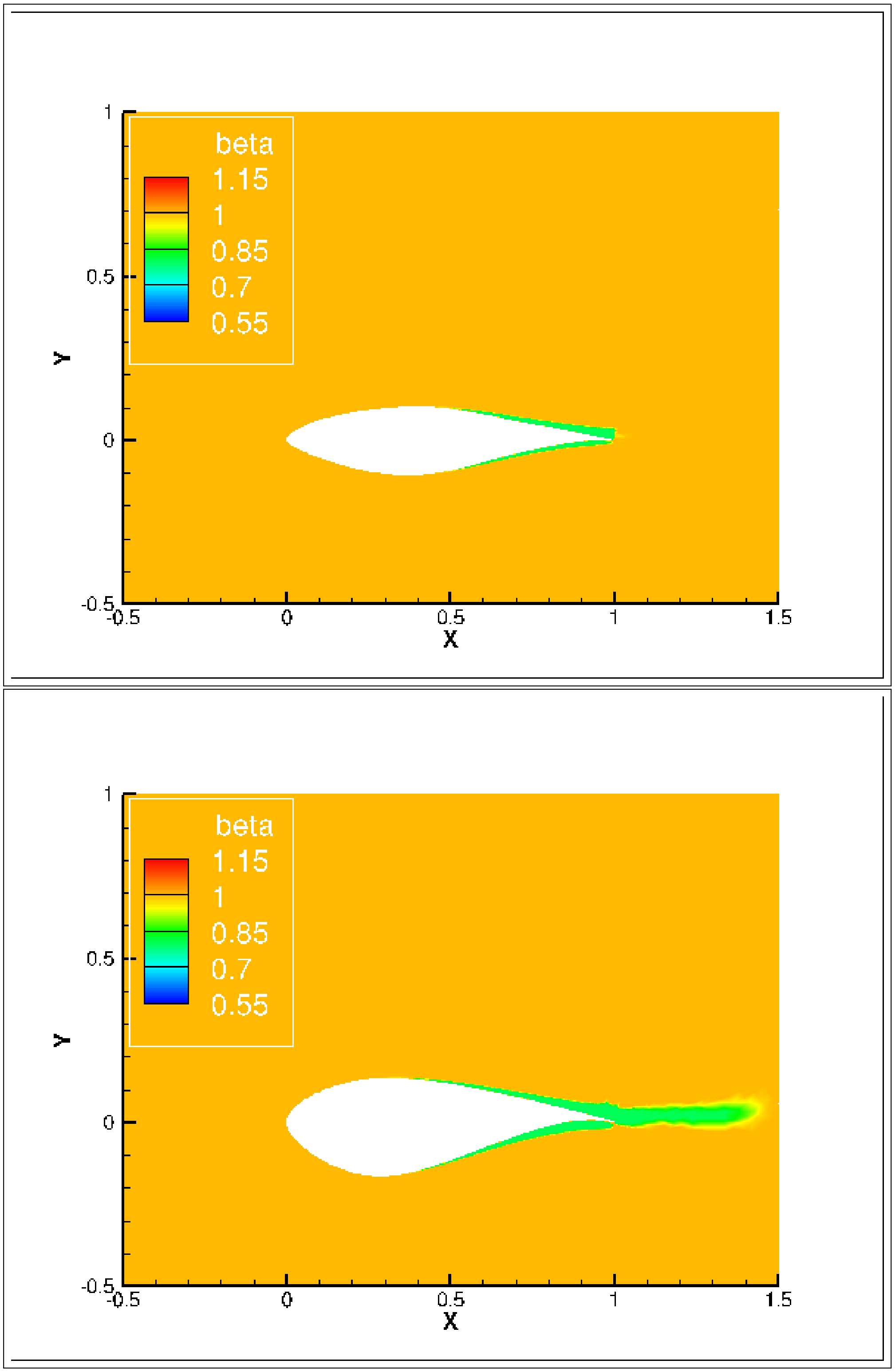

Figure 5.4: Optimal β field using field inversion for S809 airfoil and DU97-W-300 airfoil at $\alpha = 6^\circ$

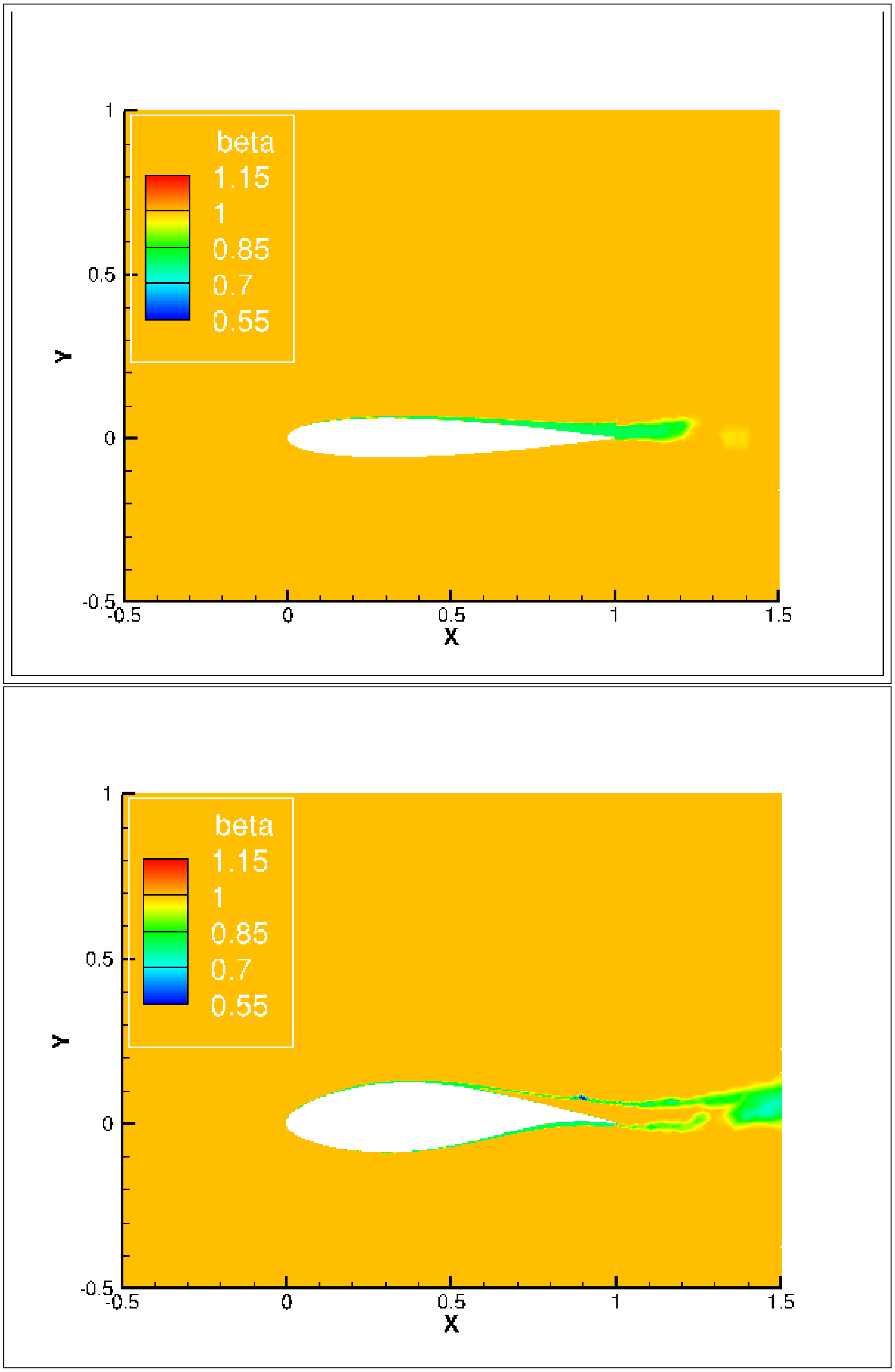

Figure 5.5: Optimal β field using field inversion for NACA0012 airfoil and DU93-W-210 airfoil at $\alpha = 10°$

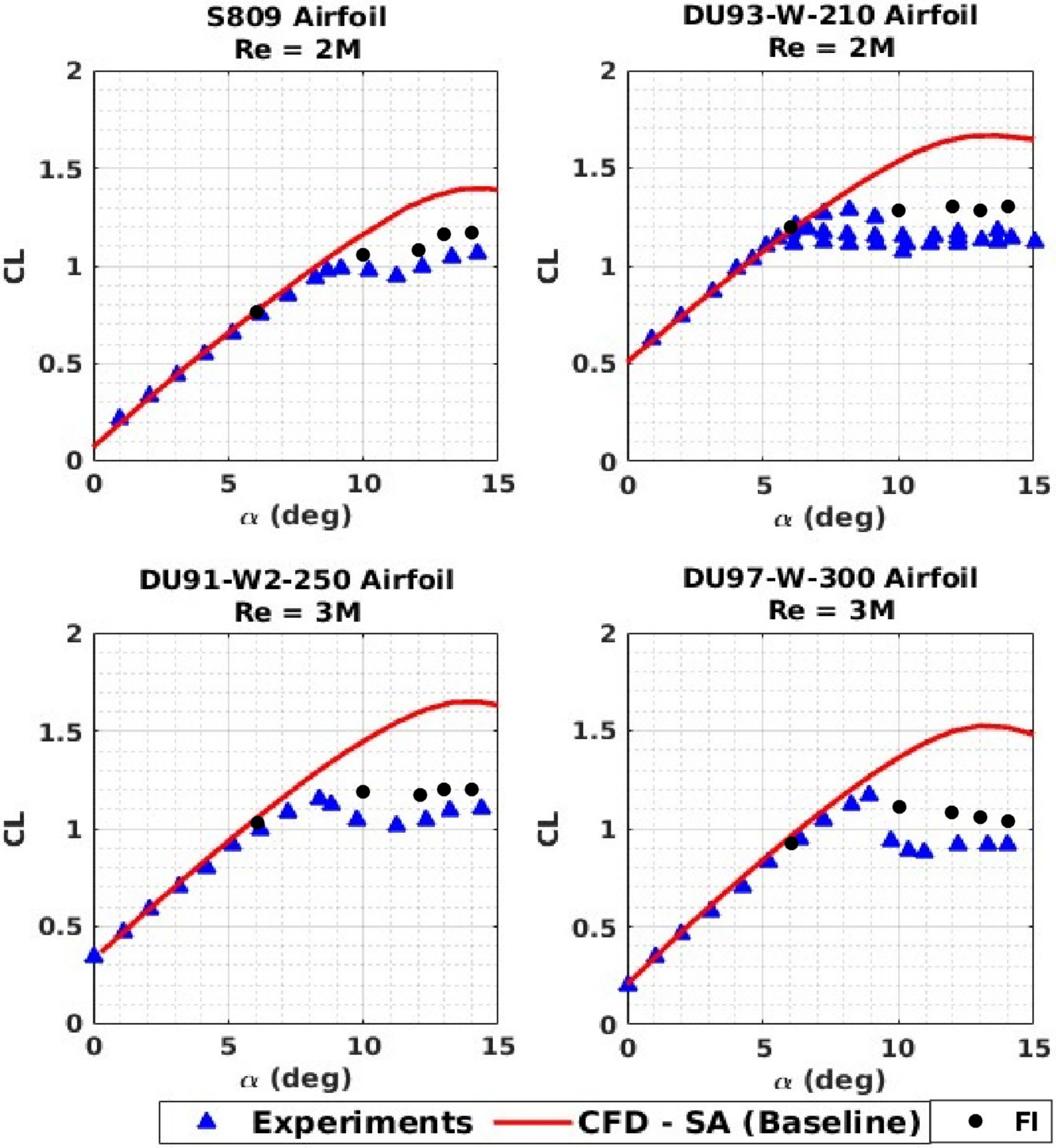

Figure 5.6: Lift coefficient improvements using Field Inversion, comparison with experiments

5.3.3: Machine Learning

While the field inversion obtains an appropriate optimal correction field to improve the turbulence model predictions, they are specific to the particular flow scenario. However, the obtained field contains physics-rich information and can be appropriately transformed into useful modeling information by learning a functional relationship projected in a suitable feature space (η). This obtained functional can be augmented with the baseline turbulence model and can be used for enhanced predictions of turbulent flows. Supervised learning techniques such as neural networks are powerful tools to extract complex functional relationships and are employed in this study to obtain a functional for the correction field term. The augmented turbulence model is given by Equation 5.9

$$\frac{D\tilde{\nu}}{Dt} = \hat{\beta}(\eta)P(\tilde{\nu}, U) - D(\tilde{\nu}, U) + E(\tilde{\nu}, U) \tag{5.9}$$

5.3.3.1: Selection of Flow Features

The selection of suitable features is important to appropriately model the functional for the correction term. A locally non-dimensional quantity that directly comes from the turbulence working variable is χ, defined as the ratio of the eddy viscosity working variable to the laminar viscosity. Similarly, the ratio of local strain rate magnitude to vorticity magnitude is another feature considered for this study. A third feature δ is defined as the ratio of local turbulent strain to the wall shear stress, appropriately scaled as in Equation 5.10. This parameter was studied by Shivaji [69] to indicate the effect of adverse pressure gradient in separating flows over airfoils.

$$\delta = \frac{1}{1.5}\frac{\mu_t|S_{ij}|}{\tau_w} \tag{5.10}$$

Equation 5.11 shows the three variables chosen as the set of flow features influencing the β correction term.

$$\eta = \left\{\chi, \frac{|S|}{|\Omega|}, \delta\right\} \tag{5.11}$$

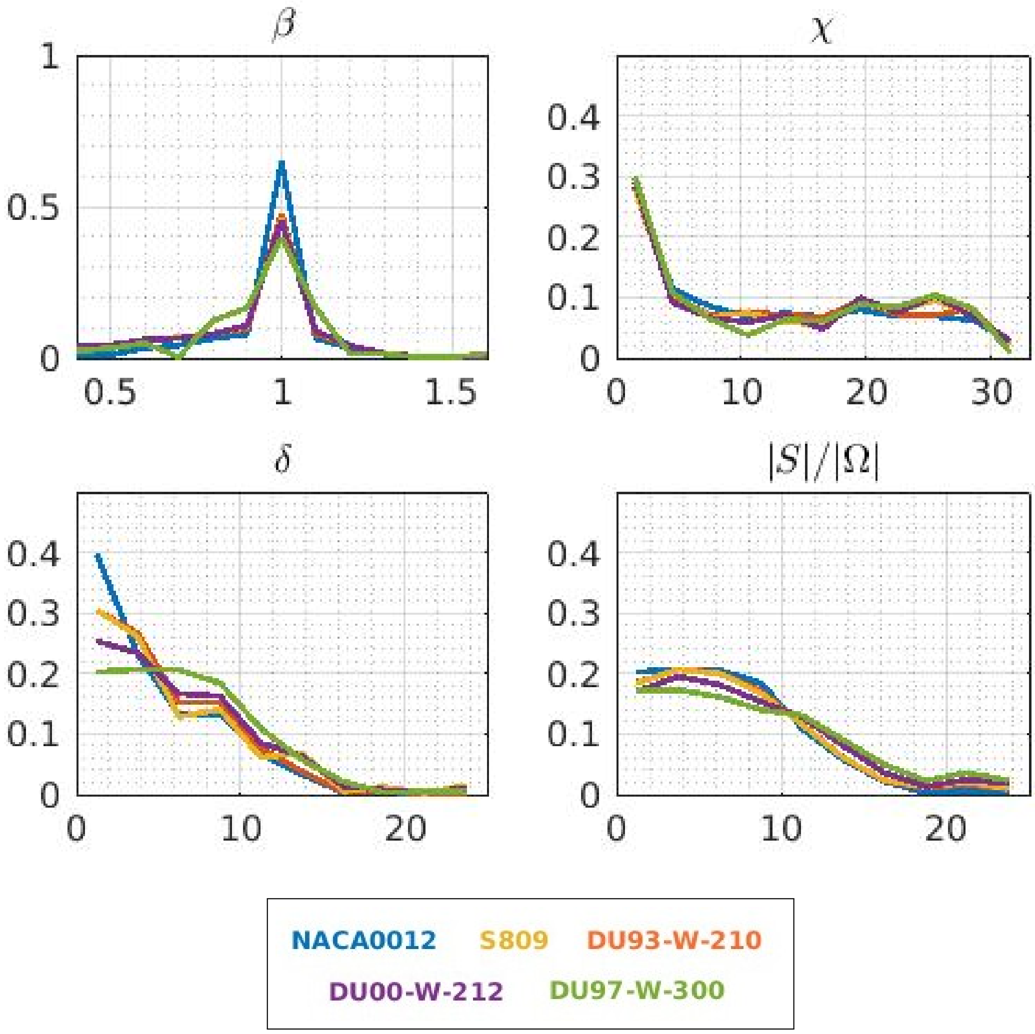

Figure 5.7: Frequency polygons of each flow feature and the correction term for all airfoils

The distributions of these flow features are compared for each airfoil. Figure 5.7 shows the normalized frequency histograms of the combined data of each flow feature and the correction parameter for each airfoil, at all tested angles of attack. The data from the far-field (far away from the airfoil) is removed before the normalization of frequencies, since each of these quantities do not exhibit any variation far from the airfoil. It can be seen that all the flow features approximately collapse into a similar distribution, indicating representative nature of the data obtained from each of these airfoils.

Figure 5.8 shows the obtained features and the correction field for DU-93-W-210 airfoil at $\alpha = 10°$.

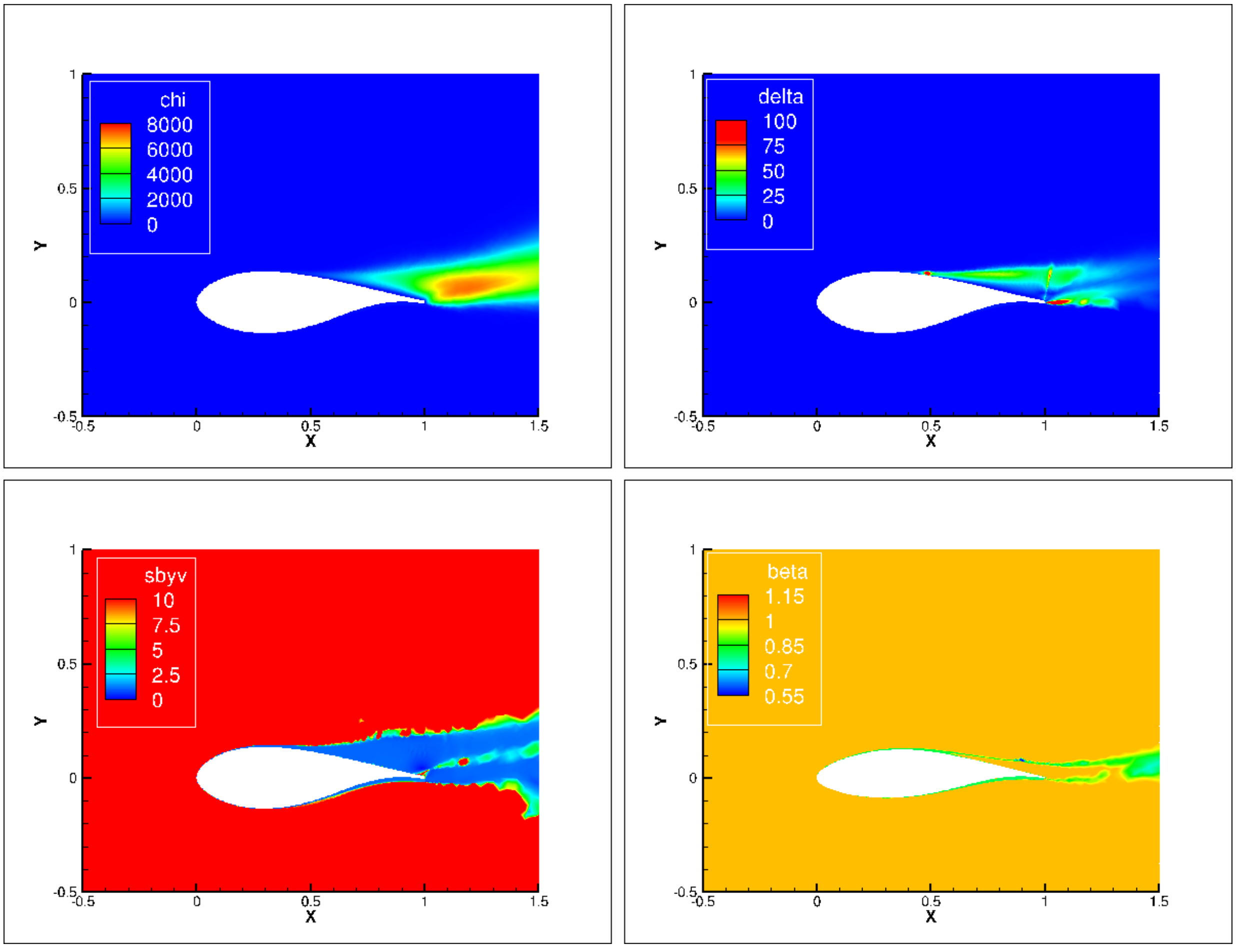

Figure 5.8: Contours of the flow features and β field for DU93-W-210 airfoil at $\alpha = 10°$

It can be seen that all the features are activated in the boundary layer region and in the separated flow region. Contours of the δ parameter rise near the separation point, and drops to lower values in the recirculation bubble. Strain to vorticity ratio is extremely low in the wake region indicating free shear and region of high vorticity.

5.3.3.2: Learning the Functional Correction

Each airfoil data is considered as a separate dataset, and a neural network is trained on each of these datasets. An ensemble dataset is generated using the mean prediction of each of these neural networks, and a new neural network is trained on this dataset. Figure 5.9 shows the schematic of the training of this ensemble neural network.

Ensemble neural networks are a robust way of training regression networks by avoiding overfitting of the data. In conventional bagging type ensemble methods [70], the final prediction is just a weighted average of the component networks. However, since the computation of β occurs at each mesh point for each iteration, the computational cost of bagging ensemble becomes high. Instead, the bagged ensemble is replaced by training a new neural network, with training data generated from the bagged ensemble predictions. The neural network has 3 hidden layers (allowing for a greater level of non-linearity) with 20 neurons in each layer, and uses a hyperbolic tangent activation function. While the output correction field is naturally distributed around $\beta = 1$, a simple affine shifting ensures the appropriate scaling of the terms.

Figure 5.10 shows the training performance of the individual component neural networks on the respective airfoil datasets. It can be seen that the machine learnt correction parameter has a larger spread around the “true” β values obtained from field inversion. This is because the functional variation can not be completely explained using the chosen flow features, indicating the importance of the choice of appropriate set flow features. However, the chosen set of features are the major influencing flow parameters to identify and quantify the onset of flow separation and stall phenomena, and the performance of the machine learning step ensures the learning of the appropriate influence of these variables on the correction term.

The retrained neural network is augmented with the baseline SA model to generate a modified turbulence model. Figure 5.11 shows the comparison of the β field predicted for DU93-W-210 airfoil at $\alpha = 10^\circ$ using the augmented model to the FI predicted optimal β field. While the predicted fields do not match completely due to the loss of information in training, the majority of the correction field is predicted near the airfoil surface, including the areas of adverse pressure gradient and the possible location of flow separation.

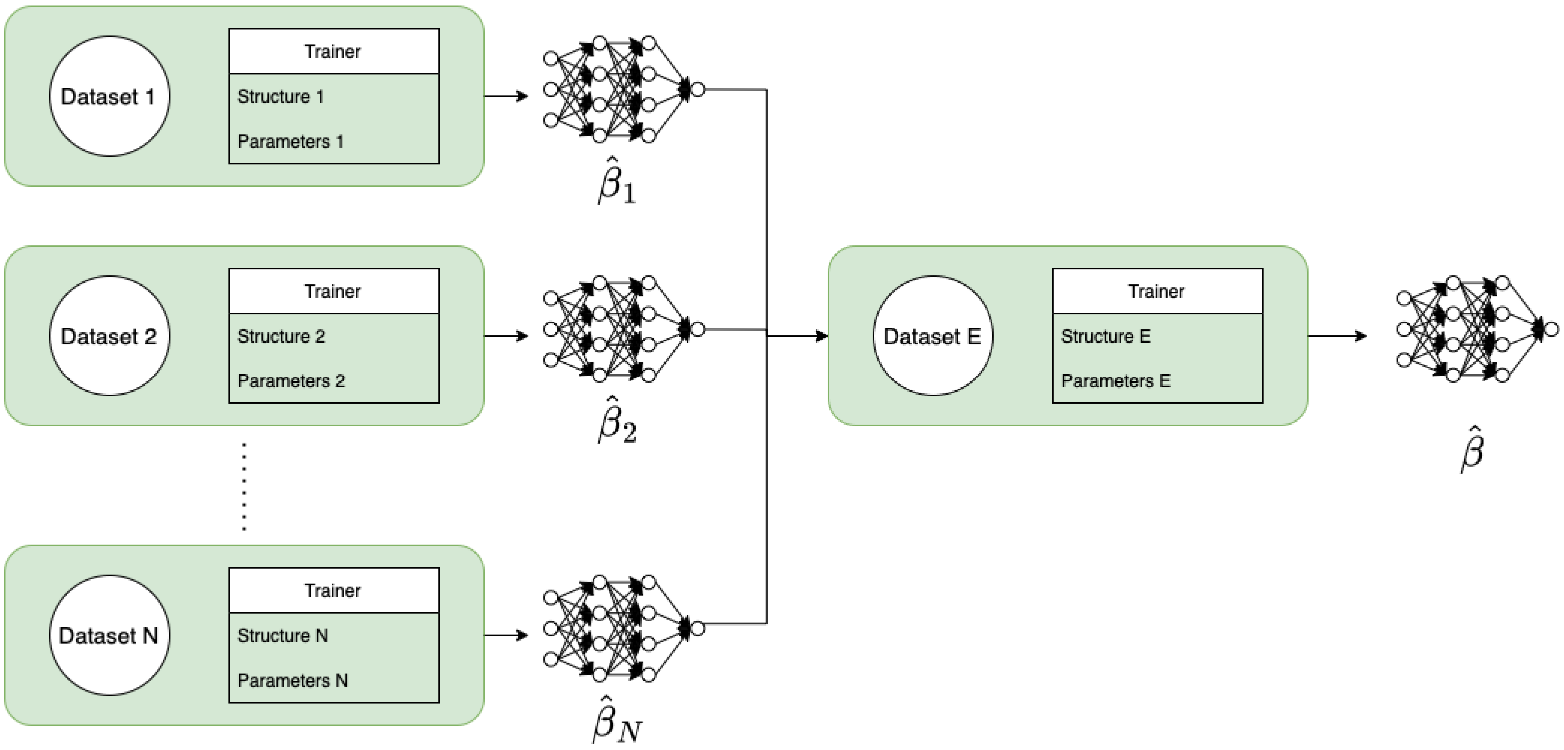

Figure 5.9: Schematic of the Ensemble Neural Network training

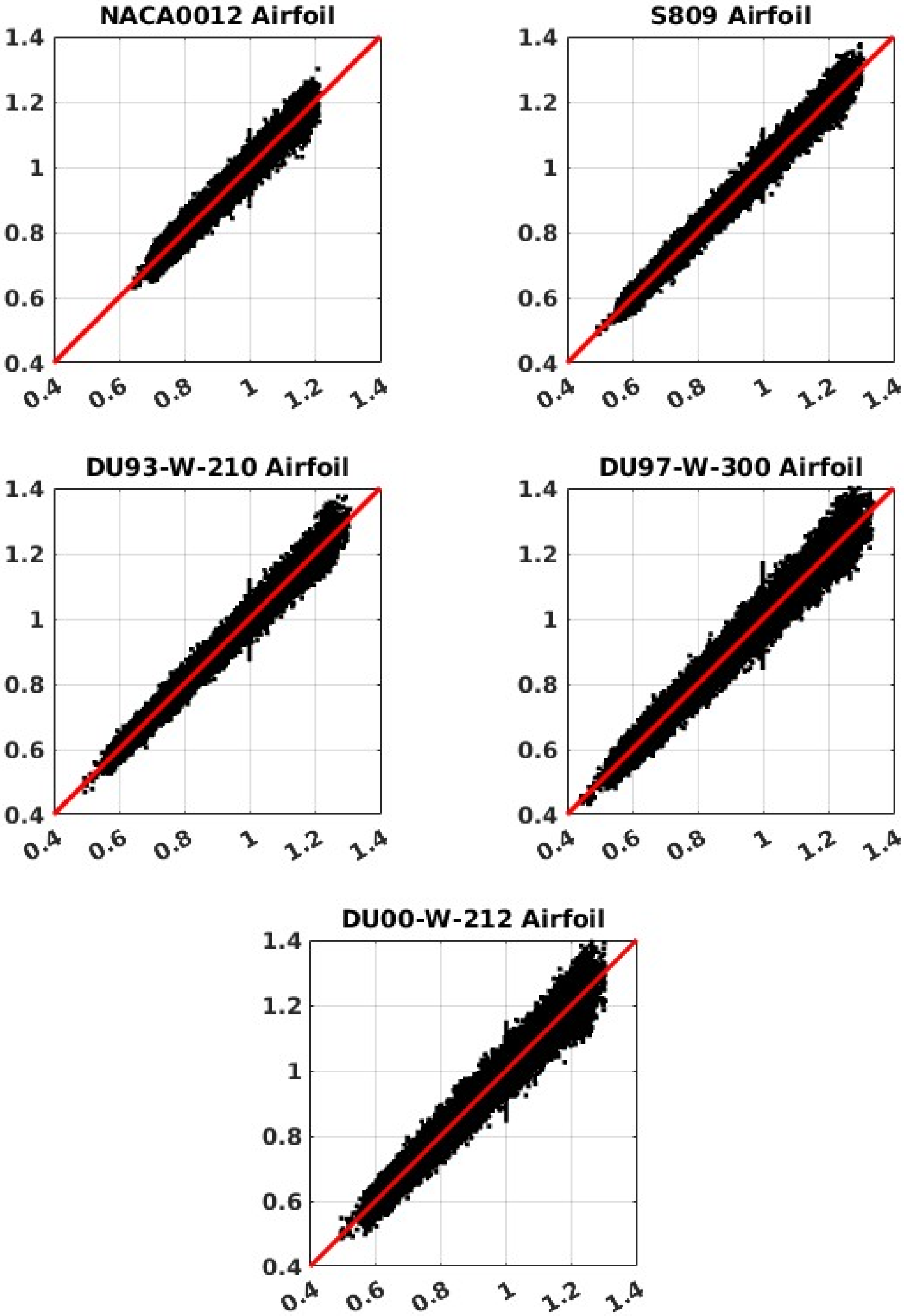

Figure 5.10: Comparison of the predicted β using the individual neural networks vs β obtained using field inversion, for each airfoil

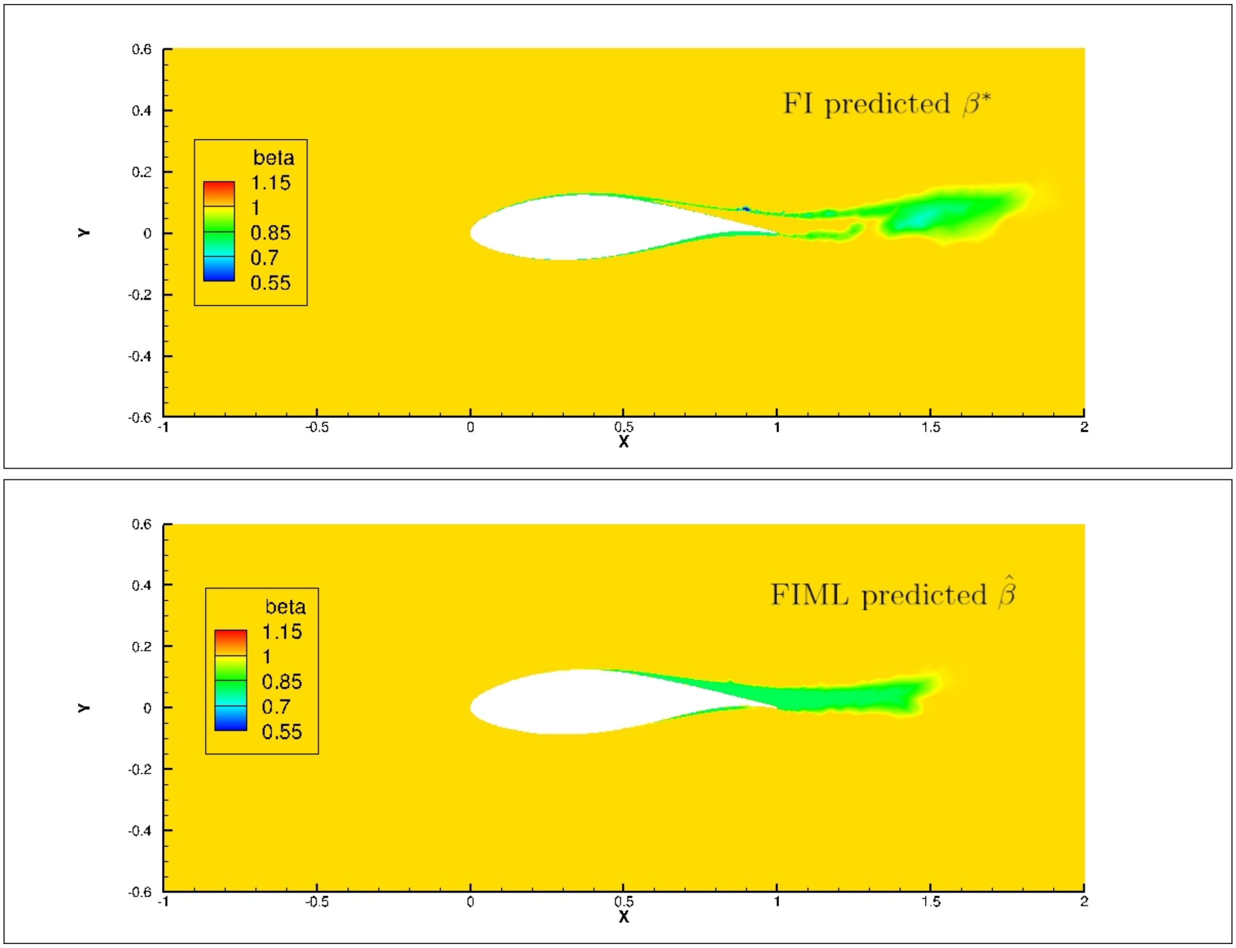

Figure 5.11: Comparison of FIML prediction (bottom) to FI estimates (top) for DU93-W-210 airfoil at $\alpha = 10°$

5.4: Improvements in Airfoil Predictions

5.4.1: Numerical considerations for the Augmented model

The modified turbulence model (Equation 5.9) uses flow features to compute the correction term to the turbulence model. However, in the initial flow iterations, the model may compute unrealistic correction terms due to the unsettled flow variables. Hence, an initial offset of 500 flow iterations is applied to the model, until which the model behaves smilar to the baseline SA model.

Further stability towards sudden changes in β is introduced by using a relaxation iteration where the computed β in the model is given by Equation 5.12. This is similar to a time delayed evolution of the β parameter, and in the steady limit, doesn't alter the flow solution. A nominal value of $\lambda = 0.5$ is used in HAM2D.

$$\beta^{(n+1)}(\vec{x}, \eta) = \lambda \beta^{(n)}(\vec{x}) + (1 - \lambda)\hat{\beta}(\eta) \tag{5.12}$$

5.4.2: Improvements in the lift coefficient predictions for airfoils

The new turbulence model is tested on the airfoils listed in Table 5.1. Each of the airfoils in the training database all show a consistent improvement in the predictions of the lift coefficient compared to the baseline SA model. However, the maximum lift coefficient and the stall angle of attack are still overpredicted by the augmented turbulence model. The FIML improvements are less than what is predicted by the field inversion alone, due to the loss of information in the machine learning step.

Table 5.2 shows the comparison of the predicted stall angles and stall lift coefficients for the tested airfoils. Figures 5.12, 5.13 show the comparison of the predicted lift curves using each of the turbulence models with the experiments.

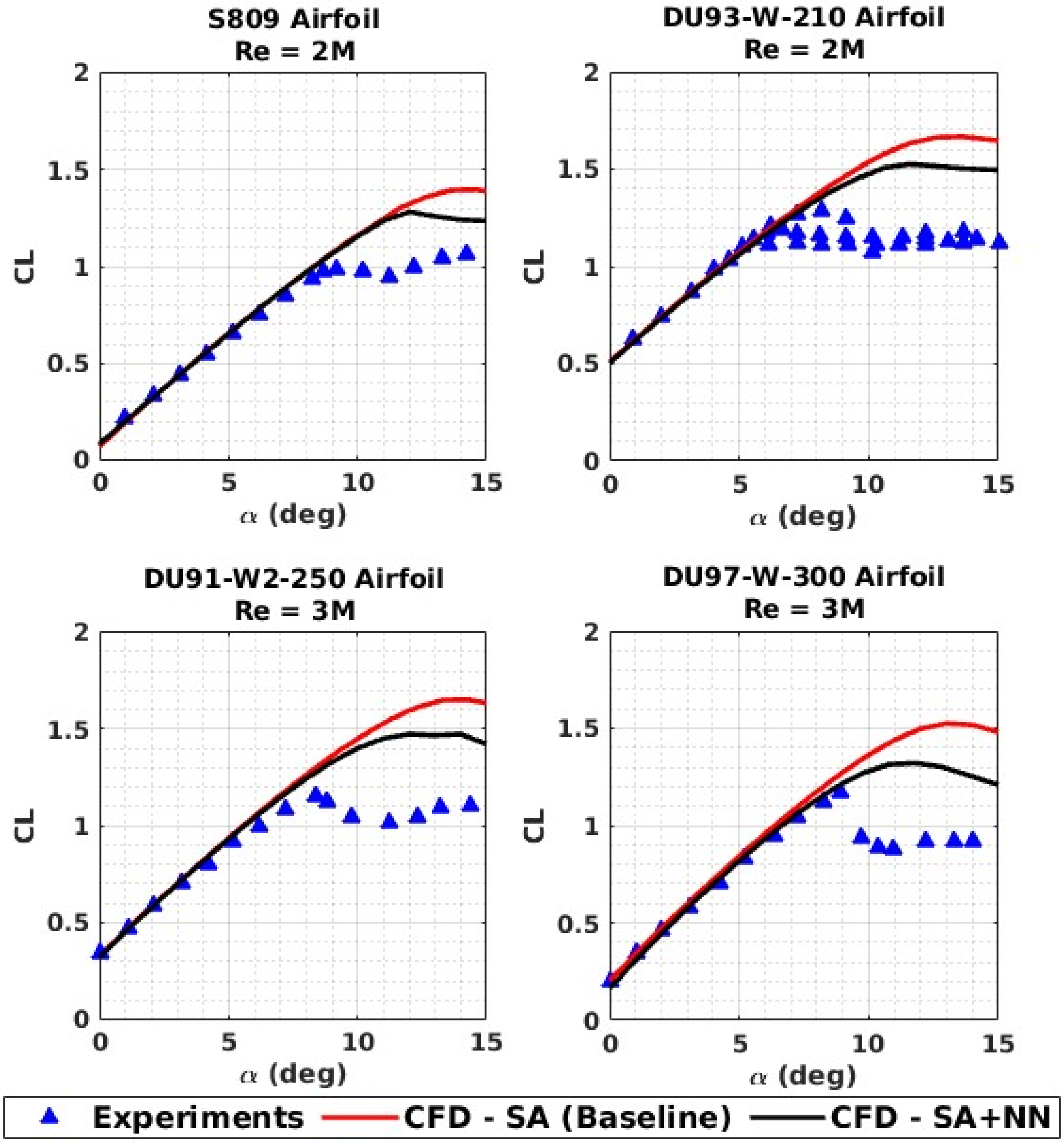

Figure 5.12: Comparison of experimental lift coefficient with improved CFD predictions for different wind turbine airfoils

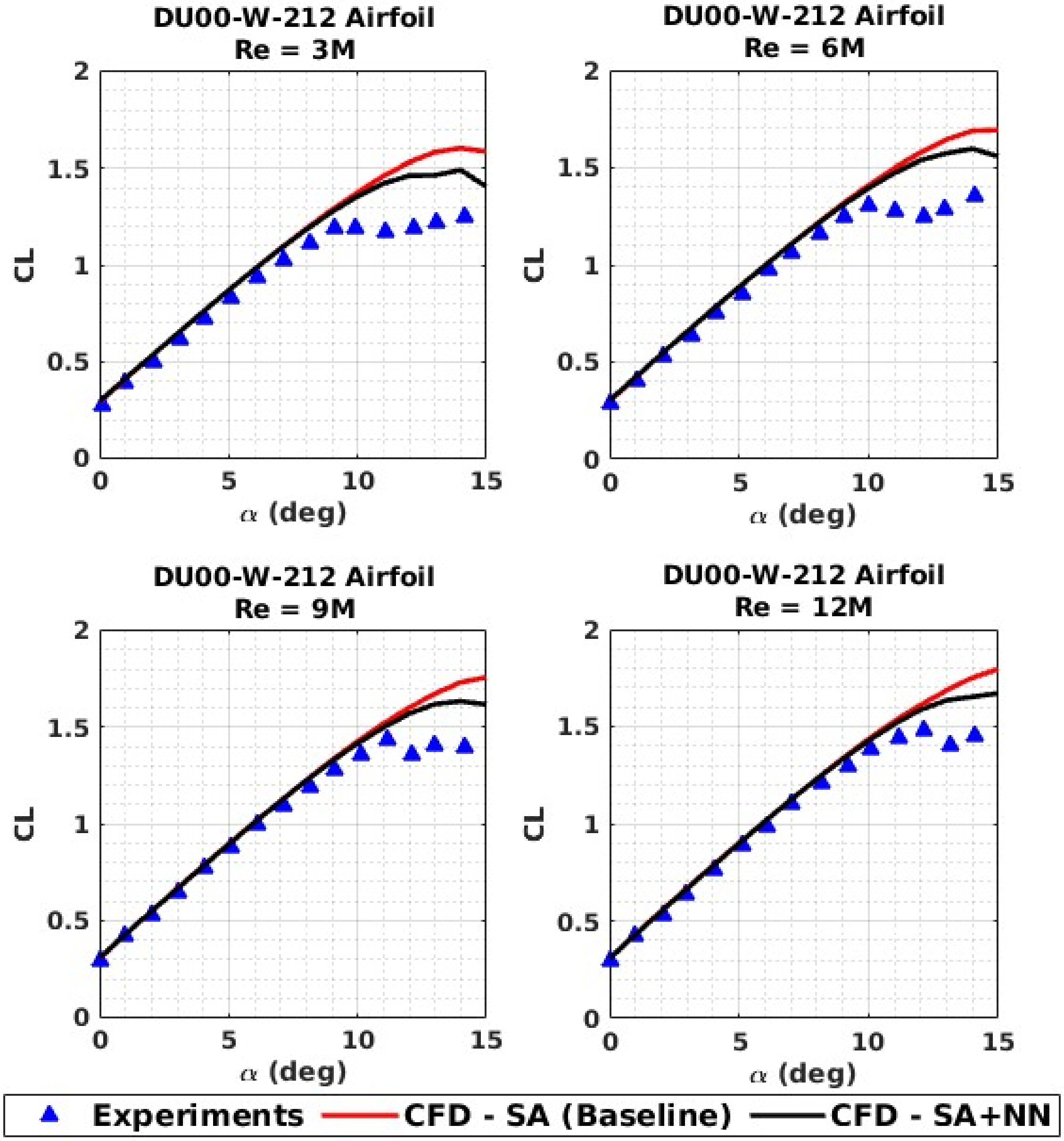

Figure 5.13: Comparison of experimental lift coefficient with improved CFD predictions forDU00-W-212 airfoil at different Reynolds numbers

Table 5.2: Improvements in stall angle of attack and lift coefficient for each airfoil using the augmented turbulence model

Airfoil	Re	Stall Angle			Stall Lift Coefficient		
		Exp.	SA	FIML	Exp.	SA	FIML
NACA0012	6M	17.1°	18°	17°	1.61	1.69	1.58
S809	2M	9.5°	14°	12°	1.07	1.41	1.28
DU93-W-210	2M	8.2°	14°	11°	1.30	1.67	1.52
DU91-W2-250	3M	8.5°	15°	11°	1.17	1.65	1.47
DU97-W-300	3M	9.1°	18°	11°	1.19	1.52	1.31
DU00-W-212	3M	9.5°	14°	12°	1.21	1.61	1.45
DU00-W-212	6M	10.0°	15°	14°	1.30	1.70	1.59
DU00-W-212	9M	11.1°	15°	14°	1.45	1.76	1.62
DU00-W-212	12M	12.1°	16°	14°	1.49	1.80	1.66

The degree of improvement (DOI) of the stall angle and stall lift coefficient are defined as shown in Equation 5.13, where f represents the quantity of interest.

$$\mathrm{DOI}_f = 1 - \frac{|f_{\mathrm{FIML}} - f_{\mathrm{exp}}|}{|f_{\mathrm{SA}} - f_{\mathrm{exp}}|} \tag{5.13}$$

The augmented turbulence model shows improvements of more than 35% in the estimation of stall angle and more than 40% in the stall lift coefficient. In the wind turbine family of airfoils, these improvements are maximum for the 30% thick DU97-W-300 airfoil reaching upto 75% improvement in the stall angle and 60% in the stall lift coefficient.

Furthermore, the accuracy of predictions in the linear lift region are unaltered, indicating that the augmented correction functional is both model consistent and physics consistent.

5.4.3: Improvements in drag coefficient and lift-to-drag ratio

Besides the improvements in lift coefficient, predictions of the rise in drag coefficient are also improved with improvements in resolving flow separation phenomena. The rise in drag results in reduction in lift-to-drag ratio, which is an important factor for the design and operation of wind turbine airfoils.

Figure 5.14 shows the lift-to-drag ratio predictions for DU00-W-212 airfoil using each of the turbulence models and compares with the experimental data. The results show a consistent improvement of the angle of maximum lift-to-drag ratio as well as the magnitude of the maximum lift-to-drag ratio. On an average, the maximum lift-to-drag ratio is improved by 40% across the range of Reynolds numbers, with higher improvements at higher Reynolds numbers.

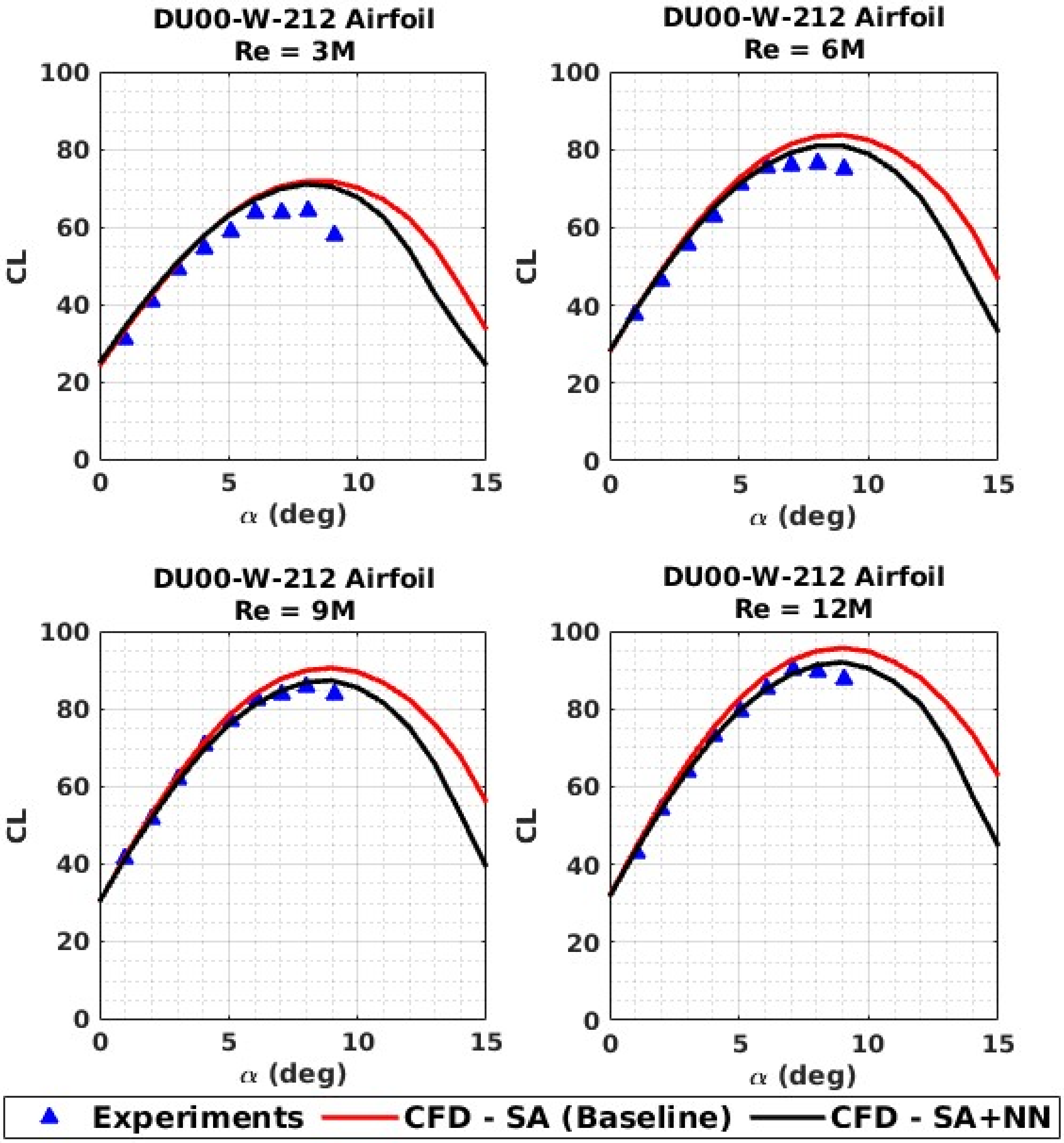

Figure 5.14: Comparison of experimental lift-to-drag ratio with improved CFD predictions for DU00-W-212 airfoil at different Reynolds numbers

5.4.4: Improvements in capturing the flow separation

The flow solutions corresponding to the separated flow show improved predictions of the flow separation point. Figure 5.15 shows the comparison of the surface pressure coefficient distribution predicted by each of the turbulence models with the experimental measurements, for S809 airfoil at $\alpha = 14°$. It can be clearly seen that the separation point predicted by the augmented turbulence model moves upstream (showing larger separation) compared to the prediction by SA model. Also, the predicted surface pressure coefficient using the augmented turbulence model are closer to the experimental measurements compared to the baseline.

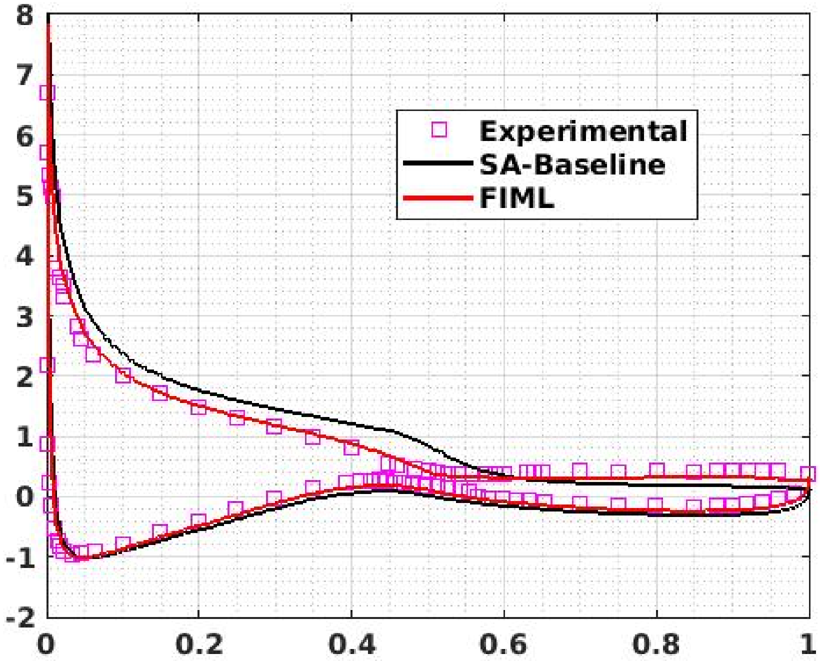

Figure 5.15: Comparison of the surface pressure coefficient predicted by the turbulence models with the experimental measurements, for S809 airfoil at $\alpha = 14°$

Figure 5.16 show the predicted streamlines and the separation bubble on the suction surface of S809 airfoil at $\alpha = 14°$. Consistent with the surface pressure measurements, the streamlines show a larger separation region predicted by the augmented model compared to SA baseline.

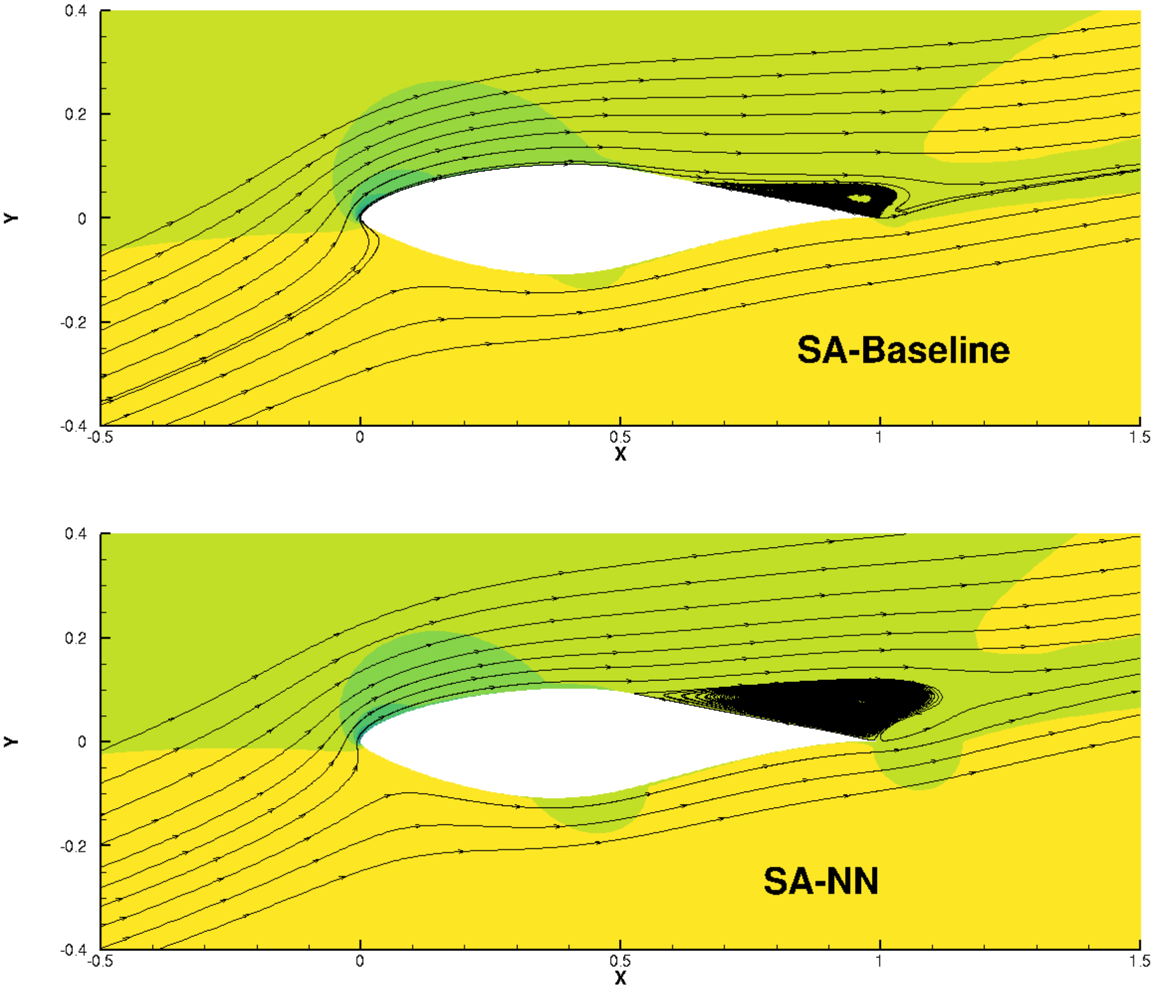

Figure 5.16: Comparison of streamlines predicted using SA baseline (top) and SA-FIML (bottom) showing flow separation region for S809 airfoil at $\alpha = 14^\circ$

It must be noted that these improvements are based on training using only the integrated lift coefficient. The improvements in other physical quantities indicate that the learnt correction is consistent with the actual physical behavior of fluids in such strong adverse pressure gradient scenarios.

5.5: Summary

Field-Inversion Machine Learning (FIML) methodology is applied to arrive at a functional correction to the SA turbulence model, to achieve a better prediction of flows over wind turbine airfoils in the stall regime. A non-dimensional functional correction is applied to the production term in the SA model, to account for the appropriate changes in the source terms governing the eddy viscosity variable.

Field inversion is performed on a representative set of airfoils ranging from 12% thickness to 30% thickness, for different Reynolds numbers and angles of attack to encompass a wide range of flow scenarios. The inversion process is based on the experimentally measured lift coefficients of the airfoils.

Three non-dimensional flow features corresponding to the SA working variable, the local shear levels in the flow and adverse pressure gradient are chosen to be the appropriate governing features for the functional correction. Ensemble methods are used to train a neural network to fit a functional relation between the identified flow features and the correction parameter and the obtained functional correction is augmented with the SA turbulence model.

The modified turbulence model (SA + NN) shows consistent improvements in predictions of various engineering variables as well as local flow solutions. Some key conclusions are

(1) Field inversion is able to dramatically improve CFD predictions near stall, and provide large amount of training data for machine learning.

(2) All the tested airfoils show similar distributions of the identified flow features

(3) Predictions of the augmented turbulence model show a consistent improvement by more

than 35% in the stall angle and by more than 40% in the stall lift coefficient.

(4) Lift-to-drag ratio predictions show consistent improvements, with an average maximum lift-to-drag ratio improvements of 40%.

(5) Flow separation point is more accurately predicted upstream of the suction surface for airfoils, with improved surface pressure estimates.

Chapter 6: Physics-informed Data-driven Surrogate Model

In this chapter, the developments in all the previous chapters are integrated into a combined physics-informed data-driven surrogate model. This is a meta-learning model, i.e., a learning algorithm is applied on the data generated using another learning algorithm.

6.1: Development of the "Meta-Learning" Aerodynamic Analysis Tool

6.1.1: Data Generation

The airfoil geometry database of wind turbine airfoil perturbations is used to develop the surrogate model in this study. CFD simulations are performed on each of the geometries, using FIML augmented turbulence model. Figure 6.1 shows the spread of the performance quantities for the geometries.

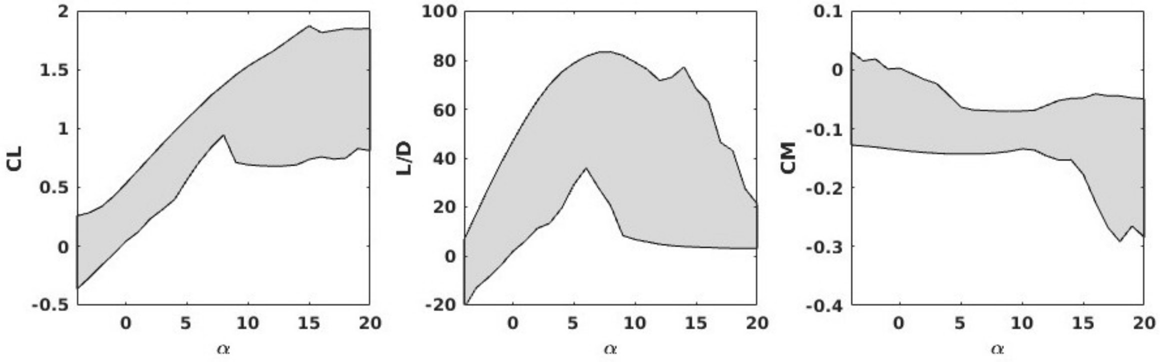

Figure 6.1: Range of the performance parameters in the FIML augmented dataset

6.1.2: Training the Model

An artificial neural network based surrogate model is developed, with the data scaling and orthogonalization steps for both inputs and outputs is implemented. Repeated k-fold cross-validation is performed, with Huber loss function as the training objective. The neural network has two hidden layers and hyperparameter tuning is performed locally around $H_1 = 25$ and $H_2 = 20$. The neural network has nearly same composite statistic score for all the possible hyperparameters satisfying $22 \leq H_1 \leq 28$ and $16 \leq H_2 \leq 24$.

The final surrogate model has two hidden layers with $H_1 = 25$ and $H_2 = 20$, with hyperbolic tangent activation function in each layer. Lasso regularization is used with a penalty parameter of $\lambda = 0.1$.

6.1.3: Performance of the Model

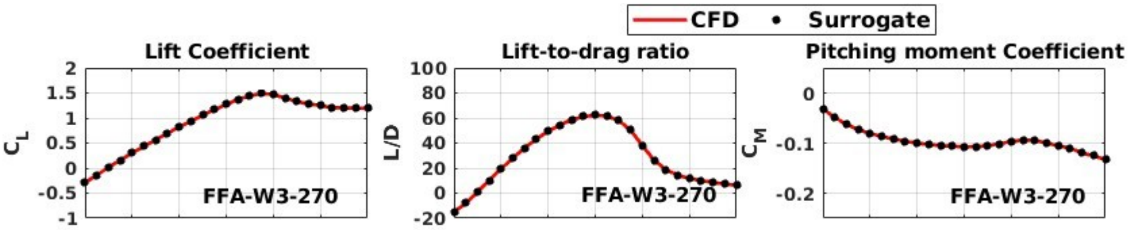

Figure 6.2: Validation of the surrogate model for FFA-W3-270 airfoil

Figure 6.2 shows the validation of the surrogate model for FFA-W3-270 airfoil. The associated L_2 error for the total dataset for the performance quantities are 0.035, 0.45 and 0.0029 for lift coefficient, lift-to-drag ratio and pitching moment coefficient respectively.

6.2: Application to Airfoil Design Problems

In this section, we demonstrate the usage of the surrogate model for airfoil design problems. Genetic algorithms are used as the optimization tool.

6.2.1: Genetic Algorithms

The sequence of steps involved in the optimization process using Genetic Algorithms (GA) is shown in Figure 6.3. These algorithms are inspired by the process of natural selection, where the designs are evaluated for a fitness criteria or the objective function and only the better performing designs survive through generations. To explore the design space, crossover and mutation operations are used to propagate the design population towards best performing designs in the design space.

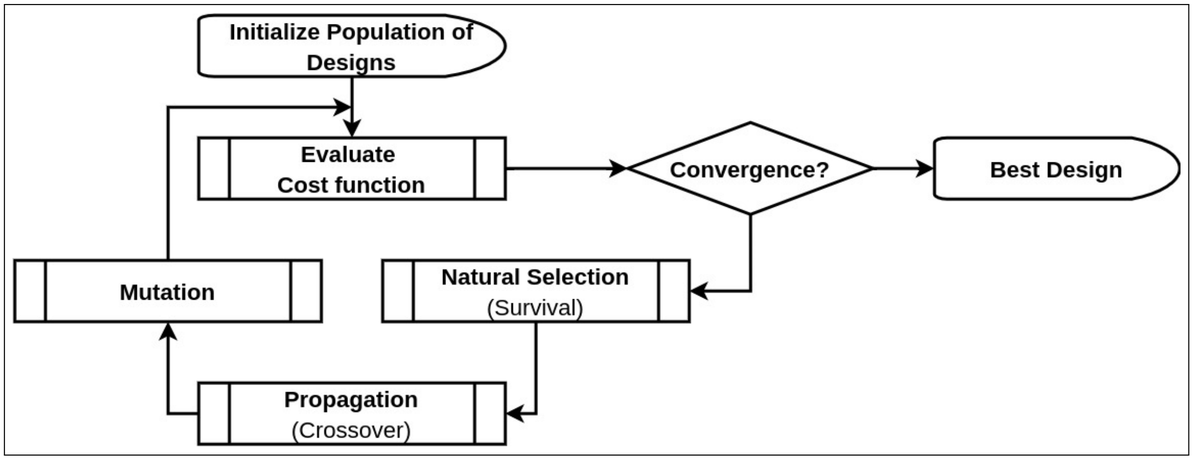

Figure 6.3: Genetic algorithms methodology and sequence of steps

A least squares objective function described in Equation 6.1 is used to arrive at a best performing geometry using GA. Three constants (λ_i) are used to tune the relative importance of each performance parameter and to remove any bias towards the quantities with large magnitudes, as the coefficients of the L_2 norm of the difference in each polar to the corresponding target performance. For each design problem tested in this work, GA is used to arrive at an optimal airfoil, with an initial randomly distributed design population of size 32, continuous random variables to define crossover of design variables, and a 5% mutation rate with a normally distributed random variable for mutations. The population is propagated through 500 generations and the best performing design is chosen at the end of 500 generations.

$$J_{\mathrm{GA}} = \lambda_1 \left\| c_l(x) - c_{l,\mathrm{target}} \right\|_2^2 + \lambda_2 \left\| \frac{L}{D}(x) - \left(\frac{L}{D}\right)_{\mathrm{Target}} \right\|_2^2 + \lambda_3 \left\| c_m(x) - c_{m,\mathrm{target}} \right\|_2^2 \tag{6.1}$$

Usage of the developed surrogate model architectures enabled a new paradigm for aerodynamic design optimization, where support is provided for a multi-point aerodynamic design. This paradigm enables aerodynamic designs for multiple design points as well as design considerations for the operational envelope.

6.2.2: Design Problem 1: Reconstruction of an airfoil shape

The first design problem is to reconstruct the DU93-W-210 airfoil from its performance curves. This testcase is a validation case similar to the method of manufactured solutions. The CFD predicted lift coefficient, lift-to-drag ratio and pitching moment coefficient are fed into the objective function as targets.

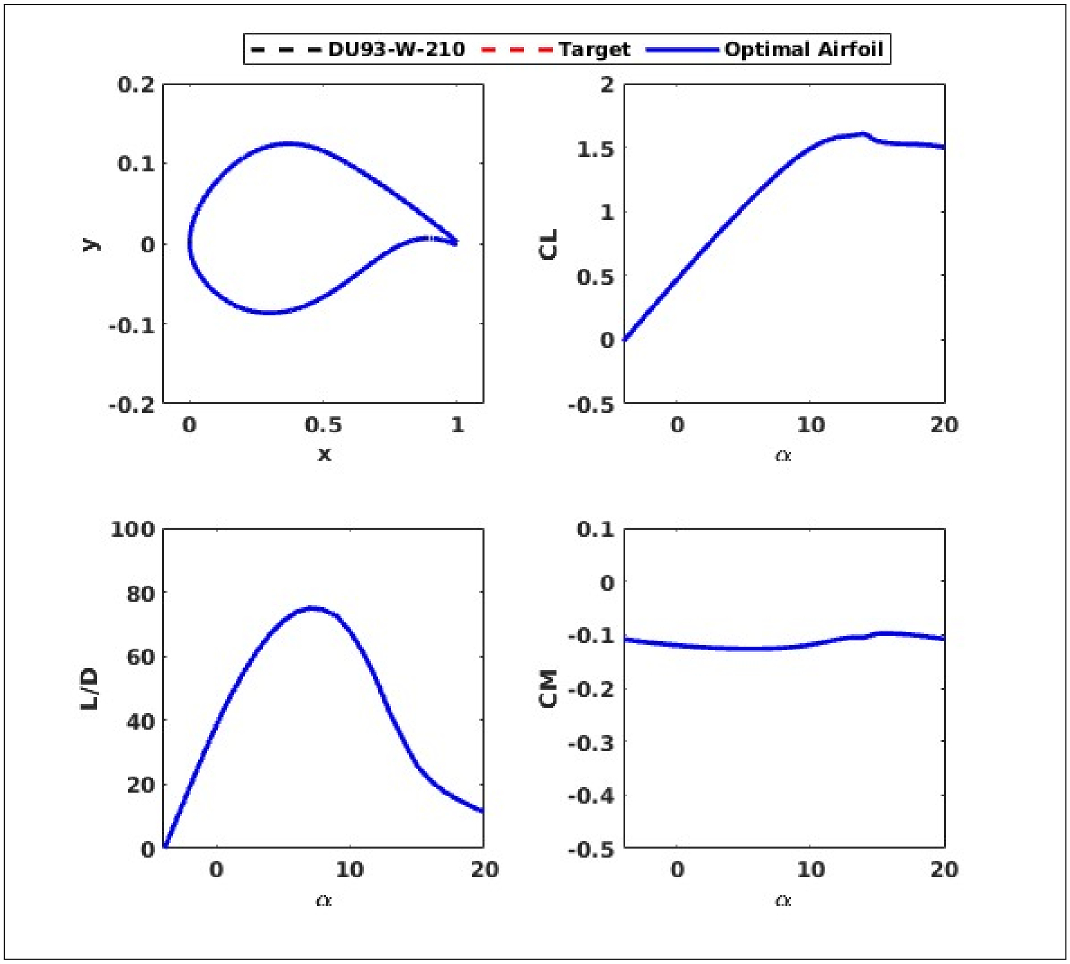

Figure 6.4: Results of Design problem 1: The DU93-W-210 airfoil shape is predicted accurately using optimization

Figure 6.4 shows the results of the optimization problem, with accurate prediction of the airfoil shape and the performance curves. This problem reconfirms the validity of the surrogate model for wind turbine airfoil geometries.

6.2.3: Design Problem 2: Increasing the lift-to-drag ratio

For the second design problem, the objective is to increase the lift-to-drag ratio of the airfoil by 5% throughout the range of angles of attack and maintain the lift and pitching moment profiles same. This essentially is a drag reduction problem. Figure 6.5 shows the target lift-to-drag ratio curve and the predicted airfoil shape and its performance predictions.

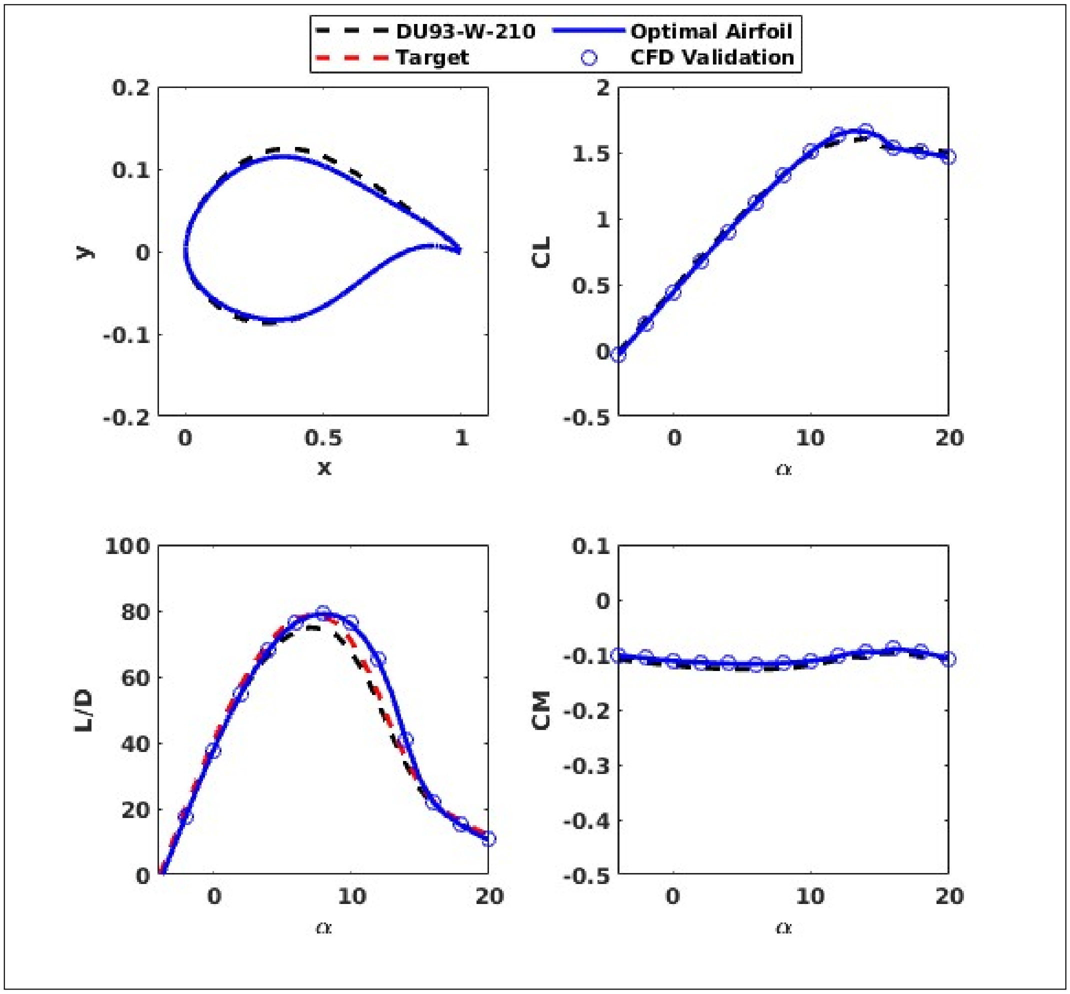

Figure 6.5: Results of Design problem 2: lift-to-drag ratio is improved by reducing the airfoil thickness

In order to increase the lift-to-drag ratio, the airfoil thickness has reduced. While the target was to increase lift-to-drag ratio alone, the stall angle is delayed due to the change in the shape of the airfoil, and thereby, the genetic algorithm arrived at an airfoil which meets the target distribution until maximum lift-to-drag ratio and then drops slower than the target curve.

CFD is performed on the obtained shape, and the predictions of the surrogate model accurately match with the CFD predictions.

6.2.4: Design Problem 3: Delaying the stall

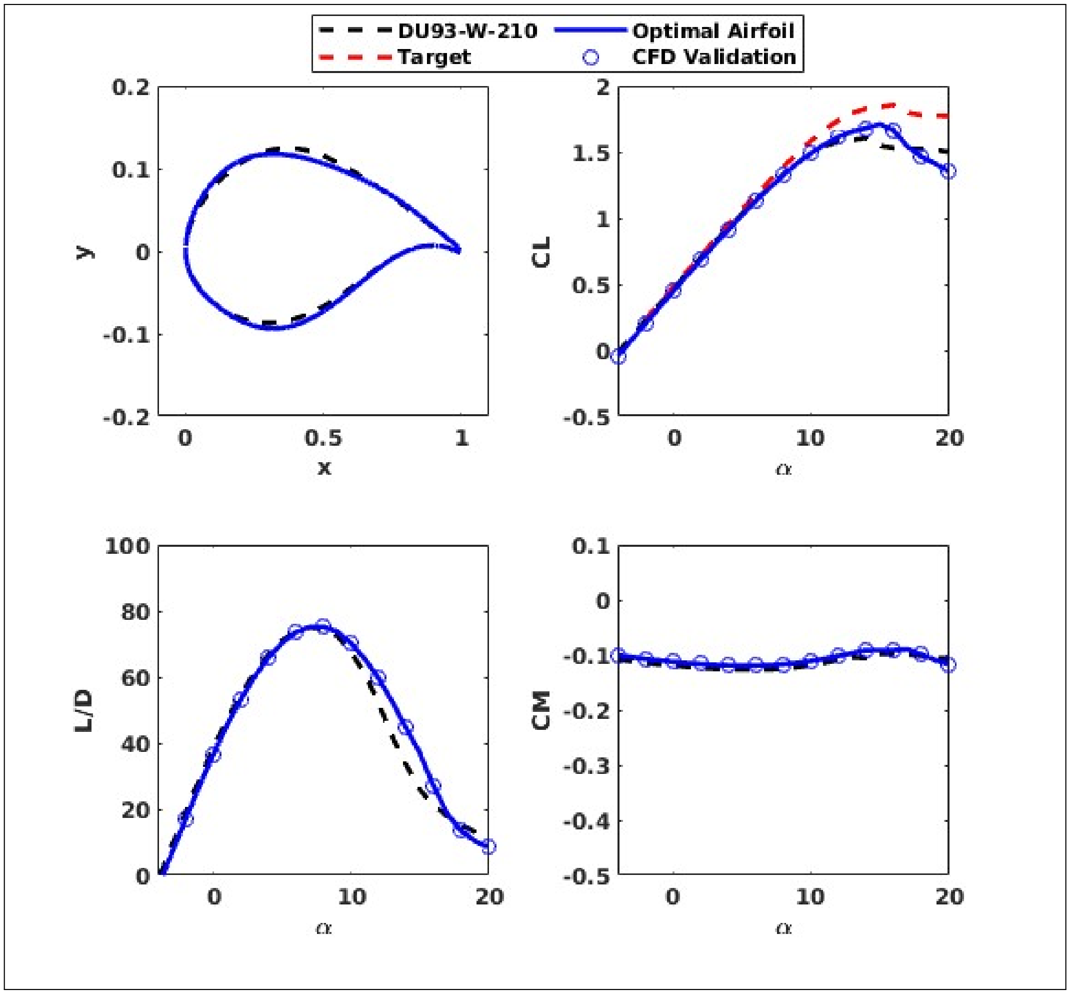

Figure 6.6: Results of Design problem 3: Stall angle is delayed by varying the airfoil suction surface

For the final design problem, the objective is to delay the stall phenomena by 1°. The target curve fed into the objective function is obtained by increasing the linear lift region by 1° and then maintaining same post stall reduction in the lift. Figure 6.6 shows the target lift curve and the predicted airfoil shape and its performance predictions.

In order to delay stall, the airfoil suction surface is modified with relative thickness of the airfoil maintained nearly the same. The stall angle is shifted by 1°, however, the lift curve did not match the target curve. The predicted stall is gradual with the drop of lift occuring at the delayed stall angle. The maximum lift-to-drag ratio remained nearly the same.

CFD is performed on the obtained shape, and the predictions of the surrogate model accurately match with the CFD predictions.

6.3: Summary

A “meta-learning” model is developed by combining the physics-informed modeling and data-driven surrogate modeling techniques. The developed model is used in a design optimization framework using genetic algorithms, on a few design problems. Some key takeaways are

(1) The developed surrogate model has uniform accuracy across a variety of wind turbine airfoils.

(2) The surrogate model is used together with genetic algorithms based optimization framework to perform inverse design of airfoils.

(3) Inverse design to arrive at a prescribed airfoil shape using its performance curves gave accurate results

(4) Inverse design for drag reduction and stall delay gave airfoils that partially reach the target

performance. However, an airfoil that perfectly matches the target is not predicted. This may be because the physics of the flow doesn't allow to attain the prescribed targets.

(5) CFD computations are performed on both the new airfoil designs, and the surrogate model predictions accurately match with the CFD predicted data.

Chapter 7: Closure

Design testing and analysis forms a major bottle-neck in the design process of wind turbine applications. Even with the advancements of high performance computing frameworks, use of high-fidelity CFD for analysis during the design optimization phase is impractical due to the overall computational cost of analysing a large number of designs. Additional overheads get added to the overall cost if the design is performed in a multidisciplinary and/or non-deterministic setting.

Surrogate modeling is an efficient data-driven tool to model the required quantities of interest without actually having to perform the computationally expensive high fidelity CFD simulations. In this thesis, a representative database of wind turbine airfoils is developed using efficient design space exploration strategies. A database of CFD solutions for the nominal operating ranges of angles of attack and Reynolds numbers is generated. A neural network based surrogate model is developed to predict the airfoil aerodynamic performance quantities of lift coefficient, lift-to-drag ratio and pitching moment coefficient for a given airfoil geometry.

Besides efficient design space exploration, efficient parameter space exploration strategies using data puncturing are explored to exploit data synergy and redundancy distributions in the data. The effects of reduction in the data on the accuracy of the surrogate model is investigated for different styles of data puncturing.

The effects of uncertainty in airfoil geometries on the performance quantities is investigated using a Monte-Carlo based uncertainty qunatification framework built on the surrogate model. A quasi Monte-Carlo based global sensitivity analysis framework is developed to rank the design variables and identify the most influencing design variables.

Additionally, physics-informed modeling technique of field-inversion machine learning is used to improve the turbulence closure model to predict the stall characteristics of the airfoils more accurately. A correction field is obtained using field inversion for multiple airfoils using the experimental lift coefficients. Appropriate flow features are identified to learn a functional correction for the SA turbulence model to improve the stall predictions on wind turbine airfoils.

The improved turbulence model is used on the generated database of wind turbine airfoils and a "meta-learning" surrogate model is developed. This surrogate model is a computationally inexpensive aerodynamic performance estimator of wind turbine airfoils, with predictions of stall closer to the experiments. This surrogate model is used along with an evolutionary optimization framework to perform wind turbine airfoil design optimization problems.

7.1: Key Observations and Conclusions

7.1.1: Data-driven Surrogate Modeling

(1) Using orthogonal or near-orthogonal shape functions like Chebyshev polynomials results in a well-behaved airfoil representation.

(2) Orthogonalization transformation of inputs and outputs using POD is more efficient for training a neural network and the accuracy of the predictive model is very high even with

shallow neural networks. This is mainly because the complex physical behavior is captured as features in the high eigen value POD modes.

(3) Huber loss is a more efficient training objective function which has both the features of MSE (mean squared error) and MAE (mean absolute error).

(4) Repeated k-fold cross-validation ensures a more representative measure of the performance of the machine learning model for smaller datasets.

(5) Trade-off between the bias and variance errors is efficiently computed using k-fold cross-validation, and is hence recommended for hyperparameter tuning.

(6) The developed surrogate model is uniformly accurate over a large range of wind turbine airfoil geometries, with an L_2 error estimate of 0.03 in lift coefficient, 0.4 in lift-to-drag ratio and 0.003 in pitching moment coefficient.

(7) Knowledge of synergy and redundancy can help to efficiently design the parameter space for training data generation.

(8) Naive data puncturing is an inefficient parameter space design technique since the synergy is lost in the process of reducing redundancy.

(9) Uniform data puncturing can efficiently reduce the required training data by 50% without loss in accuracy of the surrogate model.

(10) Uniform data puncturing causes an initial decrease in redundancy followed by a delayed decrease in synergy. As a result, the accuracy of surrogate model is plateaued until the drop in synergy kicks in.

(11) Selective data puncturing is a more efficient strategy to improve the proportion of synergy to redundancy in the dataset.

(12) 5% deviation in geometric parameters contributed to about 3% deviation of the performance quantities near the operating angle of attack for the wind turbine airfoil.

(13) Near stall angle, the uncertainty of the performance parameters of lift coefficient and lift-to-drag ratio increases to more than 10% for 5% uncertainty in geometric parameters. The pitching moment uncertainty is however uniform across all angles of attack.

(14) Global sensitivity analysis shows the top surface of the airfoil surface as the most influential design parameter for high angles of attack beyond stall.

7.1.2: Physics-informed Modeling

(1) The various airfoils all show a similar distribution of the key flow features identified as the driving features of the turbulence correction as well as the field inversion predicted correction parameter, when trained with a representative set of angles of attack.

(2) An ensemble averaging technique is a robust training strategy to train the predictive models.

(3) Predictions of the FIML augmented turbulence model (SA+NN) achieved a consistent improvement in the predictions of the stall angle of attack by more than 35% and the stall lift coefficient by more than 40%.

(4) Lift-to-drag ratio predictions show consistent improvement with an average maximum lift-to-drag ratio improvement of 40%.

(5) Airfoil surface pressure predictions are also significantly improved, with the separation point shifted upstream on the suction surface of the airfoil.

7.1.3: Meta learning model

(1) The meta-learning model showed a consistent uniform accuracy over the entire range of wind turbine airfoil database.

(2) Using this model, the inverse design problem to reconstruct an airfoil from its performance curves was accurately predicted.

(3) For the design problems to achieve higher lift-to-drag ratio and delayed stall, the results were partially matching with the target performance curves, and post stall comparison shows the maximum difference.

7.2: Recommendations for Future Work

The frameworks developed in this thesis consist of efficient design exploration strategies, design of experiments and robust training approaches. However, there is a significant scope of improvement in a few aspects of the methodology and a few potential extensions.

(1) This work aimed at developing relatively simple machine learning models and achieved significantly higher accuracies compared to the state-of-the art. This is mainly because of the appropriate processing of the data before training. Therefore, a primary suggestion to extend the work on surrogates will be to retain the simplicity of the machine learning architectures and focus more on the data generation and processing strategies. Usage of

modern design of experiments with Bayesian formalism of choosing the training data can further reduce the amount of training data required for developing the surrogate models, and should be a potential domain to investigate.

(2) Data puncturing is a new concept introduced in this work. Selective puncturing has the highest potential to reduce the training costs. However, this work tested a simple variation of the selective puncturing strategy. A possible extension of this work is to develop metrics that can quantitatively identify the distributions of synergy and redundancy in a dataset, and then develop a selective puncturing strategy using the developed metrics on the lines of adaptive strategies for design of experiments.

(3) The physics-informed modeling strategy employed a simple correction to the production term. A more rigid functional correction that is guaranteed to obey the generalized law of the wall may be formulated and used with the FIML methodology. Adverse Pressure Gradient corrections to SA model by Shivaji [69] can be a starting point of developing such functionals and training for a more consistent correction.

(4) Modern machine learning architectures like the Tandem Neural Networks (T-NN) [71] can be used for efficient inverse design applications. Development of such frameworks with efficient implementations of multidisciplinary and multi-objective constraints has a large potential to serve for various design engineering applications. The current work on data exploration and surrogate modeling can be extended to develop similar inverse design architectures with more efficiency.

Appendix A: Comparison of Chebyshev and Bernstein polynomials

The Class function - Shape function Transformation (CST) is a geometry parametrization approach for various aerospace geometries. Equation A.1 shows the CST parametrization of the airfoil surface coordinate (ζ) using class function (C) and shape function (S) along the chord-wise coordinate (ψ). The class function C encodes the typical airfoil geometric characteristics of rounded leading edge and sharp trailing edge. Finite trailing edge thickness is represented using the second term involving the trailing edge thickness ζ_T. The actual shape of the airfoil surface is governed by the shape function S.

$$\zeta(\psi) = C(\psi)S(\psi) + \psi\zeta_T \tag{A.1}$$

The original CST methodology [50] uses the set of Bernstein polynomials as the basis of the vector space of shape functions. The CST methodology accurately represents the airfoil geometries within the accepted level of tolerance limits [50], indicating that the polynomial shape functions are appropriate to represent the airfoil geometries. The choice of Bernstein polynomials was made primarily for their simple functional form and their physical insight in terms of the local contributions of each basis polynomial to the geometries. For instance, the lower Bernstein polynomial numbers influence the shape near the leading edge and the higher polynomial numbers influence the shape near the trailing edge. The physical intuitiveness of these polynomials make the design decisions easily interpretable.

However, due to the non-orthogonal nature of the Bernstein polynomials, the representation of airfoil shape using these basis functions becomes ill-posed. That is, two physically similar/close geometric shapes are represented using a largely different set of Bernstein coefficients. This is illustrated using an example airfoil surface geometry in Figure A.1. Unit airfoil in the figure corresponds to the airfoil with a shape function of unity, i.e., all the Bernstein coefficients are 1. This airfoil is compared with a perturbed airfoil with largely different Bernstein coefficients.

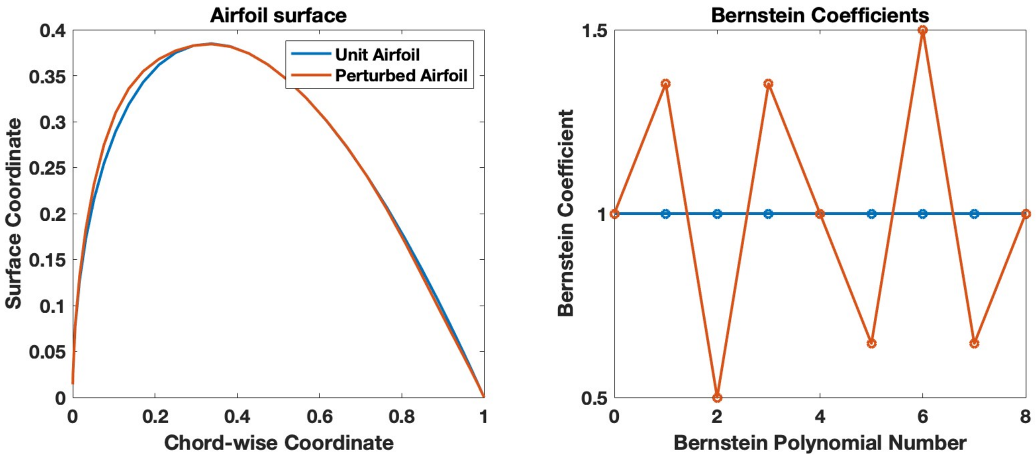

Figure A.1: Ill-posedness of the Bernstein polynomial based CST. Left: Airfoil top surface for unit airfoil and a perturbed airfoil, Right: Corrsponding Bernstein coefficients

Observe that while the airfoil geometries are similar with a small L_2 variation in shape, the corresponding Bernstein coefficients have a significantly large difference. This ill-posedness of the representation results in the Bernstein coefficients being bad inputs for surrogate modeling.

To overcome this issue, the basis functions chosen must represent an orthogonal set of polynomials. The orthogonality of the basis functions will ensure the validity of the norm across physical and CST space. With the airfoil class function, the Jacobi polynomials form an orthogonal set of polynomials corresponding to the standard inner product definitions of square integrable functions. However, these polynomials alter with changing class function, violating the universal parametrization properties of CST. We propose the use of Chebyshev polynomials, which are near orthogonal (i.e., the inner products two different basis polynomials with respect to the airfoil class function are close to zero). Figure A.2 shows the corresponding Chebyshev polynomial based CST coefficients for the example airfoils considered above.

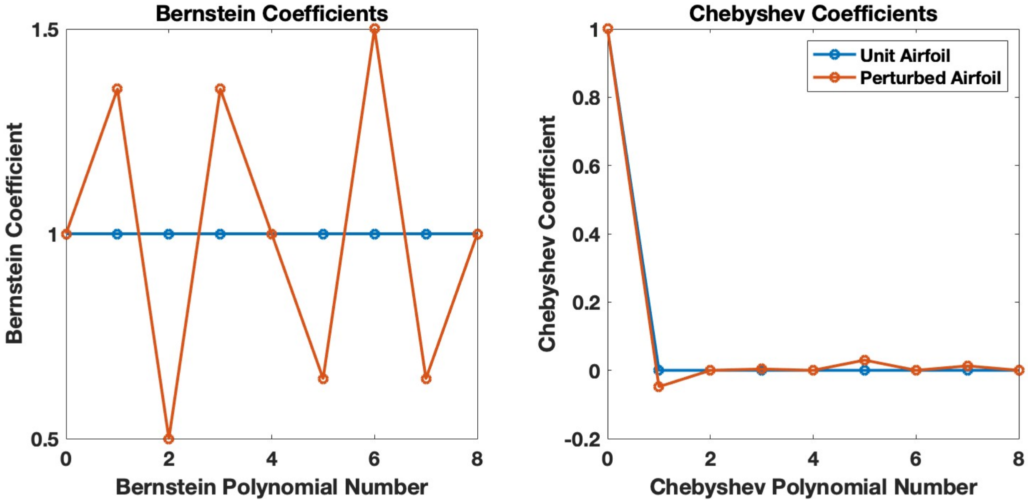

Figure A.2: Comparison of Bernstein (left) and Chebyshev polynomial (right) representation for the unit and perturbed airfoil

Clearly, the magnitude of the variation from unit airfoil to the perturbed airfoil is low in the Chebyshev representation, in alignment with small differences in the geometries. The Chebyshev representation is hence well-posed and will form a better geometric representation alternative to be combined with CST. Furthermore, the representation accuracy with the modified CST is maintained exactly the same, since both the sets of polynomials span the same vector space of geometries.

Appendix B: Neural Network Data

The surrogate models developed in this study are made available for public use and hosted on GitHub at the URL: `https://github.com/koushik-marepally/WT_Airfoil_Surrogates`.

Bibliography

[1] Murjeeb R Malik and Dennis M Bushnell. Role of computational fluid dynamics and wind tunnels in aeronautics r and d. Technical report, 2012.

[2] Antony Jameson. Re-engineering the design process through computation. *Journal of Aircraft*, 36(1):36–50, 1999.

[3] Edward Kraft. Integrating computational science and engineering to re-engineer the aeronautical development process. In *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*.

[4] Walter P Wolfe and Stuart S Ochs. Predicting aerodynamic characteristic of typical wind turbine airfoils using cfd. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 1997.

[5] Jonathon Sumner, Christophe Sibuet Watters, and Christian Masson. Cfd in wind energy: the virtual, multiscale wind tunnel. *Energies*, 3(5):989–1013, 2010.

[6] David Hartwanger and Andrej Horvat. 3d modelling of a wind turbine using cfd. In *NAFEMS Conference, United Kingdom*, 2008.

[7] WXM Koh and EYK Ng. Effects of reynolds number and different tip loss models on the accuracy of bem applied to tidal turbines as compared to experiments. *Ocean Engineering*, 111:104–115, 2016.

[8] Minghan Bao, Nagi Abdussamie, and Faisal AlThobiani. Prediction of wind loadings on offshore wind turbines using cfd. *Journal of Energy and Power Technology*, 5(1):1–19, 2023.

[9] Brian Lockwood, Mihai Anitescu, and Dimitri Mavriplis. Mixed aleatory/epistemic uncertainty quantification for hypersonic flows via gradient-based optimization and surrogate models. In *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 1254, 2012.

[10] Brian Lockwood, Markus Rumpfkeil, Wataru Yamazaki, and Dimitri Mavriplis. Uncertainty quantification in viscous hypersonic flows using gradient information and surrogate modeling. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 885, 2010.

[11] Soon-Duck Kwon. Uncertainty analysis of wind energy potential assessment. *Applied Energy*, 87(3):856–865, 2010.

[12] Jie Zhang, Souma Chowdhury, and Achille Messac. Uncertainty quantification in surrogate models based on pattern classification of cross-validation errors. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 5437, 2012.

[13] Atif Salahudeen. Application of uncertainty quantification of turbulence intensity on airfoil aerodynamics. Master's thesis, University of Maryland, 2017.

[14] Kinshuk Panda, Ryan King, Andrew Glaws, and Kristi Potter. Multi-fidelity active subspaces for wind farm uncertainty quantification. In *AIAA Scitech 2021 Forum*, page 1601, 2021.

[15] Anthony M. DeGennaro, Clarence W. Rowley, and Luigi Martinelli. Uncertainty quantification for airfoil icing using polynomial chaos expansions. *Journal of Aircraft*, 52(5):1404–1411, 2015.

[16] Narges Tabatabaei, Mehrdad Raisee, and Michel J. Cervantes. Uncertainty Quantification of Aerodynamic Icing Losses in Wind Turbine With Polynomial Chaos Expansion. *Journal of Energy Resources Technology*, 141(5), 04 2019.

[17] Xiaojing Wu, Weiwei Zhang, Shufang Song, and Zhengyin Ye. Sparse grid-based polynomial chaos expansion for aerodynamics of an airfoil with uncertainties. *Chinese Journal of Aeronautics*, 31(5):997–1011, 2018.

[18] Xiaojing Wu, Weiwei Zhang, and Song Shufang. Uncertainty quantification and sensitivity analysis of transonic aerodynamics with geometric uncertainty. *International Journal of Aerospace Engineering*, 2017:1–16, 02 2017.

[19] Francesco Papi, Lorenzo Cappugi, Simone Salvadori, Mauro Carnevale, and Alessandro Bianchini. Uncertainty quantification of the effects of blade damage on the actual energy production of modern wind turbines. *Energies*, 13(15), 2020.

[20] Hassan Moin, Hafiz Zeeshan Iqbal Khan, Surrayya Mobeen, and Jamshed Riaz. Airfoil's aerodynamic coefficients prediction using artificial neural network. In *2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 175–182. IEEE, 2022.

[21] Yolanda Mack, Tushar Goel, Wei Shyy, and Raphael Haftka. Surrogate model-based optimization framework: a case study in aerospace design. *Evolutionary computation in dynamic and uncertain environments*, pages 323–342, 2007.

[22] Juan J Alonso, Patrick LeGresley, and Víctor Pereyra. Aircraft design optimization. *Mathematics and Computers in Simulation*, 79(6):1948–1958, 2009.

[23] Trent W Lukaczyk, Paul Constantine, Francisco Palacios, and Juan J Alonso. Active subspaces for shape optimization. In *10th AIAA multidisciplinary design optimization conference*, page 1171, 2014.

[24] Seongim Choi, Juan J Alonso, Illan M Kroo, and Mathias Wintzer. Multifidelity design optimization of low-boom supersonic jets. *Journal of Aircraft*, 45(1):106–118, 2008.

[25] Richard W Fenrich, Victorien Menier, Philip Avery, and Juan J Alonso. Reliability-based design optimization of a supersonic nozzle. In *Proceedings of the 7th European Conference on Computational Fluid Dynamics*, 2018.

[26] Weifei Hu, KK Choi, and Hyunkyoo Cho. Reliability-based design optimization of wind turbine blades for fatigue life under dynamic wind load uncertainty. *Structural and multidisciplinary optimization*, 54:953–970, 2016.

[27] Quan Lin, Qi Zhou, Jiexiang Hu, Yuansheng Cheng, and Zhen Hu. A sequential sampling approach for multi-fidelity surrogate modeling-based robust design optimization. *Journal of Mechanical Design*, 144(11):111703, 2022.

[28] Yoshinori Matsuno, Takeshi Tsuchiya, and Naoki Matayoshi. Near-optimal control for aircraft conflict resolution in the presence of uncertainty. *Journal of Guidance, Control, and Dynamics*, 39(2):326–338, 2016.

[29] Gregory D Bird, Steven E Gorrell, and John L Salmon. Dimensionality-reduction-based surrogate models for real-time design space exploration of a jet engine compressor blade. *Aerospace Science and Technology*, 118:107077, 2021.

[30] Edward M Kraft. The air force digital thread/digital twin-life cycle integration and use of computational and experimental knowledge. In *54th AIAA aerospace sciences meeting*, page 0897, 2016.

[31] S Ashwin Renganathan, Kohei Harada, and Dimitri N Mavris. Aerodynamic data fusion toward the digital twin paradigm. *AIAA Journal*, 58(9):3902–3918, 2020.

[32] Pier Carlo Berri, Matteo DL Dalla Vedova, and Laura Mainini. Learning for predictions: Real-time reliability assessment of aerospace systems. *AIAA Journal*, 60(2):566–577, 2022.

[33] P Spalart and S Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting and Exhibit*, 1992.

[34] Philippe R Spalart and Andrey V Garbaruk. Correction to: A new "$\lambda 2$" term for the spalart–allmaras turbulence model, active in axisymmetric flows. *Flow, Turbulence and Combustion*, 109(1):253–253, 2022.

[35] Detlef Lohse and Axel Müller-Groeling. Anisotropy and scaling corrections in turbulence. *Physical Review E*, 54(1):395, 1996.

[36] Sai Hung Cheung, Todd A Oliver, Ernesto E Prudencio, Serge Prudhomme, and Robert D Moser. Bayesian uncertainty analysis with applications to turbulence modeling. *Reliability Engineering & System Safety*, 96(9):1137–1149, 2011.

[37] Brendan D Tracey, Karthikeyan Duraisamy, and Juan J Alonso. A machine learning strategy to assist turbulence model development. In *53rd AIAA aerospace sciences meeting*, page 1287, 2015.

[38] Anand Pratap Singh, Shivaji Medida, and Karthik Duraisamy. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA Journal*, 55(7):2215–2227, 2017.

[39] Eric J Parish and Karthik Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of computational physics*, 305:758–774, 2016.

[40] Jonathan R Holland, James D Baeder, and Karthikeyan Duraisamy. Field inversion and machine learning with embedded neural networks: Physics-consistent neural network training. In *AIAA Aviation 2019 Forum*, page 3200, 2019.

[41] Jonathan R Holland, James D Baeder, and Karthik Duraisamy. Towards integrated field inversion and machine learning with embedded neural networks for rans modeling. In *AIAA Scitech 2019 forum*, page 1884, 2019.

[42] CL Rumsey and GN Coleman. Nasa symposium on turbulence modeling: Roadblocks, and the potential for machine learning. Technical report, 2022.

[43] Tony Burton, Nick Jenkins, David Sharpe, and Ervin Bossanyi. *Component Design*, chapter 7, pages 383–473. John Wiley & Sons, Ltd, 2011.

[44] R. Barrett and A. Ning. Integrated free-form method for aerostructural optimization of wind turbine blades. *Wind Energy*, 21(8):663–675, 2018.

[45] M. H. Aa. Madsen, F. Zahle, N. N. Sørensen, and J. R. R. A. Martins. Multipoint high-fidelity cfd-based aerodynamic shape optimization of a 10 mw wind turbine. *Wind Energy Science*, 4(2):163–192, 2019.

[46] Muhammad A. R. Yass, Raghad Majeed Rasheed, and Amer Hamad Muhiesen. Contribution of lift-to-drag ratio on power coefficient of hawt blade for different cross-sections. *Open Engineering*, 12(1):716–728, 2022.

[47] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., USA, 1 edition, 1997.

[48] Raymond M. Hicks and Preston A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412, 1978.

[49] Helmut Sobieczky. *Geometry generator for CFD and applied aerodynamics*. Springer, 1997.

[50] Brenda M. Kulfan. Universal parametric geometry representation method. *Journal of Aircraft*, 45(1):142–158, 2008.

[51] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, NREL, Golden, CO, 2009.

[52] C. Bak, C. Zahle, R. Bitsche, T. Kim, A. Yde, L. C. Henrikson, M. H. Hansen, J. P. A. A. Blasques, M. Guanaa, and A. Natarajan. The DTU 10-MW reference wind turbine. Technical report, DTU, 2013.

[53] Evan Gaertner, Jennifer Rinker, Latha Sethuraman, Frederik Zahle, Benjamin Anderson, Garrett Barter, Nikhar Abbas, Fanzhong Meng, Pietro Bortolotti, Witold Skrzypinski, George Scott, Roland Feil, Henrik Bredmose, Katherine Dykes, Matt Shields, Christopher Allen, and Anthony Viselli. Definition of the IEA 15 MW Offshore Reference Wind Turbine. Technical report, International Energy Agency, 2020.

[54] Patrick J Moriarty and A Craig Hansen. Aerodyn theory manual. Technical report, National Renewable Energy Lab., Golden, CO (US), 2005.

[55] Jayanarayanan Sitaraman and Beatrice Roget. Solution algorithm for unstructured grids using quadrilateral subdivision and hamiltonian paths. In *52nd AIAA Aerospace Sciences Meeting - AIAA Science and Technology Forum and Exposition, SciTech 2014*, 01 2014.

[56] Y. S. Jung, G. Vijayakumar, S. Ananthan, and J. Baeder. Local correlation-based transition models for high-reynolds-number wind turbine airfoils. *Wind Energy Science Discussions*, 2021:1–30, 2021.

[57] M Costenoble, B Govindarajan, Y Jung, and J Baeder. “automated mesh generation and solution analysis of arbitrary airfoil geometries. *Wind Energy Science Discussions*, 2021.

[58] Koushik Marepally, Yong Su Jung, James Baeder, and Ganesh Vijayakumar. Uncertainty quantification of wind turbine airfoil aerodynamics with geometric uncertainty. In *Journal of Physics: Conference Series*, volume 2265, page 042041. IOP Publishing, 2022.

[59] Russel E Caflisch. Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7:1–49, 1998.

[60] Habib N Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual review of fluid mechanics*, 41:35–52, 2009.

[61] Ilya M Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280, 2001.

[62] Dan M Somers. Design and experimental results for the s809 airfoil. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 1997.

[63] W. A. Timmer and R. P. J. O. M. van Rooij. Summary of the Delft University Wind Turbine Dedicated Airfoils . *Journal of Solar Energy Engineering*, 125(4):488–496, 11 2003.

[64] O Pires, X Munduate, O Ceyhan, M Jacobs, and H Snel. Analysis of high reynolds numbers effects on a wind turbine airfoil using 2d wind tunnel test data. *Journal of Physics: Conference Series*, 753(2):022047, sep 2016.

[65] P. Bortolotti, N. Johnson, N. J. Abbas, E. Anderson, E. Camarena, and J. Paquette. Land-based wind turbines with flexible rail-transportable blades – part 1: Conceptual design and aeroservoelastic performance. *Wind Energy Science*, 6(5):1277–1290, 2021.

[66] Eric J. Parish and Karthik Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305:758–774, 2016.

[67] Michael Giles and Niles Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65:393–415, 04 2000.

[68] W. J. McCroskey and Ames Research Center. *A critical assessment of wind tunnel results for the NACA 0012 airfoil [microform] / W.J. McCroskey*. National Aeronautics and Space Administration, Ames Research Center Moffett Field, Calif, 1987.

[69] Shivaji Medida and James Baeder. Adverse pressure gradient modification to turbulence models for wall-bounded flows. In *21st AIAA Computational Fluid Dynamics Conference*, 2013.

[70] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[71] Xiaopeng Xu, Chonglei Sun, Yu Li, Jia Zhao, Junlei Han, and Weiping Huang. An improved tandem neural network for the inverse design of nanophotonics devices. *Optics Communications*, 481:126513, 2021.

[72] Ganesh Vijayakumar, Yong Su Jung, James D. Baeder, and Shreyas Ananthan. Design-space exploration for inverse-design of wind turbine blades using data-driven methods. In *AIAA SCITECH 2022 Forum*.

[73] A.D. Wright and L.J. Fingersh. Advanced control design for wind turbines; part i: Control design, implementation, and initial tests. Technical report, National Renewable Energy Laboratory, 2008.

[74] R.W. Derksen and Tim Rogalsky. Bezier-parsec: An optimized aerofoil parameterization for design. *Advances in Engineering Software*, 41(7):923–930, 2010. Advances in Structural Optimization.

[75] I. Abdallah, A. Natarajan, and J.D. Sørensen. Impact of uncertainty in airfoil characteristics on wind turbine extreme loads. *Renewable Energy*, 75:283–300, 2015.

[76] Rick R Damiani. Uncertainty and risk assessment in the design process for wind. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2018.

[77] J Slotnick PI, A Khodadoust PM, J Alonso, D Darmofal, W Gropp, E Lurie, and D Mavriplis. Cfd vision 2030 study: a path to revolutionary computational aerosciences. *NASA: Washington, DC, USA*, 2013.

[78] Koushik Marepally, Nick Paternostro, Bumseok Lee, and James D Baeder. Data puncturing and training strategies for cost-efficient surrogate modeling of rotorcraft airfoil aerodynamics. In *AIAA SCITECH 2023 Forum*, page 2042, 2023.

[79] Eric J Tuegel, Anthony R Ingraffea, Thomas G Eason, and S Michael Spottswood. Reengineering aircraft structural life prediction using a digital twin. *International Journal of Aerospace Engineering*, 2011, 2011.

[80] R.L. Haupt and S.E. Haupt. *Practical Genetic Algorithms*. Wiley InterScience electronic collection. Wiley, 2004.