Islamic University of Technology

Department of Mechanical and Production Engineering

DATA DRIVEN APPROACH TO CONSTRUCT FLOW FIELD OF A 2D AIRFOIL

A Thesis by

NAHIYAN CHOWHDURY
FAHMID HOSSAIN
ABRAR MAHI-Al-RASHID

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Bachelor of Science in Mechanical Engineering

June, 2022

DATA DRIVEN APPROACH TO CONSTRUCT FLOW FIELD OF A 2D AIRFOIL

NAHIYAN CHOWHDURY
STUDENT ID: 170011024

FAHMID HOSSAIN
STUDENT ID: 170011049

ABRAR MAHI-AL-RASHID
STUDENT ID: 170011053

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Bachelor of Science in Mechanical Engineering

DEPARTMENT OF MECHANICAL AND PRODUCTION ENGINEERING

June, 2022

CERTIFICATE OF RESEARCH

This thesis titled "DATA DRIVEN APPROACH TO CONSTRUCT FLOW FIELD OF A 2D AIRFOIL" submitted by NAHIYAN CHOWDHURY (170011024), FAHMID HOSSAIN (170011049) and ABRAR MAHI AL RASHID (170011053) has been accepted as satisfactory in partial fulfillment of the requirement for the Degree of Bachelor of Science in Mechanical Engineering.

_____

Dr. Md. Rezwanul Karim *(Supervisor)*

Associate Professor

Department of Mechanical and Production Engineering (MPE)

_____

Mr. Tahsin Sejat Saniat (*Co-Supervisor)*

Lecturer

Department of Mechanical and Production Engineering (MPE)

_____

Prof. Dr. Md. Anayet Ullah Patwari ***(Head of the Department)***

Professor

Department of Mechanical and Production Engineering (MPE)

Islamic University of Technology (IUT)

DECLARATION

I hereby declare that this thesis entitled "DATA DRIVEN APPROACH TO CONSTRUCT FLOW FIELD OF A 2D AIRFOIL" is an authentic report of our study carried out as requirement for the award of degree B.Sc. (Mechanical Engineering) at Islamic University of Technology, Gazipur, Dhaka, under the supervision of Dr. Md. Rezwanul Karim, Associate Professor, MPE, IUT in the year 2022

The matter embodied in this thesis has not been submitted in part or full to any other institute for award of any degree.

_____

Nahiyan Chowdhury

Student ID: 170011024

_____

Fahmid Hossain

Student ID: 170011049

_____

Abrar Mahi-al-rashid

Student ID: 170011053

Acknowledgments

In the Name of Allah, the Most Beneficent, the Most Merciful

At first, we show our gratitude to our creator Allah (SWT), the most merciful and the most benevolent for giving us the strength, skills and ability to complete this project and write this dissertation in spite of facing many struggles.

We show our greatest gratitude to our honorable supervisors, Dr. Md. Rezwanul Karim and co-supervisor Mr. Sayedus Salehin and Mr. Tahsin Sejat Saniat for their guidance and support without which it would have been almost impossible to complete our work.

Nahiyan Chowdhury
Fahmid Hossain
Abrar Mahi-al-rashid
June, 2022

Table of Contents

ABSTRACT

Machine learning is quickly becoming a significant scientific computer tool, with enormous potential to broaden the field of computational fluid dynamics. Various research has been conducted in the recent past which emphasized on how the use of different Machine Learning algorithm is playing an important role in the enhancement of computational fluid dynamics. In this work we try to discuss about how we created an architecture of a Machine Learning algorithm by using U-net, which is a type of convolutional neural network and tried to apply it in order to reproduce a flow field around a 2D airfoil, which can be easily, if not quickly, produced using CFD analysis. The principal aim of this thesis is to check whether our Deep learning architecture is capable of providing an acceptable prediction of the flow field or not. If the flow field from DL matches with that of CFD, then we can use this observation for further study and if it does not match, there is still room for further correction. In order to execute the experiment, 3325 CFD simulations were carried out and the flow fields achieved from the simulation as the result of the experiment were separated into two groups.80% of the data was taken for training the DL algorithm and 20% were used for validation. After implementing the DL algorithm, some of the results were found to be almost similar to the results produced from the CFD simulation.

Keywords: Computational Fluid Dynamic, Machine learning, Deep learning, U-net, Airfoil, Attention

NOMENCLATURE

CFD	Computational fluid dynamics
ML	Machine Learning
DL	Deep Learning
CNN	Convoluted Neural Network
AOA	Angle of Attack
DM	Designmodeller
DOE	Design of Experiment
Re	Reynolds Number
T	Absolute temperature
LES	Large-Eddy Simulation
RANS	Reynolds-averaged Navier–Stokes
MSE	Mean Squared Error
MAE	Mean Absolute Error
RMSE	Root-Mean-Square-Deviation

LIST OF FIGURES

LIST OF TABLES

Chapter 1: INTRODUCTION

1.1 Background

For many industrial applications, Fluid mechanics is of greatest significance. Governing equations for most of the fluid flow are non-linear Navier–Stokes equations which are partial differential equations of mass conservation and momentum conservation of Newtonian Fluid. Computational Fluid Dynamic (CFD), one of the significant subjects of numerical simulation, aims to solve these Navier–Stokes equations to model actual fluid flow. It offers detail information on different types of fluid flow. But one of the glaring limitations of CFD is that it can become computationally expensive or may be intractable due to the complexity of fluid mechanics especially for turbulent flow. Here with availability of sufficient amount of data, data driven approach can enhance the performance of CFD.

Deep learning, a subdivision of machine learning, is swiftly becoming a core component in data science, enabling numerous advances across many disciplines of engineering. [1] It is inspired form the neural structure of human body. The building block for Deep learning is Artificial Neural Network (ANN) and Convolutional Neural Networks (CNN) which make excellent non-linear approximation. CNN uses numerous layers to extract features from raw data. CFD could become one of its potential applications. Deep learning can increase the speed of high-fidelity simulations, develop both high and low fidelity simulation and also can enhance the understanding of some fluid flow by Reduced order Models. [2]

In this work we have presented a deep learning model “U-net with Attention block” to create velocity field over an airfoil from pressure field of that airfoil.

1.2 Deep Learning Algorithm:

Before discussing about the U-net architecture, we are discussing about the building block of the architecture which are given as follow.

1.2.1 CNN:

A convolutional neural network (CNN) is a kind of artificial neural network (ANN) which is generally applied to analyze visual imagery. CNN in U-net architecture consists of three layers which are,

1. Convolutional Layer
2. ReLU Layer
3. Max Pooling

These layers in contracting path reduces the spatial information while increasing feature information. In the expending the feature and spatial information are combined through up-convolutions and concatenations.

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning system capable of taking an input picture, assigning weight (learnable weights and biases) to certain aspects in the image, and discriminating between them. ConvNet requires far less pre-processing than other classification techniques. While traditional techniques need the hand-engineering of filters, ConvNets can learn these filters/characteristics with sufficient training.

A ConvNet's design is inspired by the structure of the Visual Cortex and is similar to the Human Brain's connection pattern of Neurons. Individual neurons may respond to stimuli only in a restricted area of the visual field known as the Receptive Field. Many of these fields overlap. A collection of similar fields can be utilized to fill the whole visual field.

An image is nothing more than a matrix of pixel values. Why not simply flatten the picture (for example, converting a 3x3 image matrix to a 9x1 vector) and feed it to a Multi-Level Perceptron for classification?

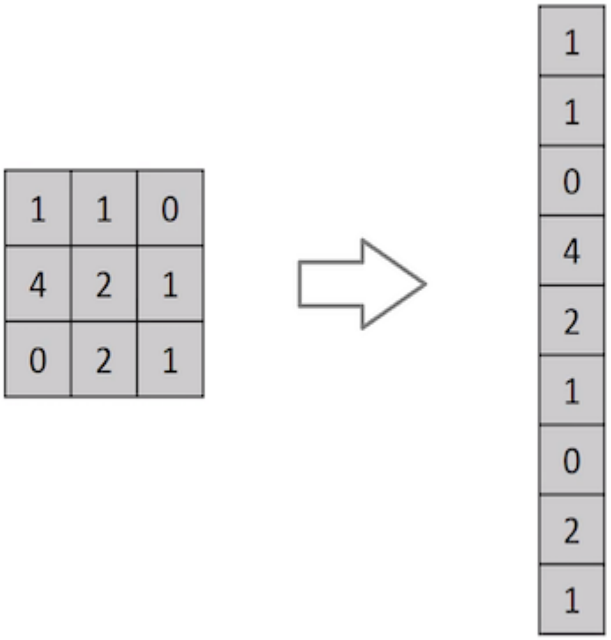

Figure 1: Flattening of a 3x3 image matrix into a 9x1 vector

When class prediction is applied to basic binary images, the approach may produce an average precision score; however, when applied to complex images with pixel dependencies throughout, the method yields little to no accuracy. Using proper filters, a ConvNet may capture the Spatial and Temporal correlations in an image. Because to the reduced number of parameters involved and the reusability of weights, the architecture achieves superior fitting to the picture dataset. In other words, the network may be trained to recognize the complexity of a picture.

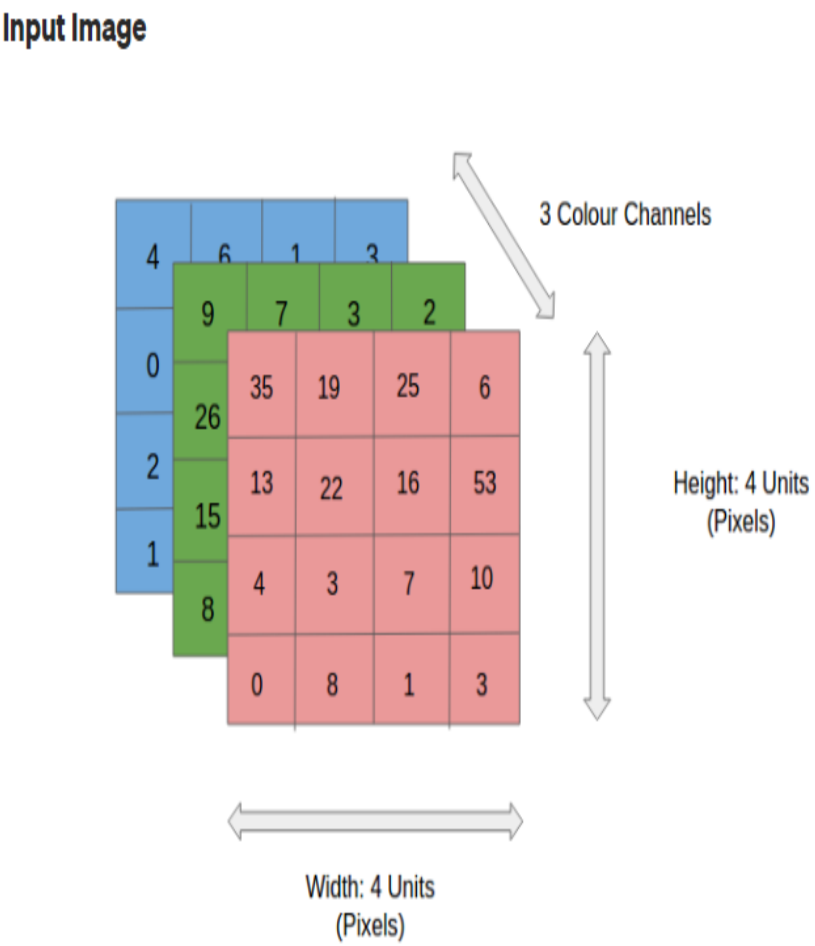

Figure 2: Matrix form of a 4x4x3 RGB Image

In the picture, we have an RGB image which has been divided by its three-color planes — Red, Green, and Blue. Grayscale, RGB, HSV, CMYK, and other color spaces exist in which images can be found.

One can anticipate how computationally hard things will get once the photos exceed 8K (76804320) dimensions. The role of the ConvNet is to compress the images into a form which is easier to process, without sacrificing features which are crucial for generating a decent forecast. This is critical for designing an architecture that is capable of learning features while also being scalable to large datasets.

1.2.2 Convolutional Layer:

Convolutional Layer is the first layer in a CNN and also the core building block. Through images

a convolution sweeps and calculates its input. By this convolution, a particular feature is detected from the input and feature maps are produced. The input is convolved using kernels. Using the sliding window method Kernels are applied across the image. If more kernels are applied then more features would be extracted. This kernel is slided across the input file, and computes a dot product which is provided to an activation map. Padding is used to assist the kernel with processing the image. Padding is the method of adding pixels to an image. In conjunction with padding, Stride is a parameter of the kernel that signifies the amount of movement over the image.

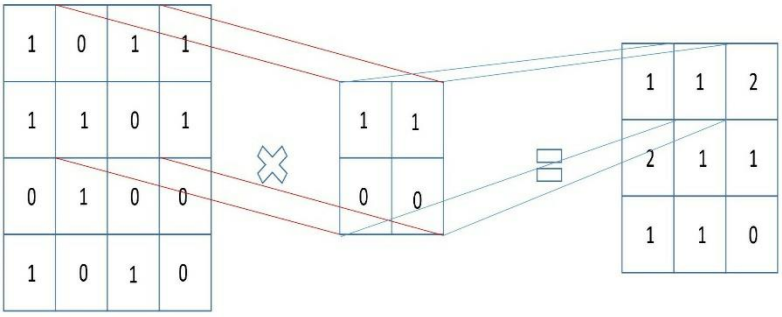

Figure 3: Convolutional operation with 2*2 kernel, 1 stride

Different kernel detects different features of the input. There is a formula which is used in determining the dimension of the activation maps:

$$Y = (N + 2P — F)/S + 1$$

Where, Y= Dimension of the output, N = Dimension of image, P = Padding, F = Dimension of Kernel, S = Stride.

1.2.3 ReLU Layer:

ReLU (rectified linear activation function) layer is one of the most widely used activation functions in Deep learning. It returns 0 if it gets negative value, and if a positive value is given, it returns the same value. ReLU function converts all the negative values to zero and it activates only for positive input. ReLU increases non-linearity in the CNN. The function is as below:

$$f(x) = max(0, x)$$

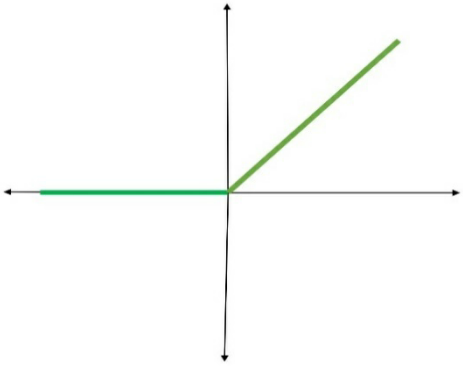

Figure 4: ReLU Function

1.2.4 Max Pooling:

In all the CNN block we have used Max Pooling. Max pooling is used to replace output with the max values to reduce data size. This reduces overfitting. Strides and Size are the two hyperparameter used for pooling.

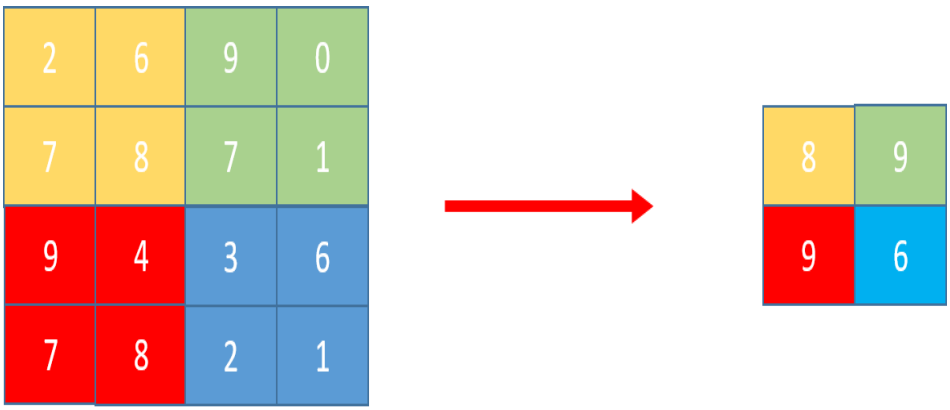

Figure 5: Max Pooling

1.2.5 U-net Architecture:

In “BioMedical Image segmentation” by Olag Ronneberger et al. [3]U-net was first introduced. It proved its potential in image segmentation and image transformation. Just like its predecessor Auto-encoders U-net has two parts which are contracting path or encoder and expending path or decoder. However, it differs in having a concatenating relation between the encoder and the decoder.

In the encoder there is a large number of kernel channels. This makes the network propagate context information to higher resolution layers. The encoder is symmetric to the decoder in the contracting path which make the architecture U-shape. The encoder has a typical CNN. It consists of two 3*3 Convolution layer each having a 3*3 max pooling layer and a ReLU layer. Similarly, the decoder has a 2*2 up-convolution, a concatenation with the correspondingly encoder’s cropped feature map, two 3*3 convolutions each having a ReLU layer. The architecture is shown in Figure 1.6:

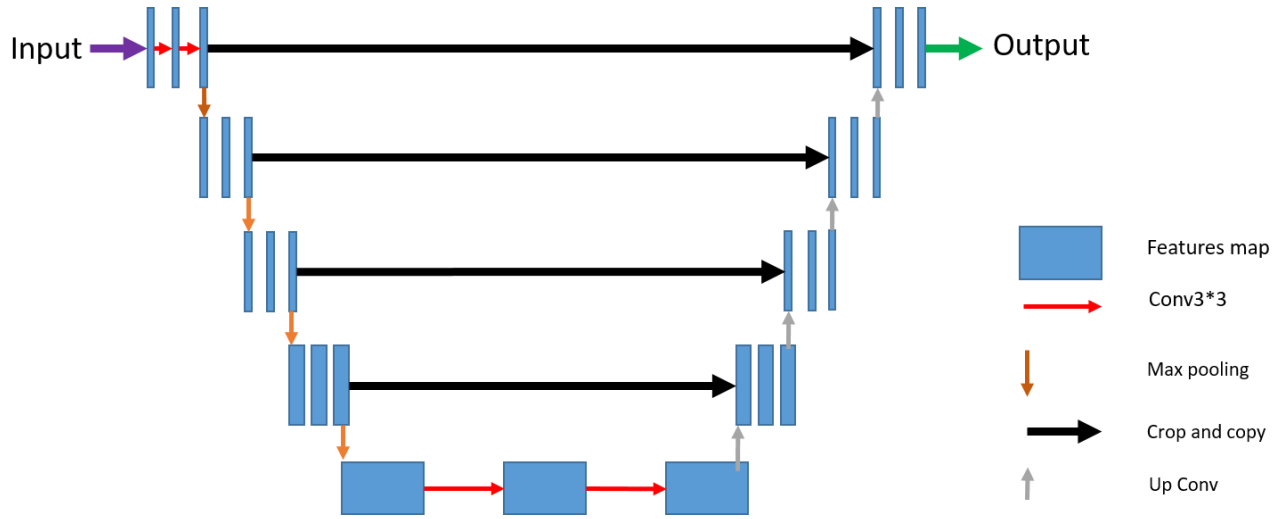

Figure 6: U-net Architecture

1.2.6 Attention Unit:

In deep learning Attention Unit imitate closely to cognitive attention. It was first introduced in "Attention Is All You Need" [4] paper. In image segmentation it reduces computational wastages, make the training of the algorithm faster and smother.

The attention gate takes two inputs which are vectors x and y. From the next lowest layer y vector is taken. The two vectors are summed. It provides larger aligned weights and relatively smaller unaligned weights. The resultant vector passes a ReLU layer and a sigmoid layer. Then it is up sampled using trilinear interpolation.

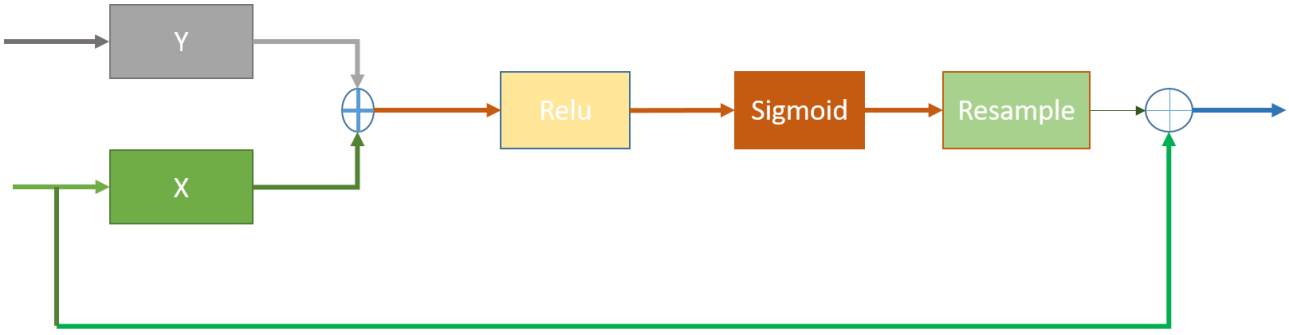

Figure 7: Attention Unit

1.2.7 U-net with attention unit:

During the connection between encoder and decoder many low levels of feature extractions occurs due to the poor feature representation in the initial layers. To avoid this redundant feature concatenation in the expending path attention units are added in the skip connection.

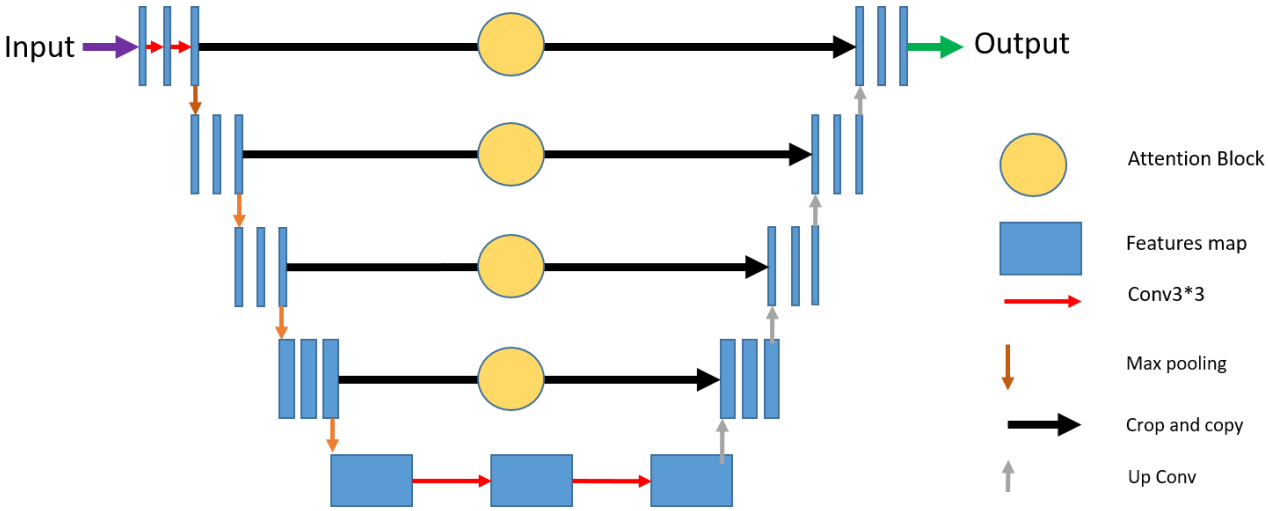

Figure 8: U-net with attention gate

1.3 Background of CFD

Computational fluid dynamics is the mathematical description and numerical solution of a physical event involving fluid flow (CFD). It is a field of fluid mechanics that investigates and solves fluid flow issues using numerical analysis and data structures. Computers are utilized to do the computations necessary to model the fluid's free-stream flow and interaction with boundary-constrained surfaces. High-speed supercomputers, which are often required to handle the most complex and large-scale problems, can produce superior results. Ongoing research results in software that increases the accuracy and speed of complicated modeling situations like transonic or turbulent flows. The first validation of such software is frequently accomplished using experimental equipment such as wind tunnels. Additionally, previous analytical or empirical judgments of a certain issue may be utilized for comparison. For final validation, full-scale testing, such as flight tests, is typically performed. CFD is used in a variety of applications, including aerodynamics, aerospace analysis, weather simulation, hypersonic, natural science and environmental engineering, biological engineering, fluid flows and heat transfer, industrial system analysis and design, visual effects for cinema and gaming, and engine and combustion analyses.

The following is a summary of CFD history:[5]

- Improvements in mathematical models and numerical approaches till 1910.
- Model and technique integration to obtain numerical answers based on manual computations between 1910 to 1940
- Early computers were able to solve flow over and around a cylinder with the aid of a mechanical desk calculator in 1953 by enabling the transition to computer-based computations (ENIAC). This was executed by Kawaguti.
- Between 1950 and 1960, the United States' Los Alamos National Laboratory conducted the first computer-based research on flow of fluid based on the famous equation derived by Navier and Stokes. Vorticity is calculated using the stream function method. This is the first implementation of 2D, transient, incompressible flow in the world.
- Hess and Smith released "Calculation of potential flow about arbitrary bodies" in 1967, the first scientific study on 3D computer research. Contribution of many methodologies, such as the Arbitrary Lagrangian-Eulerian model, the k-turbulence model, and the SIMPLE algorithm, which are all still employed in commercial code production.[6]
- Between 1960 and 1970, Boeing, NASA, and others debuted proceeded to deploy several deliverables like submarines, ships, motor transport and automobiles, helicopters and planes.[7], [8]
- Between 1980 and 1990, Jameson et al. enhanced the reliability of transonic flow solutions in three dimensions. Commercial codes are being used in both academics and industry.
- Significant advances in informatics are observed from 1990 to present where widespread use of CFD in nearly every field has been achieved.

1.3.1 Governing equations

The main framework of thermo-fluids research is guided by governing equations based on fluid physical property conservation laws. The three conservation laws are the fundamental equations:[9]

- The Continuity Equation and Mass Conservation
- Newton's Second Law of Momentum Conservation
- Energy Conservation from First Law of Thermodynamics

Within a closed system, mass, momentum, and energy are stable constants, according to these

notions. Everything, in essence, must be kept.

For the investigation of fluid flow with temperature fluctuations, certain physical parameters are essential. The three variables that are unknown, are required to be determined concurrently from the three widely established conservation equations are velocity **v**, pressure **p**, and absolute temperature **T**. Regardless, **p** and **T** are variables that are independent thermodynamically. The conservation equations' final version adds four extra thermodynamic variables: density, enthalpy **h**, viscosity, and thermal conductivity **k**. These latter parameters are determined in a unique fashion by the values of **p** and **T**.

Fluid flow should be evaluated to determine **vecv, p**, and **T** at all points of the flow regime. This is crucial when developing any device that incorporates fluid flow. Furthermore, measuring fluid flow using kinematic characteristics is a key topic. Lagrangian and Eulerian methods can be used to study fluid movement. The Lagrangian theory of fluid motion is founded on the notion of following a sufficiently large fluid particle to distinguish properties. Initial coordinates at time $\mathbf{t_0}$ must be checked, as well as the instantaneous coordinates at time $\mathbf{t_1}$. Following millions of individual particles down the journey is nearly difficult.

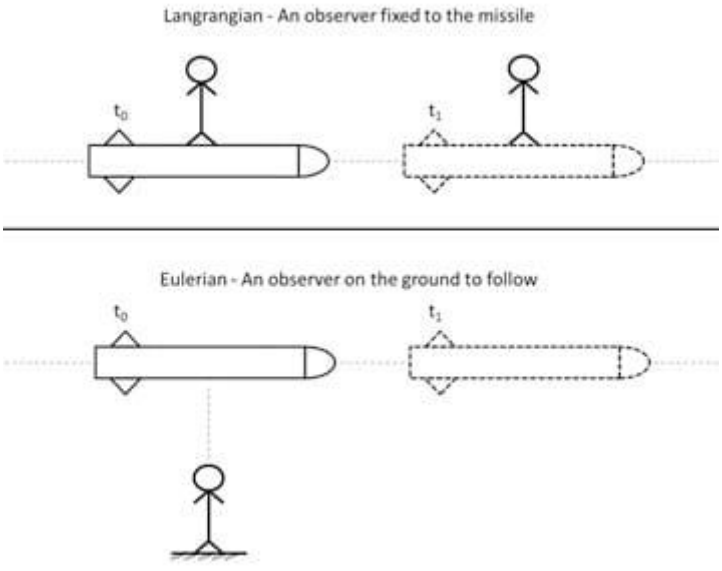

Figure 9: Difference between Lagrangian and Eulerian Approach [1]

Motion is always time-dependent in the Lagrangian formulation. As a, b, and c are a particle's

starting coordinates, x, y, and z are the same particle's coordinates at time t. A Lagrangian flow's motion is described as follows:

$$x = x(a, b, c, t) y = y(a, b, c, t) z = z(a, b, c, t)$$

The constituents of velocity at the position (x,y,z) at time t are denoted by u, v, and w in the Eulerian method. As a result, the independent variables x, y, z, and t are functions of the unknown variables u, v, and w. The velocity of an Eulerian flow at time t is described as follows:

$$u = u(x, y, z, t) v = v(x, y, z, t) w = w(x, y, z, t)$$

1.3.2 Mass Conservation

The equation for mass conservation is as follows:

$$D\rho Dt + \rho(\nabla \cdot v^{\rightarrow}) = 0$$

Where. ρ is the density, $v^{\rightarrow}$ the velocity and ∇ the gradient operator.

$$\nabla^{\rightarrow} = i^{\rightarrow} \partial\partial x + j^{\rightarrow} \partial\partial y + k^{\rightarrow} \partial\partial z$$

The flow is considered to be incompressible if the density is constant, and the continuity equation becomes:

$$D\rho Dt = 0 \rightarrow \nabla \cdot v^{\rightarrow} = \partial u \partial x + \partial v \partial y + \partial w \partial z = 0$$

1.3.3 Momentum Conservation

The Navier-Stokes Equation, which describes the conservation of momentum, is given by:

$$\frac{\partial}{\partial t}(\rho\vec{v})^{\frown I} + \nabla \cdot (\rho\vec{v}\vec{v})^{\frown II} = -\nabla p^{\frown III} + \nabla \cdot (\underline{\tau})^{\frown IV} + \rho\vec{g}^{\frown V}$$

Where, p is static pressure, $\tau^{\underline{\ }}$ is viscous stress tensor and $\rho g^{\rightarrow}$ is the gravitational force per unit volume. Here, the roman numerals denote:

I: Local change with time

II: Momentum convection

III: Surface force

IV: Diffusion term

V: Mass force

According to Stoke's Hypothesis, the viscous stress tensor may be defined as follows:

$$\tau ij = \mu \partial vi \partial xj + \partial vj \partial xi - 23(\nabla \cdot v^{\rightarrow})\delta ij$$

If the fluid is incompressible and has a constant viscosity coefficient, the Navier-Stokes equation is reduced to:

$$\rho Dv^{\rightarrow} Dt = -\nabla p + \mu \nabla 2v^{\rightarrow} + \rho g^{\rightarrow}$$

1.6.4 Energy Conservation

The first law of thermodynamics asserts that the amount of work and heat contributed to a system will result in an increase in the system's energy:

$$dEt = dQ + dW$$

Where, *dQ* is the amount of heat delivered to the system, *dW* is the amount of work done on the system, and *dEt* is the increase in total energy. An example of a frequent sort of energy equation is:

$$\frac{\partial}{\partial t}(\rho \vec{v})^{\frown I} + \nabla \cdot (\rho \vec{v} \vec{v})^{\frown II} = -\nabla p^{\frown III} + \nabla \cdot (\underline{\tau})^{\frown IV} + \rho \vec{g}^{\frown V}$$

The roman numerals are:

I: Local change with time

II: Convective term

III: Pressure work

IV: Heat flux

V: Source term

1.3.5 Partial Differential Equation (PDE)

The mathematical model only provides interrelationships between transport factors that are either directly or indirectly involved in the entire operation. Regardless of whether or not every parameters in those equations discussed has a relative influence on the real circumstances, any interchange in parameters must be evaluated concurrently using the numerical methods for solution, which includes differential equations, vector and tensor notations. A PDE has several variables and is represented by the symbol "". When an equation is derived using "d," it is referred to as an Ordinary Differential Equation (ODE) since it only has one variable and its derivation. PDEs are used to convert the differential operator () into an algebraic operator in order to achieve a solution. PDEs are frequently employed in heat transfer, fluid dynamics, acoustics, electronics, and quantum physics to address issues. Thomee's work might be valuable in building on a concept. [10]

Example of ODE:

$$d2xdt2 = x \rightarrow x(t)$$

Where, T is the single variable

Example of PDE:

$$\partial f \partial x + \partial f \partial y = 5 \rightarrow f(x, y)$$

Where, both x and y are the variables

The numerical solution is an approach based on discretization for providing approximate answers to tough problems that analytic methods are incapable of handling. Furthermore, the precision of the discretization has a substantial influence on the numerical solution's validity. Some of the most common discretization methods include finite difference, finite volume, finite element, spectral (element) approaches, and boundary element.

1.3.6 CFD Methodology

CFD methodology can be divided into three parts[11], [12], which are:

1. Preprocessor
2. Solver
3. Postprocessor

A brief description of the whole CFD methodology is given in the figure in Figure 1.10.

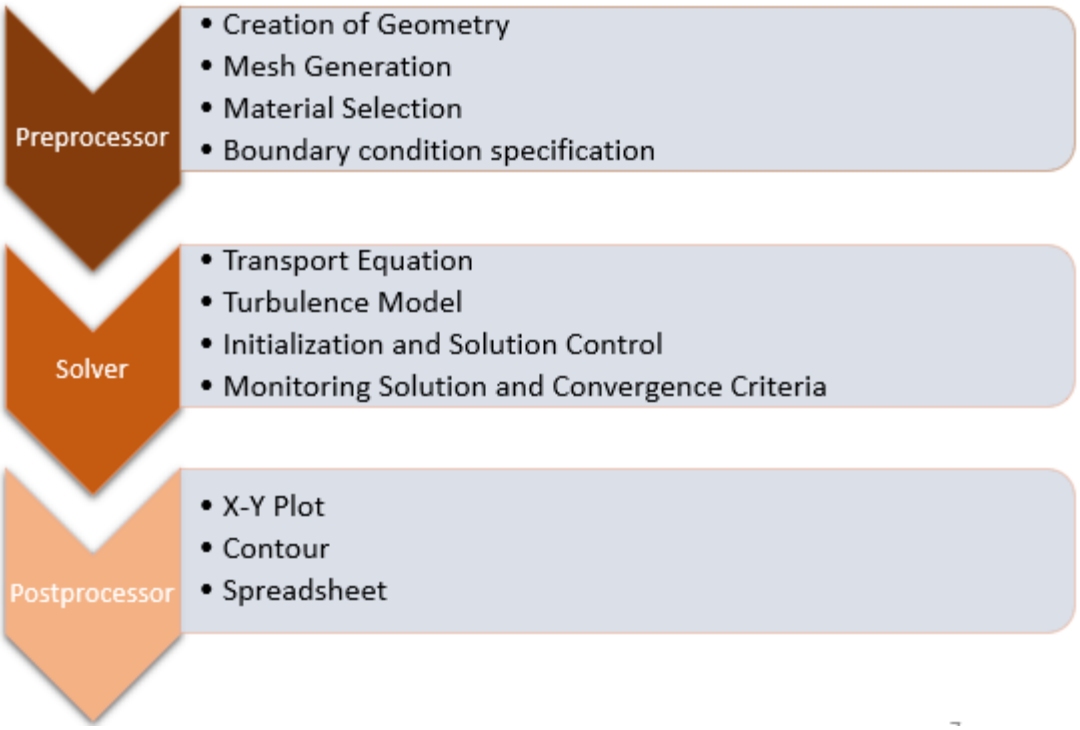

Figure 10: Methodology of CFD

1.3.7 Preprocessor

The first step in any CFD study is to define and construct the geometry of the flow area, which acts as the computational domain for the CFD calculations. The geometry can be created in any CAD software like Solidworks, Autocad, Autodesk and other CAD softwares or the built-in software in ANSYS, which are Spaceclaim and Designmodeller. [13]–[16]

For investigation, the solution domain is divided into multiple cells. Mesh refers to the computational framework's integration of these cells. Mesh is a method of dividing a domain into tiny cells or components, allowing a mathematical model to be applied to each cell under the condition of linearity. This indicates that the behavior and attitude of the variable parameters to be resolved must be linear inside each cell. This criterion also suggests that a finer mesh is required in locations where the predicted physical qualities are likely to be highly dynamic. Readers are highly encouraged to read these literatures [17], [18] in order to understand more about mesh. Mesh structure errors are a common cause of simulation failure. This might be because the mesh applied is sufficiently coarse, not covering all the results that happens to occur in this single cell element in a synchronized manner, but rather covering a variety of results that change as the mesh becomes finer. Thus, an examination of independence is required. The structure of mesh has a major effect on the precision of the solution. The analyst must pay close attention to the kind of cell, the quantity of cells, and the calculation time in order to implement precise solutions and acquire reliable results. Mesh convergence is defined as the optimization of these constraints, which may be organized in the following manner:

a. Produce a mesh structure with an acceptable number of parameters and ensure that the quality of mesh and coverage of the CAD model are sufficient for inspection. Carry out the inquiry.

b. In mesh constructions, increase the number of elements. Repeat the analysis and suitably compare the solutions in the aftermath.

c. Continue to fine-tune the mesh until the results match the previous ones.

After meshing, we have to select the materials to be used and also select the flow physics. A flowchart for flow physics is shown in the Figure 1.11.

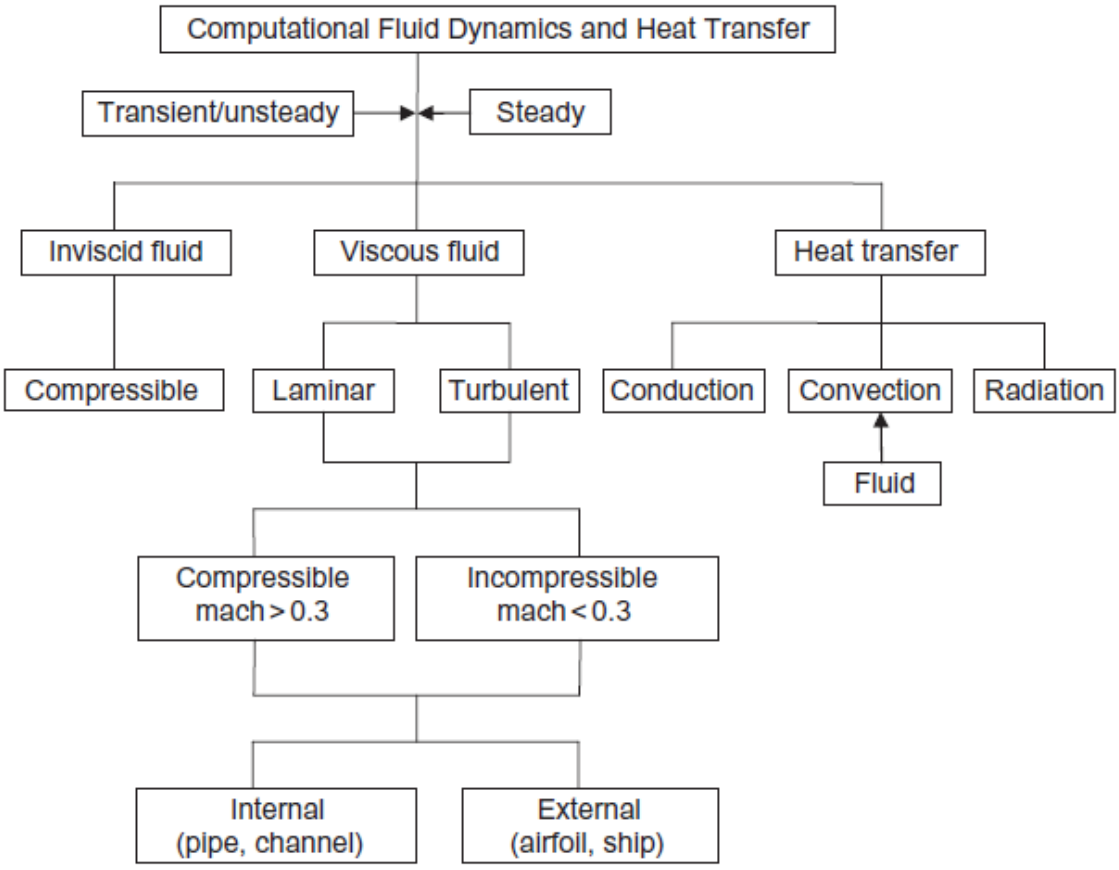

Figure 11: Flowchart for flow physics in CFD [7]

Proper starting and boundary conditions must be utilized while solving the Navier-Stokes equation and the continuity equation. Here we shall discuss about Neumann and Dirichlet Boundary Conditions briefly. For further knowledge of boundary condition, readers can look into this. [19]

Neumann and Dirichlet Boundary Conditions

• A Dirichlet boundary condition describes a variable's value at the boundary, such as

$$u(x) = constant$$

• When using a Neumann boundary condition, one defines the gradient normal to the variable's border, for example

$$\partial_n u(x) = constant$$

• Remember that multiple types of boundary conditions may be used for different variables at the same boundary.

Inlet

Inflow

• Variables transferred on the boundary, either by a predefined profile or (at the user's discretion) by executing an initial 1D, fully developed-flow calculation.

Stability

• Fixed total pressure and temperature (in compressible flow) or total head (in incompressible flow); typical compressible input condition

Outlet

Outflow

• The normal gradient of all variables is 0.

$$\partial_n(\emptyset) = 0$$

Pressure

• If the exit is subsonic, except for the constant pressure value, the usual outlet condition in compressible flow.

1.3.8 Solver

In the solver section, let us begin with turbulence modeling. One of the most challenging tasks in science and mathematics is calculating turbulent flows. For decades, scholars have been confused by exact turbulence solutions, and it is commonly believed that there is no closed form solution to

any fluid flow issue except the most fundamental laminar circumstances. Regardless, there are ways for doing calculations with enough precision to make engineering and design decisions. The accuracy of turbulent simulations has gradually grown as processing capabilities have become more powerful and numerical modeling has progressed. The first CFD problems were fairly simple, 2D incompressible steady-state settings with laminar flows. The first three-dimensional CFD simulation, as far as we know, was not completed until 1967. [6] Simultaneously, the first climate models were being developed to represent global fluid circulation. Shortly thereafter, progress increased as processing power and modeling approaches improved. The incorporation of turbulence modeling into CFD solutions was a considerable step forward. Early turbulence models accounted for turbulence impacts using a concept known as "eddy viscosity." Eddy viscosity is a notable increase in viscosity caused by small-scale chaotic disturbances in a fluid (or turbulent viscosity). The models do not attempt to replicate microscale turbulent dynamics, but rather attempt to mimic them by increasing fluid viscosity. As we shall see, turbulent viscosity is significant in Reynolds Averaged Navier Stokes (RANS) models. Other options, as we will see, do not rely substantially on the turbulent viscosity concept. A detailed study on turbulence modelling was conducted by Gorman et al [20], which will give the readers a good idea about turbulence modelling. The categorization of turbulence modelling is shown in Figure 1.12.

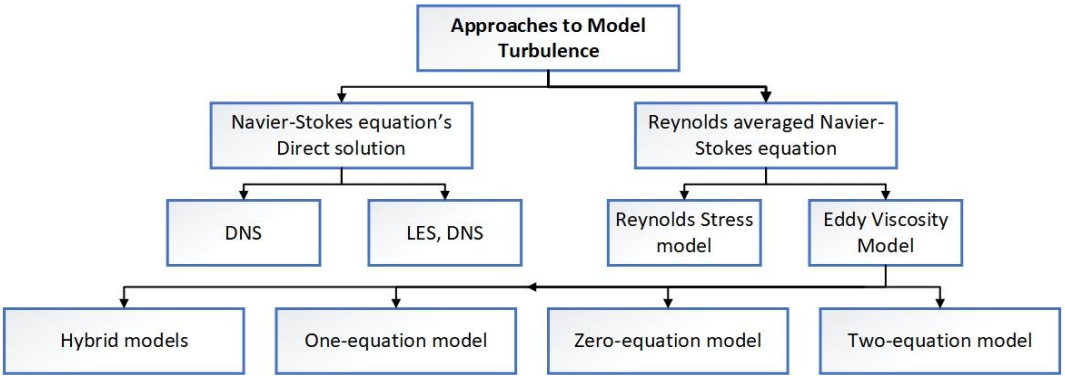

Figure 12: Categorization of Turbulence Modelling

Convergence is an important concept in computational analysis. The flow of fluid has a nonlinear mathematical model with several intricate models such as turbulence, phase shift, and mass transfer, all of which have a substantial influence on convergence. In addition to the analytical model, the numerical solution uses an iterative process to get answers by lowering errors between stages. The error is indicated by the difference between the final two values. The result gets more

dependable as the absolute error falls, suggesting that it is reaching a stable solution. How can analysts know about the convergence of a solution? Even if the intended situation is transient, convergence should continue until a steady-state condition is reached, suggesting that outputs change over time. Each time step must achieve convergence as if they were all steady-state processes. What are the requirements for convergence? Like stone remnants, equation residuals change with each iteration. Convergence happens as iterations get closer to a certain value. Those activities must be conducted for every time step in a temporary circumstance. Convergence can also be altered in the following manner:

- Initial conditions, Courant number, under-relaxation may all be used to speed up the process.
- The answer does not necessarily have to be right, but it must converge; the chosen mesh and mathematical model would be erroneous or have ambiguities.
- Stabilization approaches include adequate quality and refinement of mesh, and first- to second-order discretization procedures.
- If required, verify that the solution is reproducible to avoid misunderstanding.

1.3.9 Postprocessor

After solving the Navier-Stokes equations numerically, it is now time for postprocessing. In the postprocessor, we retrieve the output by means of X-Y plots, vectors, contours, line contours, streamlines and many other formats. It is plausible to create animations of the results.

1.4 Problem Statement

The generation of flow field around an external object is a common simulation experiment which is practiced by many beginners. In our thesis we focused on generating the flow field of a 2D airfoil by using a data driven approach. This is a common and beginner level simulation that can be achieved by any CFD software. But generating the flow field using Machine Learning (ML), and making prediction is something that is not much studied. The challenge here is to build the architecture of ML using CNN in order to predict the flow field of any given model of NACA airfoils at a different angle of attack (AOA) and Reynolds number (Re).

1.4 Objectives and Goals

The objectives and goals of our experiment is simple. In short, the aim of this thesis is:

1. To simulate and analyze the flow regime of a 2D airfoil using ANSYS Fluent.
2. To collect the velocity and pressure filed around the airfoil from Fluent.
3. To collect the velocity and pressure filed around the airfoil from U-net.
4. To compare and contrast the results of Fluent with that of U-net.

1.5 Methodology

The methodology can be summarized by the following:

- At first, the different geometry of the NACA models were collected.
- Then geometry of the models was generated.
- After this meshing was applied.
- Boundary conditions were then specified.
- With a suitable turbulence model, in this case Spalart-Allmaras model is used, the calculation is executed.
- The velocity and pressure fields were collected.
- The experiment was repeated 3325 times by changing parameters like NACA models, angle of attack and Reynolds number.
- The ML algorithm is trained with 80%of the data and 20% data were used for validation.

Chapter 2: LITERAURE REVIEW

2.1 Introduction

From the literature review it can be found that there is a huge potential for Machine Learning in improving CFD. The areas where Machine learning is applicable are as follow, increasing the computational power of DNS (direct numerical simulations), Turbulence modelling and developing reduced order model for better the physical understanding. [21] Our work mainly focuses on Turbulence modelling particularly RANS modeling. Ling et al. [22] gave a feasibility study on ML for RANS modeling. Duraisamy et al. [23] review work gives an excellent understanding on ML application in RANS modeling. Ahmed et al. [24] also provides emerging ML approaches to RANS modeling.

Obiols-Sales et al. [25] proposed a process in order to help increase the processing power for convergence of RANS simulations on the basis of Spalart–Allmaras (SA). Kutz. [26] provided an architecture to embed Galilean invariance. There are also physical informed ML models to improve RANS modeling [27]. Jiang et al. [28] developed a physics-informed residual network for RANS modelling based. Weatheritt and Sandberg [29] using gene-expression programming provided interpretable RANS models. J.-L. Wu [30] proposed a physical informed Random Forest for RANS modeling. Mi et al. [31] developed neural networks which can detect the different flow regimes. Use of Artificial Neural Network can also be seen in CFD. Beck et al. [32] used ANN with convolutional filter to develop the mapping of the flow. Lapeyre et al.[33] also used the similar approach but with CNN. Another interesting approach developed by Novati et al.[34] using multi-agent reinforcement-learning in turbulence modeling. Guo et al [35] used CNN to predict velocity field over several geometrical shapes achieving 98% accuracy. Ma et al [36] proposed DL model for closure of two-fluid bubble flow. Gibou [37] provided different directions for ML applications in multiphase flow. Skinner [38] used CNN for airfoil optimization from airfoil parameters.

Machine learning is seen to be applied in high-fidelity cases too. Many machine learning approaches are developed recently to increase the efficiency of DNS. Bar-Sinai et al. [39] proposed a deep learning technique to estimate spatial derivatives in low-resolution grids. Jeon

and Kim [40] developed a deep neural network to simulate finite-volume discretization and they tested it with reactive flows which obtained excellent agreement with reference. Stevens and Colonius [41] improved the accuracy of finite-difference/finite-volume methods by using fully convolutional LSTM network. Decreasing the size of the computational domain needed to retain physical properties of the system is an approach to accelerate numerical simulations. Fukami et al. [42] used a convolutional autoencoder with an MLP to develop a time-dependent inflow generator for wall-bounded turbulence simulation. It was tested in a turbulent channel flow at 180 Reynolds Number. Another approach could be without simulating the far field to set the right pressure-gradient distribution. Morita et al. [43] developed such a method using Bayesian optimization based on Gaussian-process regression which showed promising results. T. Shan et al. [44] solved Poisson's equation to accelerate CFD using deep learning technique. A. Ozbay et al. [45] used fully-convolutional neural networks to solve the Poisson problem. It decomposed the problem into a homogeneous Poisson problem and multiple inhomogeneous Laplace subproblems which resulted in lower percentage errors. This method could also be used in lower fidelity that rely on turbulence models.

2.2 Scopes and Limitations

In this work, we have used ML to enhance the performance of CFD and provide an alternative route to achieve the flow field of a 2D airfoil. In case of using ML as a catalyst for the performance of CFD, there are mainly three major areas in which we can focus on. The areas are – direct numerical solution (DNS), turbulence modelling and reduced-order model (ROM). The relationship of ML and these three models are presented visually in the Figure 2.1.

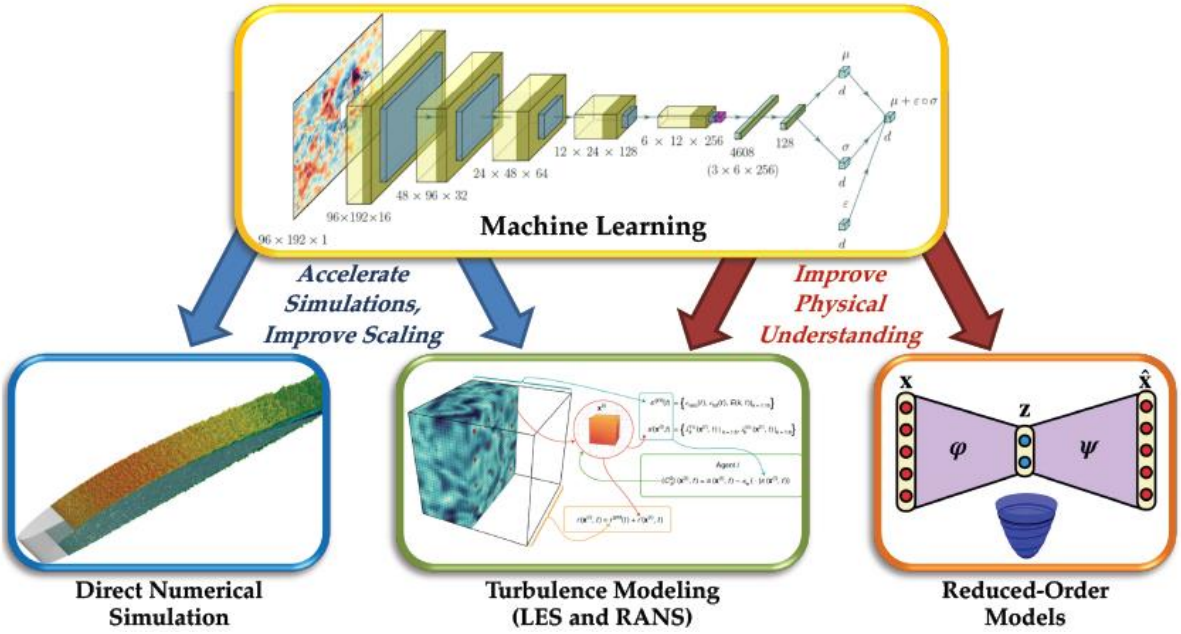

Figure 13: In the context of DNS, turbulence modeling, and ROM, a breakdown areas where ML could help CFD to improve. Image source: [36]–[39]

Deep learning, for instance, may be costly to train and requires a large amount of data. As a result, it is vital to identify areas where machine learning outperforms decades-old methodologies with increased accuracy and efficiency. Besides, the question of how training data is collected and if the associated costs are included when benchmarking. In this circumstance, transfer learning is a viable area for enhancing CFD. [50]

It should also be noted that there are deep learning options that may be better suited for specific applications. Finally, the information available to the user on the training data must be evaluated: certain flow factors was included in the ML model to increase learning efficiency and prediction accuracy. [21]

A variety of machine learning algorithms have recently been developed to increase DNS efficiency. Bar-Sinai et al. [51] suggested a deep learning-based strategy for estimating spatial derivatives in low-resolution grids that outperformed classic finite-difference techniques. Stevens and Colonius devised a similar strategy to enhance the performance of fifth-order finite-difference techniques in shock-capturing simulations.[52]

DNS is unfeasible for many application scenarios as a result of the computational cost of resolving all scales for high Reynolds number flows, as well as challenges caused by complicated geometries. Industrial CFD often depends on either RANS models, which simulate no turbulent scales, or coarsely resolved LES, which resolve just the biggest turbulent scales and model lesser ones.[21]

Machine learning is also being utilized in fluid dynamics to create reduced-order models (ROMs). ROMs are based on the notion that even complicated flows frequently have a few prominent coherent features. [53], [54]

Creating a ROM entails determining a set of reduced coordinates, which often describe the amplitudes of critical flow structures, and determining a differential-equation model for how these amplitudes change over time. Either of these phases have experienced significant breakthroughs in machine learning in recent years. One popular ROM strategy is to learn a low-dimensional system of coordination also with correct orthogonal decomposition (POD). [54], [55]

Chapter 3: EXPERIMENTAL METHODOLOGY

3.1 Introduction

In chapter 1, we have briefly discussed about our methodology. Here we discuss the methodology in details. For clarity, we have divided our experimental methodology into three parts which are – data preprocessing, training and validation, and inference. The figure below shows the brief methodology of our experiment. We have used the works of Vinothkumar et al. [56] , who developed a DL approach to predict the flow field around 2D airfoil, as our benchmark.

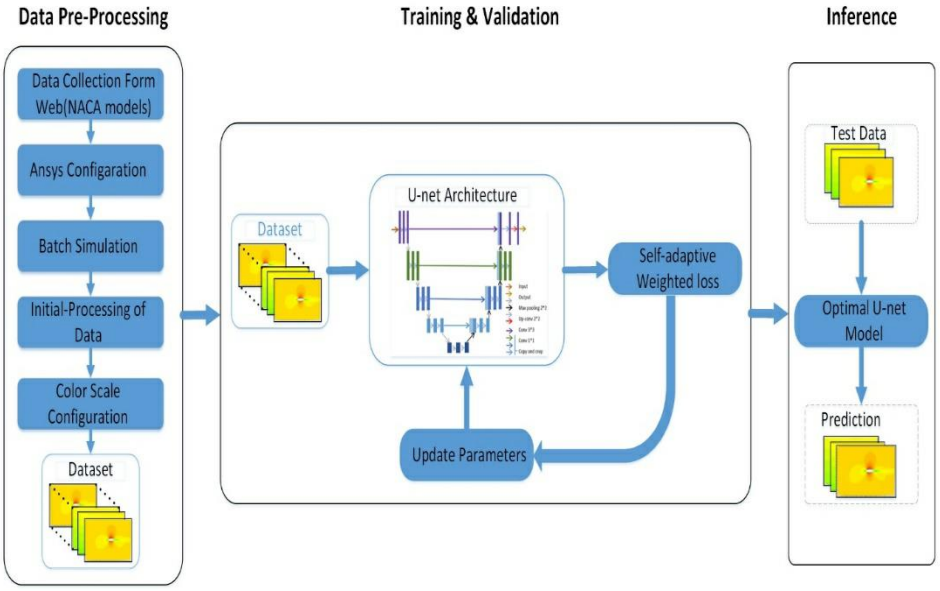

Figure* 14*: Experiment Methodology

3.2 Data Pre-processing

In this part, we had done several works. The CFD simulation as well as the collection of data had been executed in this step. At first, we have collected the coordinates of different NACA models from this site [57]. 95 different NACA models are selected. For each model, simulation was run for five sets of angle of attack (AOA), which are 0º, 3º, 6º, 9º and 12º, and seven sets of Reynolds number (1100000, 1200000, 1300000, 1500000, 1600000, 1800000, 2000000). In total 3325 cases were simulated. The framework of

data collection is shown in the Figure 3.2.

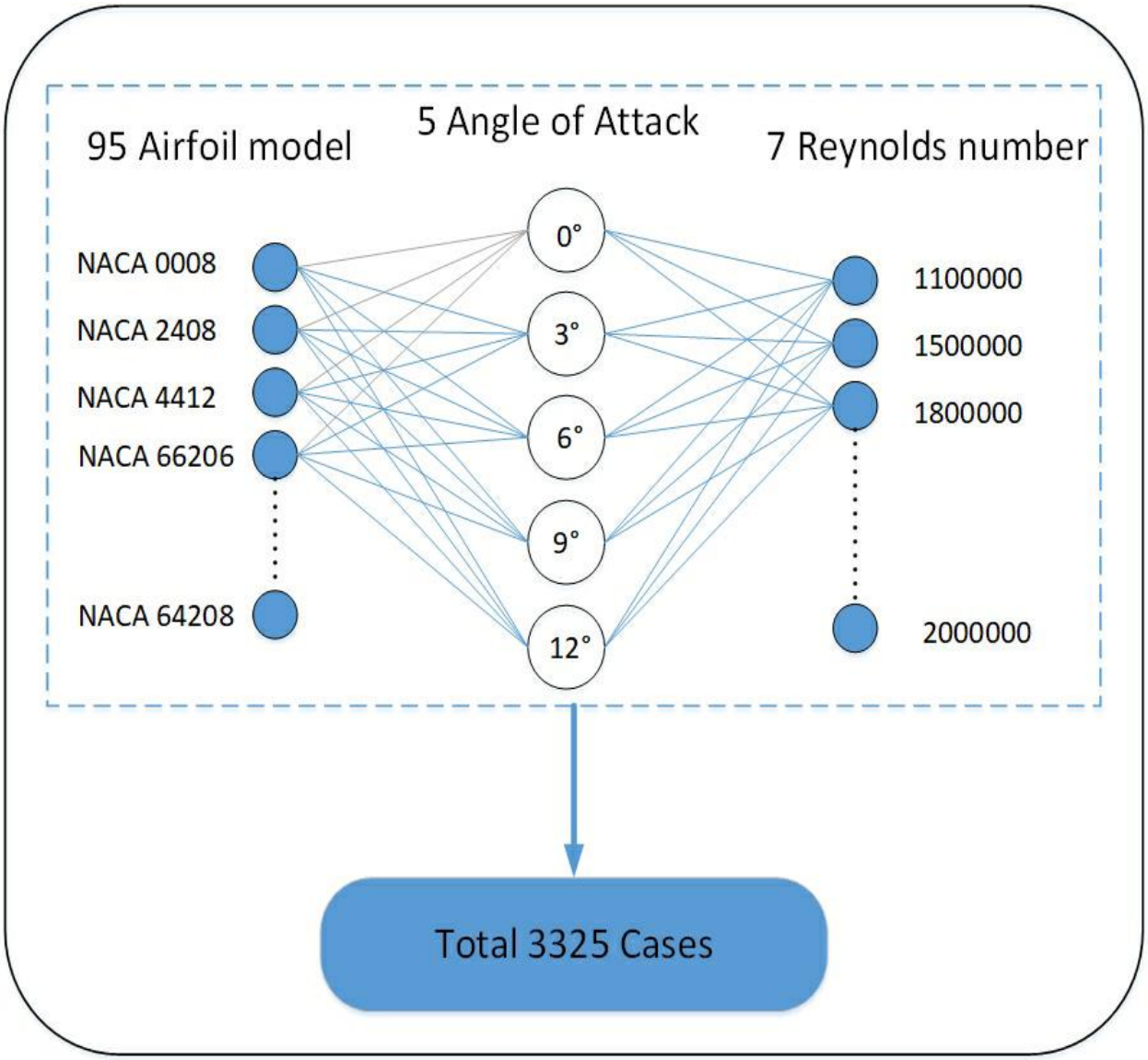

Figure* 15*: Framework of Data Collection

After the NACA models were collected, geometry of the model is formed. After the formation of geometry, a suitable structured mesh was generated. The mesh was relatively coarse. For the solution, Spallart-Almaras model was chosen as the turbulent model. In this way different sets of NACA were configured in ANSYS FLUENT and the boundary conditions were selected. Then batch simulation was executed. From the results in post-CFD, we had to reconfigure the color scale and normalize the dimension of the geometry. The images of pressure and velocity regimes were collected using a fixed scale. The data generated was then used for training and validation. The flowchart of data pre-processing is shown in Figure 3.3.

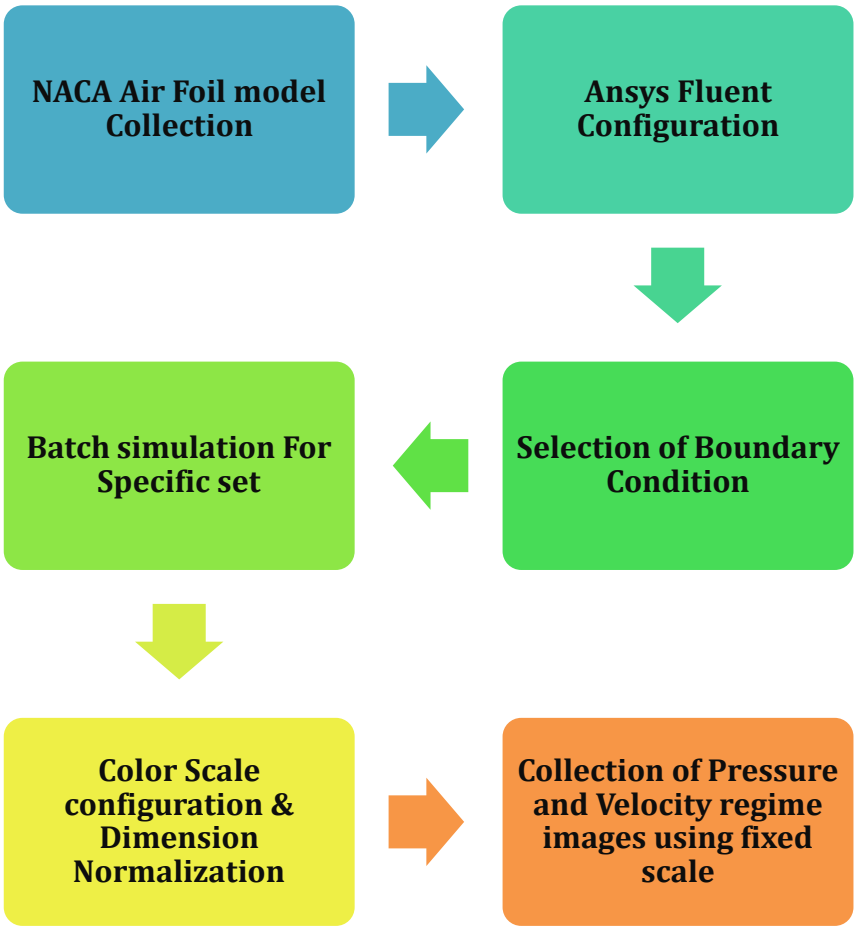

Figure* 16*: Flowchart of Data Pre-processing

Our workflow in the workbench of ANSYS Fluent is shown in the figure below. This workflow is for NACA 644221. In the workbench, we have arranged this NACA model into 35 different projects. In each row, we have kept the Reynolds number constant and changed the angle of attack. In each column, we have kept the angle of attack constant and the Reynolds number is changed.

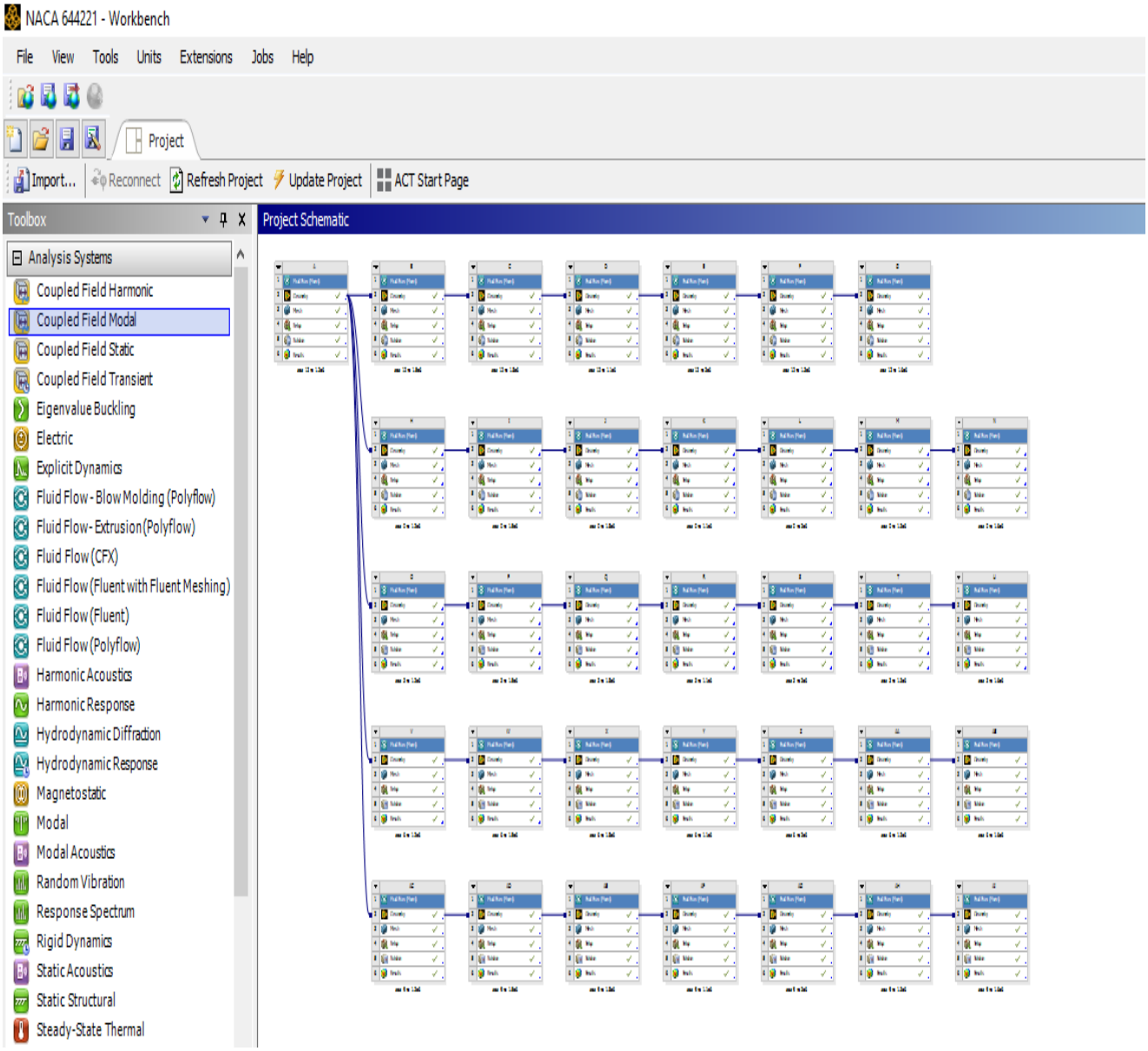

Figure 17: Workbench arrangement for NACA 644221

We have made our geometry in Designmodeller. At first, extracted the csv files of different NACA models and imported it into the DM. In the DM we made a sketch of the airfoil and the environment around it with using operations like surfaces from edges, surfaces from sketch, sketch, rotation and boolean. The geometry of NACA 644221 is shown in Figure 3.5.

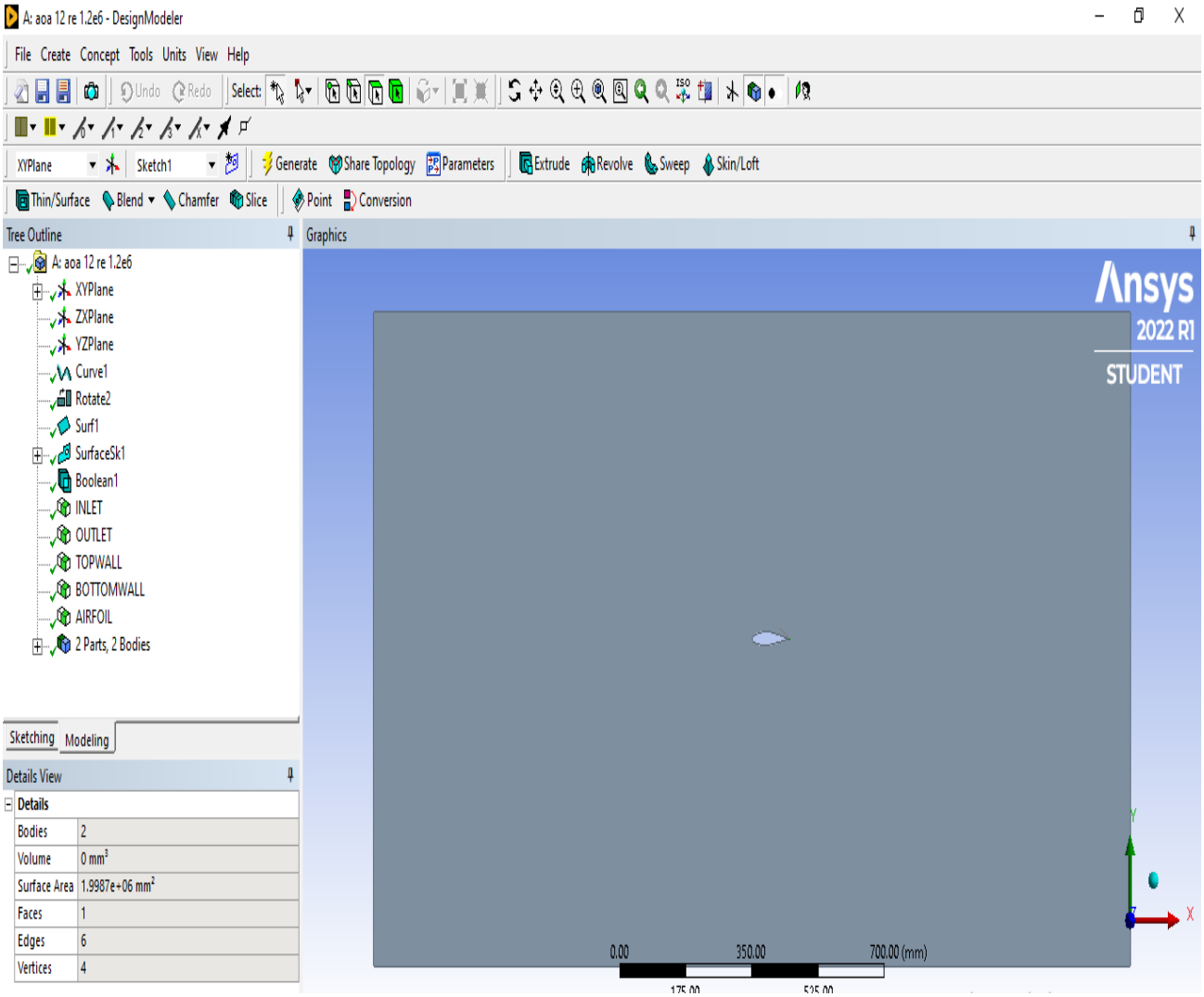

Figure* 18*: Geometry of NACA 644221

In mesh generation, we have used structured mesh. Linear element order had been used. There were five inflation layers. For the inflation, the growth rate is 1 and for the edge sizing the growth rate is 1.2. The mesh generated is shown in the figure 3.6.

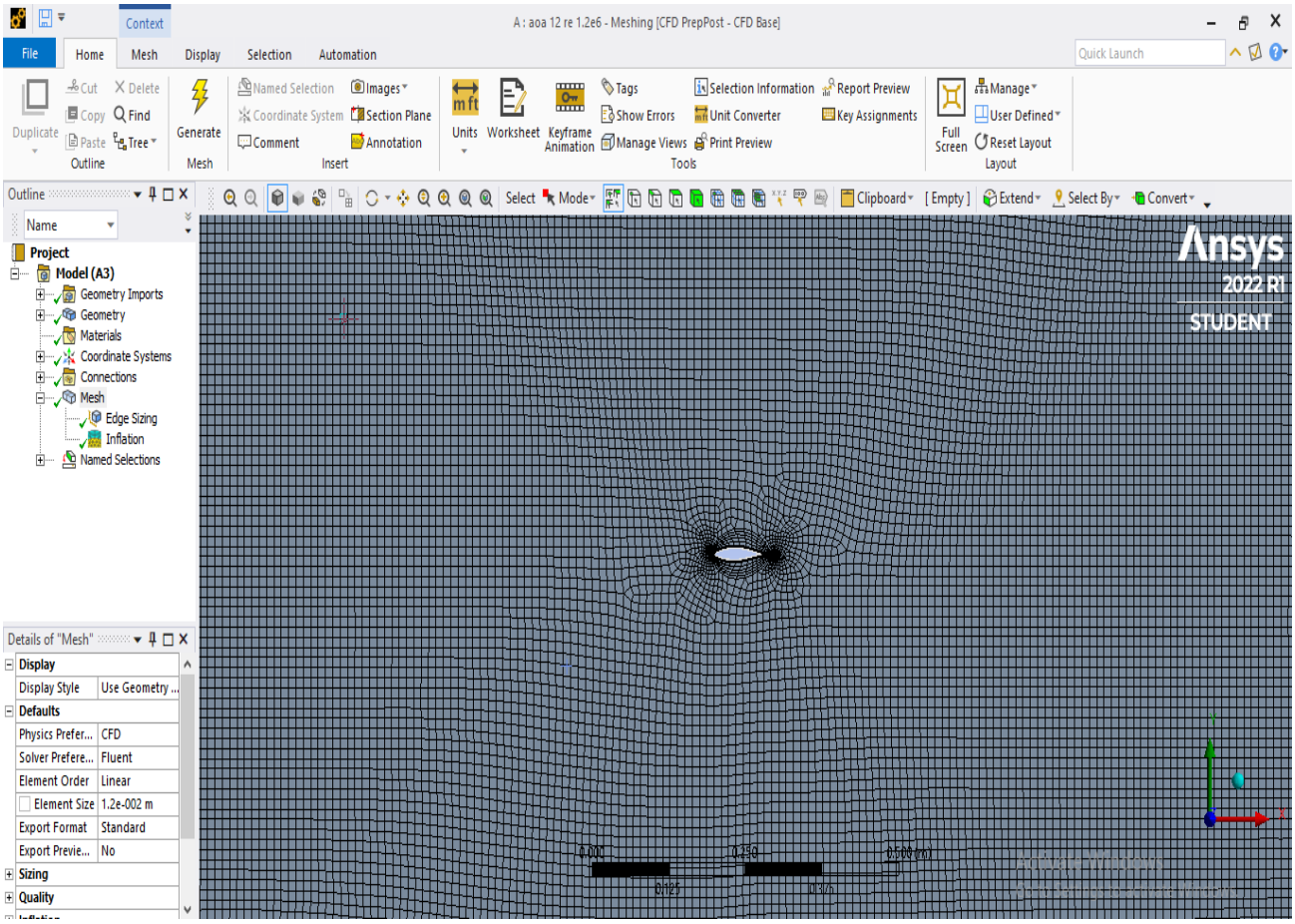

Figure* 19*: Mesh of NACA 644221

After meshing, in the solver, the material is chosen. Pressure-based solver is used along with absolute velocity formulation. Air is chosen as the flowing fluid. Spalart-Allmaras model is chosen for turbulence modelling. This is a one-equation based turbulence model for aerodynamic flows. [58] After coverging, from the post-CFD, we formatted the velocity and pressure flow field and collected the picture to train in the CNN. In the figure below, the picture for velocity profile and pressure profile are shown for NACA 644221 at AOA 0^{o} and Re 1.8×10^{6}.

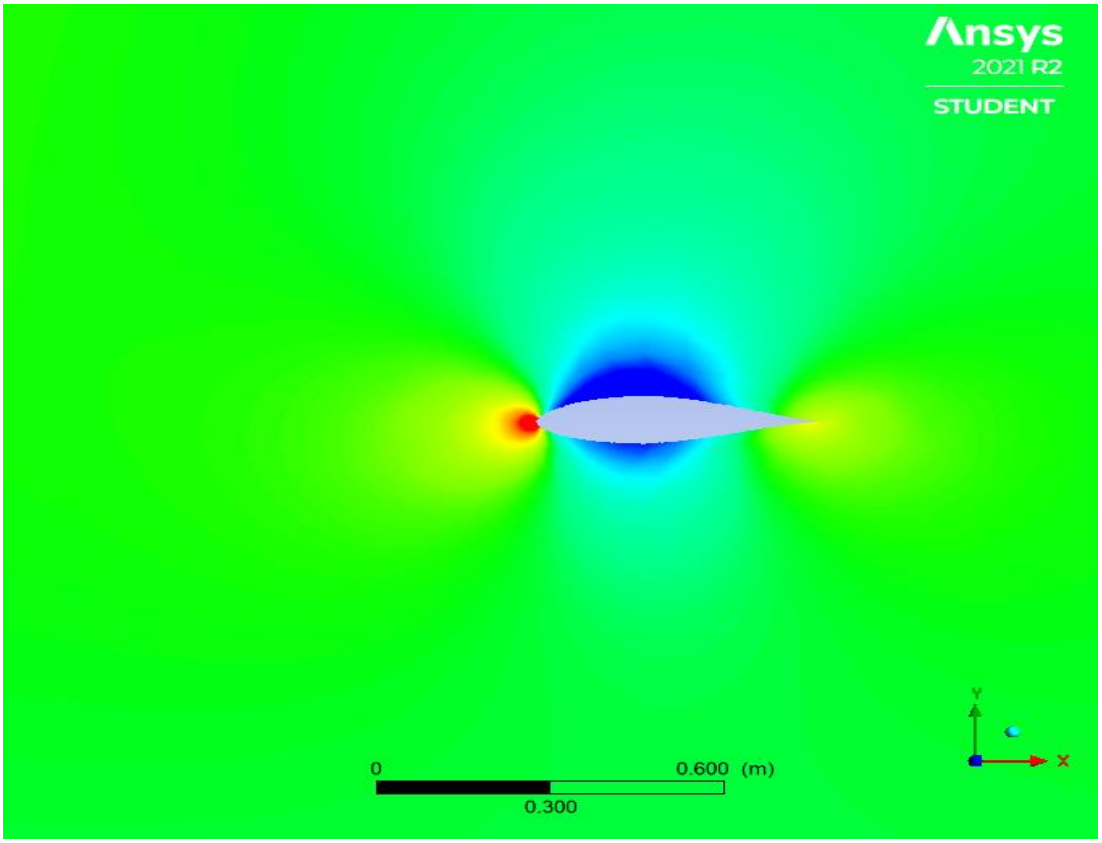

Figure* 20*: Pressure Profile of NACA 644221 at AOA: 0^o & Re: 1.8 × 10^6

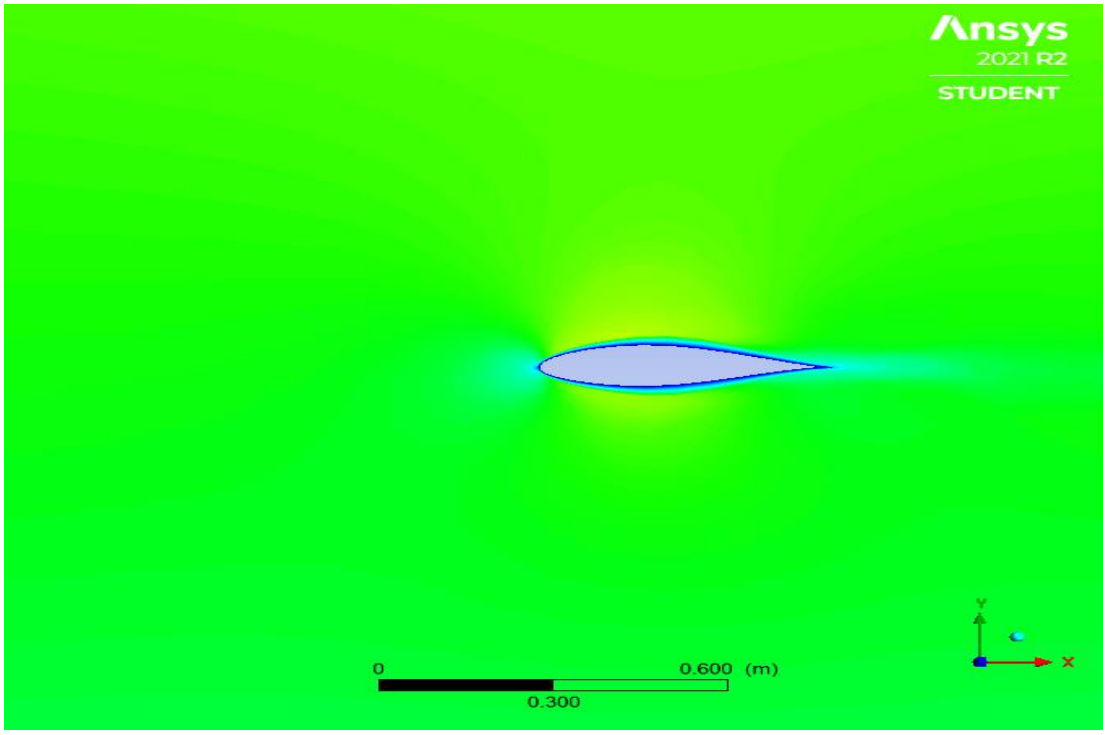

Figure* 21*: Velocity Profile of NACA 644221 at AOA: 0^o & Re: 1.8 × 10^6

3.3 Training and Validation

The data achieved from the data pre-processing is processed in this step. 80% of the data were used

for training the ML algorithm and 20% data were used for validation. The algorithm is based on U-net. U-Net is a CNN for biological image segmentation developed at the University of Freiburg's Computer Science Department. The architecture of the network was improved and enlarged to function with less training photographs and produce more precise segmentations. The basic idea is to add successive layers to a standard contractual network, with up sampling operators replacing pooling operations. As a result, the output resolution is improved by these layers. A subsequent convolutional layer may learn to create a precise output based on this information. The architecture of the ML is shown in Figure 3.9.

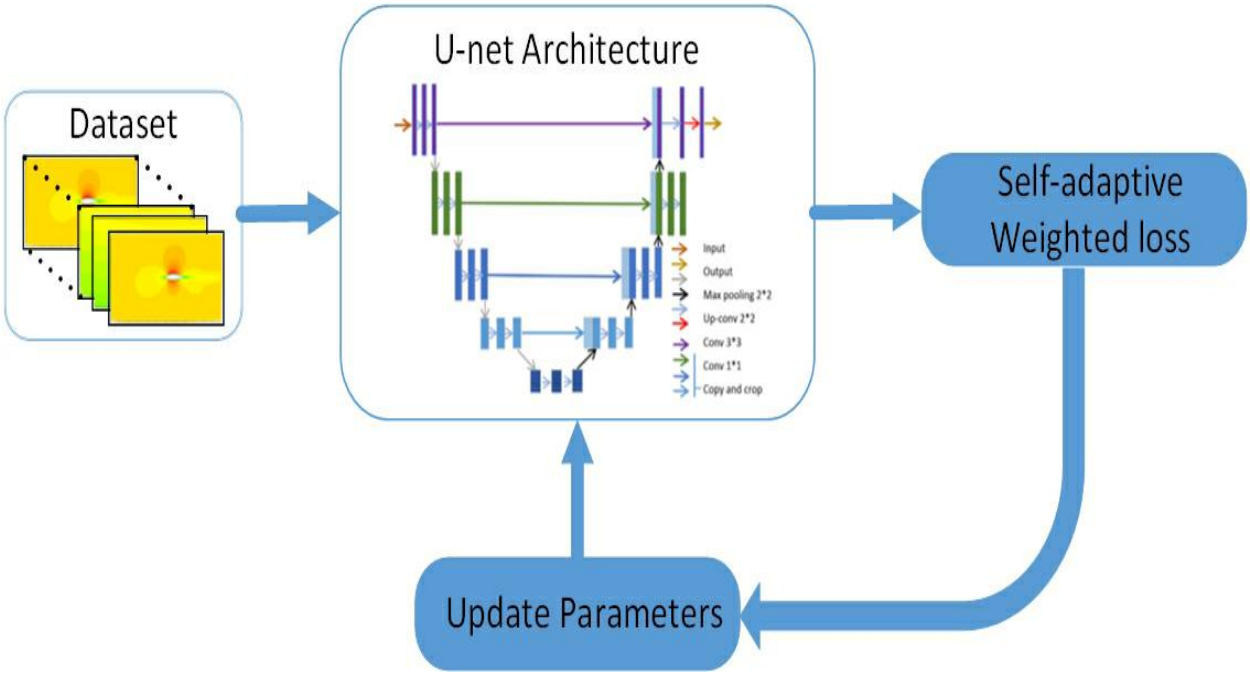

Figure 22: Machine Learning Architecture

In this process we have used U-net with attention mode. In the context of picture segmentation, attention is a method of highlighting just the important activations during training. There are two kinds of attention: hard and soft. Hard attention works by emphasizing significant portions of a picture by cropping or iterative region suggestion. Soft attention works by giving different sections of the image distinct weights. The structure of U-net with attention mode can be observed in the

Figure 3.10.

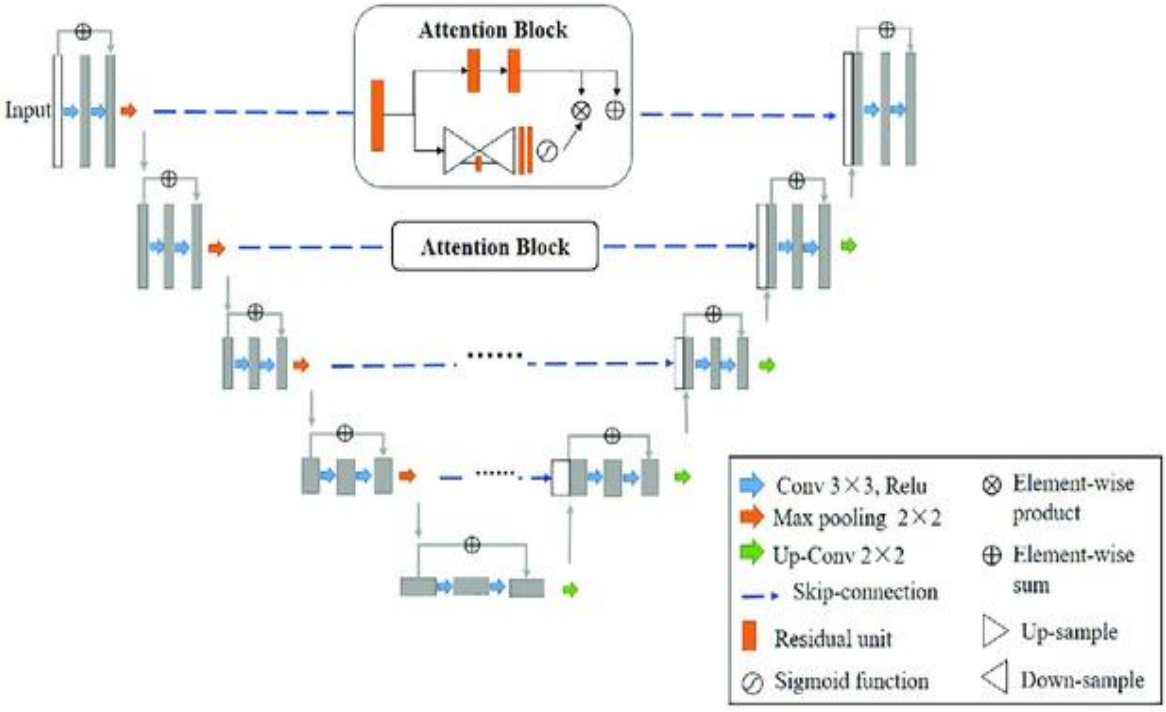

Figure 23: Architecture of U-net with attention mode [59]

Chapter 4: RESULT and DISCUSSION

4.1 Inference

This step is the post processing part. Although this is a part of the experimental methodology, the

inclusion of this part in this chapter is more relevant because in this step we deal with the results. Below is the inference flowchart. At first, we have collected data from CFD in the pre-processing part and then trained them using U-net. Then based on the results achieved, we made a comparison of the two types of data, one from CFD and other from U-net, in this step.

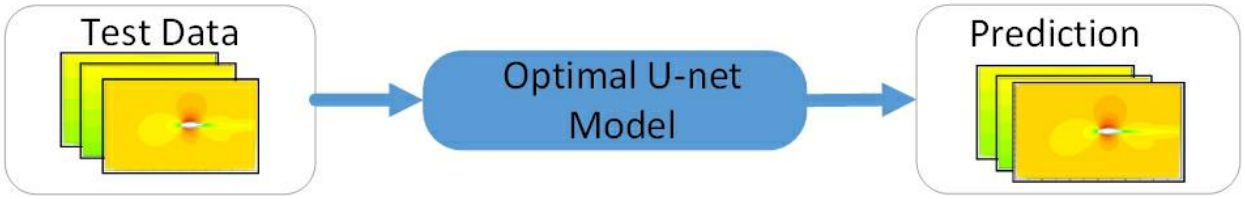

Figure 24: Inference flowchart

4.2 Results and Discussion

In this paper, we used deep learning approaches to estimate the flow field over an airfoil. The prediction strategy consists of two steps. In the first U-net (with attention model) is used phase to parameterize the model. The flow field across airfoils is predicted in the second stage. The model is trained to forecast the flow field using the parameters collected from the U-net, as well as Reynolds number, angle of attack, and x-y coordinates (pressure and velocity components. Both networks' training information and prediction results are shown in Figure 4.2-4.6. As a flow prediction network, the U-net model with convolutional layer, max-pooling as well as attention model is used. The model is trained in such a way that the training RMSE is decreased.

Case 1:

Reynolds Number: **1800000**	Angle of Attack: **0°**
Comment: hardly any visible fluctuation	

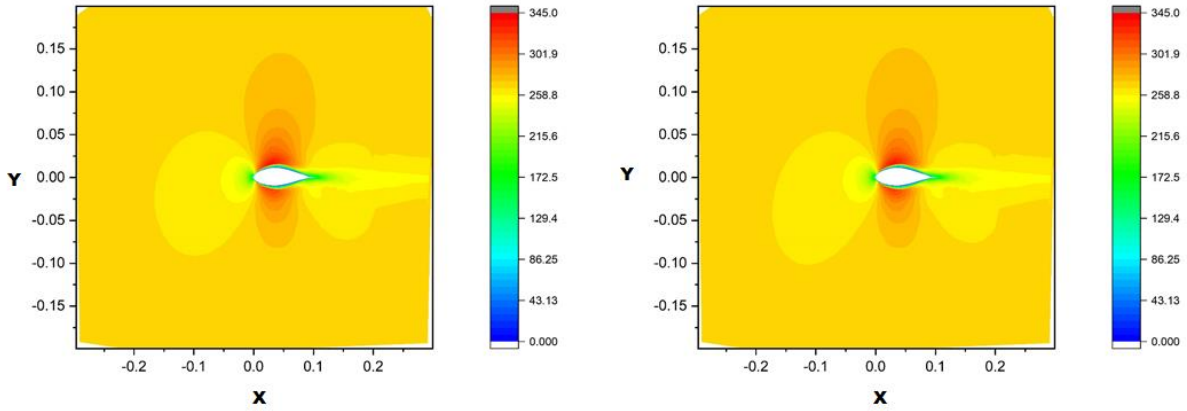

Figure* 25*: Testing airfoil NACA 63415 comparisons of the flow velocity field. From left to right, v-CFD, v-prediction (u-net)

Case 2:

Reynolds Number: **2000000**	Angle of Attack: **3°**
Comment: Visible diverge of flow regime, Acceptable hue	

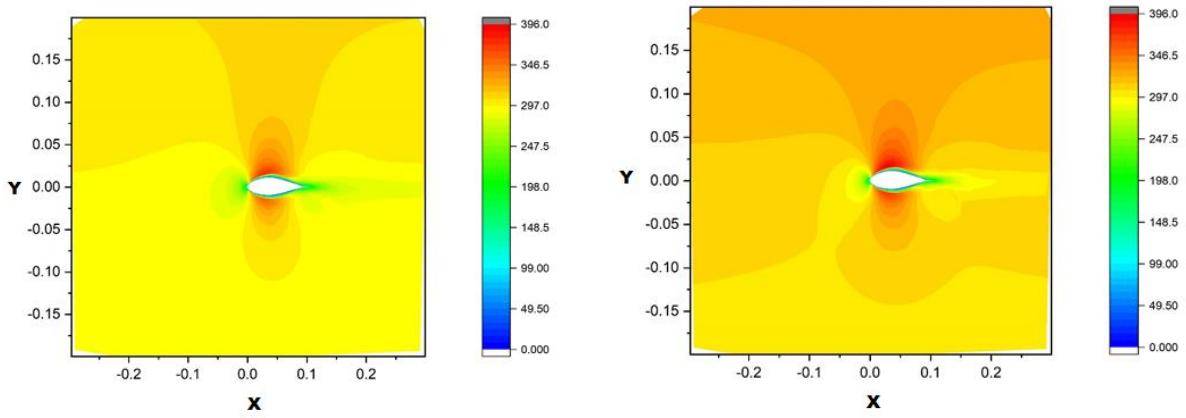

Figure 26: From left to right, v-CFD and v-prediction (u-net)

Case 3:

Reynolds Number: **1600000**	Angle of Attack: **9°**
Comment: diverge of flow regime, vague hue formation	

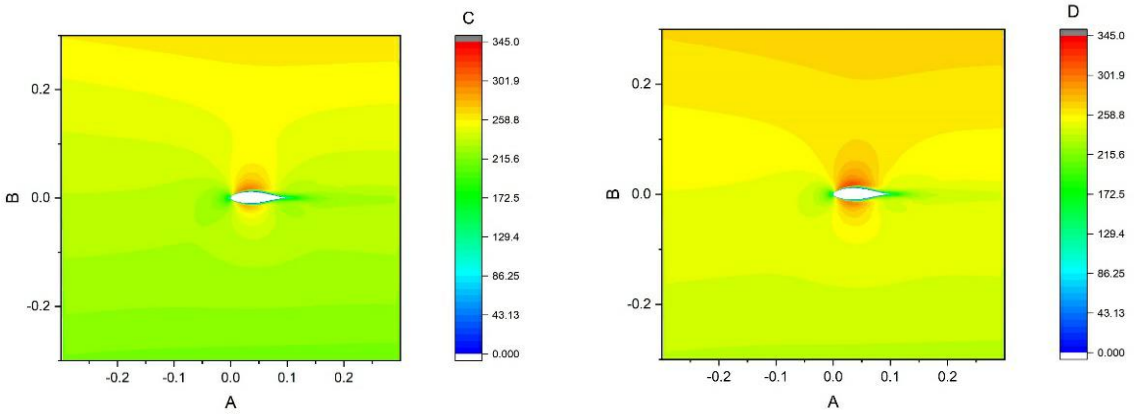

Figure 27: From left to right, v-CFD, and v-prediction (u-net)

Case 4:

Reynolds Number: **1600000**	Angle of Attack: **12°**
Comment: Prominent flow regime inaccuracy	

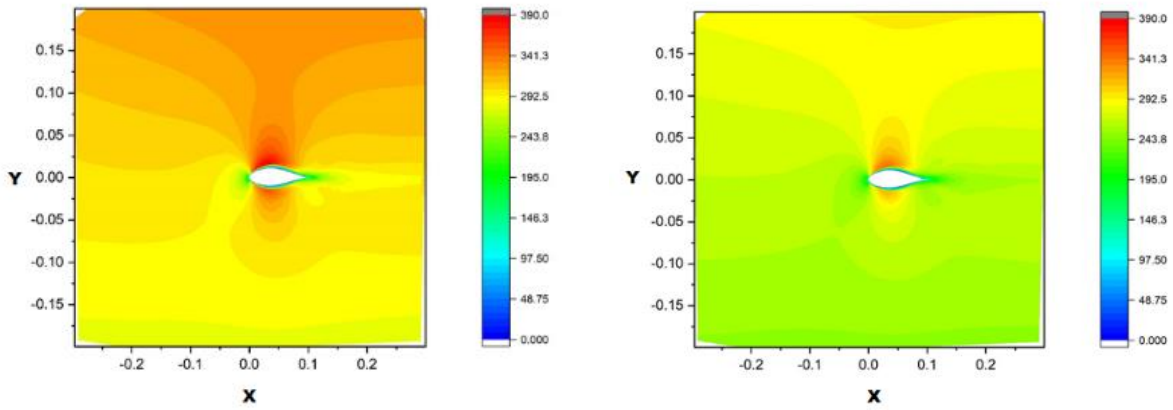

Figure 28: From left to right, v-CFD and v-prediction (u-net)

Figure 25 illustrated in Case 1 shows the flow field (velocity components) prediction results for a training airfoil case and a testing airfoil case. The anticipated flow field contour patterns closely resemble the CFD data. Figures 26 and 27 compare velocity profiles for the testing airfoils at 3^{o} and 9^{o} angles of attack in Case 2 and 3. Despite the fact that various color regimes shift, the projected velocity profiles nearly match the CFD data, as seen by the comparison. Case-4, Figure 28, in contrast to the other instances, is significantly inaccurate because to its relatively high angle of attack and high Reynolds number. As a result, projections for modest requirements fall within

an acceptable range.

The results reveal that the trained model does a good job of predicting the flow field. This also suggests that the parameterization is effective and that the u-net network can recover geometric properties from the parameters, allowing for accurate flow field prediction. Once trained, the present technique is far quicker than classic CFD algorithms, forecasting the flow field at a particular airfoil in only a few seconds. Furthermore, because to the presence of boundary conditions, CFD techniques require a fairly large flow domain to be addressed surrounding the airfoil once trained. The flow domain of interest, on the other hand, is selectable using the current approach. In most cases, such as flow around an airfoil, the flow field near the airfoil and in the wake is quite significant, as shown in this approach close to the airfoil. Furthermore, prediction is performed even when the flow field values are unknown inside the airfoil design. This might be one source of the decline in accuracy near the surface. Because airfoil coefficients are calculated from surface flow field distributions, values near the boundary are typically more significant. Overall, the suggested technique is well-suited to forecasting flow fields.

In the table below, we have shown an error assessment for case 3. The Root-Mean-Square-Deviation (RMSE), Mean Absolute Error (MAE), Mean Squared Error (MSE), standard deviation and BIAS is shown.

Error matrix	**Value**
RMSE	10.59888787
MAE	9.599160503
MSE	11.23436917
BIAS	-8.110127022
Standard deviation	4.49375656

Table 4.1: Error assessment example for case 3

Chapter 5: CONCLUSION and RECOMMENDATIONS

Although the errors of some simulation is within acceptable range, for some case the errors are substantially prominent. This may occur for various factors. The most important factor is we had sparsity of data pool. Throughout our literature review, we could not find any dataset. So, at first, we had to manually generate data, which was time consuming and tedious. Despite simulating 3325 data, we lacked the information to meet the threshold point for the training of ML algorithm.

Apart from this, the computers we had for the execution of the training and simulation, were not of compatible power. We had low configuration computers, which consumed precious time in data pre-processing, as well as, in training. Moreover, our mesh could have been finer. And lastly, the algorithm is not perfect. It needs to be optimized through trial and error, which can be a concern for further future.

Further research may allow for the reduction of the time required to expand the breadth of a CNN (convolutional neural network) model without the need to completely retrain it. The study of dissimilarity may be used to guide the selection of preprocessing strategies for ML aided turbulence modeling in order to improve prediction performance. Future work might concentrate on developing more appropriate assessment measures.

There are a number of emerging ML subjects that offer potential for CFD. Non-intrusive sensing, or the capacity to anticipate flow based on reports at the wall, is one such field. This problem, which has serious effects for closed-loop flow management, was completed using CNNs in turbulent channels.[50]

In relation to this study, a number of research have been published that demonstrate the feasibility of executing super-resolution predictions (e.g., when little flow in wall-bounded turbulence using CNNs, autoencoders, and generative adversarial networks (GANs). [60]–[63]

Another potential path is to impose restrictions on the ML model based on physical invariances and symmetries, which has been applied for SGS modeling [64], ROMs [65]and geophysical fluxes.[66]

There are also significant challenges in CFD that necessitate unique ML techniques. A fascinating problem in CFD is producing efficient coarse-resolution simulations in uncontrolled 3D wall-bounded turbulent flows. Because Turbulent Kinetic Energy (TKE) production is subject to turbulent changes in these high-tech flows, adjusting coarse and fine grades may not be sufficient to deliver correct results. These challenges will need the creation of new approaches in order to advance the profession.

Despite the limitations, we anticipate that the trend of using ML to build CFD will continue. This advancement will be fueled by a rise in the amount of high-fidelity data, massive processing power, and a full understanding of and proficiency with these successful models. It is also vital to increase the use of reproducible research standards. Given the significance of data in the development of ML modes, we recommend that the community continue to establish appropriate benchmark

mechanisms and best guidelines for accessible data and software in order to fully realize the promise of ML to enhance CFD. [67], [68].

REFERENCES

[1] S. A. Niederer, M. S. Sacks, M. Girolami, and K. Willcox, "Scaling digital twins from the artisanal to the industrial," *Nature Computational Science 2021 1:5*, vol. 1, no. 5, pp. 313–320, May 2021, doi: 10.1038/s43588-021-00072-5.

[2] M. P. Brenner, J. D. Eldredge, and J. B. Freund, "Perspective on machine learning for advancing fluid mechanics," *Physical Review Fluids*, vol. 4, no. 10, p. 100501, Oct. 2019, doi: 10.1103/PhysRevFluids.4.100501.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4_28.

[4] "Attention is All you Need." https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (accessed Jun. 04, 2022).

[5] "What is Computational Fluid Dynamics (CFD)? | SimScale | SimScale." https://www.simscale.com/docs/simwiki/cfd-computational-fluid-dynamics/what-is-cfd-computational-fluid-dynamics/ (accessed May 07, 2022).

[6] J. L. Hess and A. M. O. Smith, "Calculation of Potential Flow About Arbitrary Bodies," *Progress in Aerospace Sciences*, vol. 8, no. C, pp. 1–138, 1967, doi: 10.1016/0376-0421(67)90003-6.

[7] S. Samant *et al.*, *25th AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics (AIAA), 1987. doi: 10.2514/6.1987-34.

[8] R. Carmichael and L. Erickson, *14th Fluid and Plasma Dynamics Conference*. American Institute of Aeronautics and Astronautics (AIAA), 1981. doi: 10.2514/6.1981-1255.

[9] F. White, "Viscous Fluid Flow (McGraw-Hill Mechanical Engineering)," p. 640, 2005.

[10] V. Thomée, "From finite differences to finite elements A short history of numerical analysis of partial differential equations," *Numerical Analysis: Historical Developments in the 20th Century*, pp. 361–414, Jan. 2001, doi: 10.1016/B978-0-444-50617-7.50016-1.

[11] J. Tu, G.-H. Yeoh, and C. Liu, *Computational Fluid Dyanmics*, vol. 53, no. 9. 2013.

[12] A. Sharma, *Introduction to Computational Fluid Dynamics*. 2022. doi: 10.1007/978-3-030-72884-7.

[13] "SOLIDWORKS." https://www.solidworks.com/ (accessed May 12, 2022).

[14] "AutoCAD Software | Get Prices & Buy Official AutoCAD 2023 | Autodesk."

https://www.autodesk.com/products/autocad/overview?term=1-YEAR&tab=subscription (accessed May 12, 2022).

[15] "Introduction to Ansys DesignModeler CFD | Ansys Training." https://www.ansys.com/training-center/course-catalog/fluids/introduction-to-ansys-designmodeler (accessed May 12, 2022).

[16] "Ansys SpaceClaim | 3D CAD Modeling Software." https://www.ansys.com/products/3d-design/ansys-spaceclaim (accessed May 12, 2022).

[17] "What is a Mesh? | SimWiki Documentation | SimScale." https://www.simscale.com/docs/simwiki/preprocessing/what-is-a-mesh/ (accessed May 12, 2022).

[18] "Handbook of Grid Generation - Google Books." https://books.google.com.bd/books?hl=en&lr=&id=fxABEAAAQBAJ&oi=fnd&pg=PR3&dq=Thompson,+J.+F.,+Soni,+B.+K.,+Wheatherill,+N.+P.,+%E2%80%9CHandbook+of+Grid+Generation%E2%80%9D,+1999&ots=0VUygSSuAr&sig=Yin-XFxQV5IUBAJKXuPM6wnZ2aY&redir_esc=y#v=onepage&q&f=false (accessed May 12, 2022).

[19] "Quick overview of different 'Boundary Conditions' in CFD." https://www.manchestercfd.co.uk/post/quick-overview-of-different-boundary-conditions-in-cfd (accessed May 12, 2022).

[20] J. Gorman, S. Bhattacharyya, L. Cheng, and J. Abraham, "Turbulence Models Commonly Used in CFD," *Computational Fluid Dynamics [Working Title]*, Aug. 2021, doi: 10.5772/INTECHOPEN.99784.

[21] R. Vinuesa and S. L. Brunton, "The Potential of Machine Learning to Enhance Computational Fluid Dynamics," Oct. 2021, doi: 10.48550/arxiv.2110.02085.

[22] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *Journal of Fluid Mechanics*, vol. 807, pp. 155–166, Nov. 2016, doi: 10.1017/JFM.2016.615.

[23] K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence Modeling in the Age of Data," *https://doi.org/10.1146/annurev-fluid-010518-040547*, vol. 51, pp. 357–377, Jan. 2019, doi: 10.1146/ANNUREV-FLUID-010518-040547.

[24] S. E. Ahmed, S. Pawar, O. San, A. Rasheed, T. Iliescu, and B. R. Noack, "On closures for reduced order models—A spectrum of first-principle to machine-learned avenues," *Physics of Fluids*, vol. 33, no. 9, p. 091301, Sep. 2021, doi: 10.1063/5.0061577.

[25] O. Obiols-Sales, A. Vishnu, N. Malaya, and A. Chandramowliswharan, "CFDNet: A deep learning-based accelerator for fluid simulations," *Proceedings of the International Conference on*

Supercomputing, Jun. 2020, doi: 10.1145/3392717.3392772.

[26] J. N. Kutz, "Deep learning in fluid dynamics," *Journal of Fluid Mechanics*, vol. 814, pp. 1–4, Mar. 2017, doi: 10.1017/JFM.2016.803.

[27] J. X. Wang, J. L. Wu, and H. Xiao, "Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data," *Physical Review Fluids*, vol. 2, no. 3, p. 034603, Mar. 2017, doi: 10.1103/PHYSREVFLUIDS.2.034603/FIGURES/11/MEDIUM.

[28] C. Jiang, R. Vinuesa, R. Chen, J. Mi, S. Laima, and H. Li, "An interpretable framework of data-driven turbulence modeling using deep neural networks," *Physics of Fluids*, vol. 33, no. 5, p. 055133, May 2021, doi: 10.1063/5.0048909.

[29] J. Weatheritt and R. Sandberg, "A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship," *Journal of Computational Physics*, vol. 325, pp. 22–37, Nov. 2016, doi: 10.1016/J.JCP.2016.08.015.

[30] J. L. Wu, H. Xiao, and E. Paterson, "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework," *Physical Review Fluids*, vol. 7, no. 3, p. 074602, Jul. 2018, doi: 10.1103/PHYSREVFLUIDS.3.074602/FIGURES/16/MEDIUM.

[31] Y. Mi, M. Ishii, and L. H. Tsoukalas, "Flow regime identification methodology with neural networks and two-phase flow models," *Nuclear Engineering and Design*, vol. 204, no. 1–3, pp. 87–100, Feb. 2001, doi: 10.1016/S0029-5493(00)00325-3.

[32] A. Beck, D. Flad, and C.-D. Munz, "Deep neural networks for data-driven LES closure models," *Journal of Computational Physics*, vol. 398, p. 108910, Dec. 2019, doi: 10.1016/j.jcp.2019.108910.

[33] C. J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, and T. Poinsot, "Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates," *Combustion and Flame*, vol. 203, pp. 255–264, May 2019, doi: 10.1016/j.combustflame.2019.02.019.

[34] G. Novati, H. L. de Laroussilhe, and P. Koumoutsakos, "Automating turbulence modelling by multi-agent reinforcement learning," *Nature Machine Intelligence*, vol. 3, no. 1, pp. 87–96, Jan. 2021, doi: 10.1038/s42256-020-00272-0.

[35] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, Apr. 2016, doi: 10.1016/J.NEUCOM.2015.09.116.

[36] M. Ma, J. Lu, and G. Tryggvason, "Using statistical learning to close two-fluid multiphase flow equations for a simple bubbly system," *Physics of Fluids*, vol. 27, no. 9, p. 092101, Sep. 2015, doi: 10.1063/1.4930004.

[37] F. Gibou, D. Hyde, and R. Fedkiw, “Sharp interface approaches and deep learning techniques for multiphase flows,” *Journal of Computational Physics*, vol. 380, pp. 442–463, Mar. 2019, doi: 10.1016/J.JCP.2018.05.031.

[38] S. N. Skinner and H. Zare-Behtash, “State-of-the-art in aerodynamic shape optimisation methods,” *Applied Soft Computing*, vol. 62, pp. 933–962, Jan. 2018, doi: 10.1016/J.ASOC.2017.09.030.

[39] Y. Bar-Sinai, S. Hoyer, J. Hickey, and M. P. Brenner, “Learning data-driven discretizations for partial differential equations,” *Proc Natl Acad Sci U S A*, vol. 116, no. 31, pp. 15344–15349, Jul. 2019, doi: 10.1073/PNAS.1814058116.

[40] J. Jeon and S. J. Kim, “FVM Network to Reduce Computational Cost of CFD Simulation,” *International Journal of Energy Research*, May 2021, doi: 10.1002/er.7879.

[41] B. Stevens and T. Colonius, “FiniteNet: A Fully Convolutional LSTM Network Architecture for Time-Dependent Partial Differential Equations,” Feb. 2020, doi: 10.48550/arxiv.2002.03014.

[42] K. Fukami, Y. Nabae, K. Kawai, and K. Fukagata, “Synthetic turbulent inflow generator using machine learning,” *Physical Review Fluids*, vol. 4, no. 6, p. 064603, Jun. 2019, doi: 10.1103/PHYSREVFLUIDS.4.064603/FIGURES/11/MEDIUM.

[43] Y. Morita, S. Rezaeiravesh, N. Tabatabaei, R. Vinuesa, K. Fukagata, and P. Schlatter, “Applying Bayesian optimization with Gaussian process regression to computational fluid dynamics problems,” *Journal of Computational Physics*, vol. 449, p. 110788, Jan. 2022, doi: 10.1016/J.JCP.2021.110788.

[44] W. Tang *et al.*, “Study on a Poisson’s equation solver based on deep learning technique,” *2017 IEEE Electrical Design of Advanced Packaging and Systems Symposium, EDAPS 2017*, vol. 2018-January, pp. 1–3, Jan. 2018, doi: 10.1109/EDAPS.2017.8277017.

[45] A. G. Ozbay, A. Hamzehloo, S. Laizet, P. Tzirakis, G. Rizos, and B. Schuller, “Poisson CNN: Convolutional neural networks for the solution of the Poisson equation on a Cartesian mesh,” *Data-Centric Engineering*, vol. 2, no. 6, Jun. 2021, doi: 10.1017/DCE.2021.7.

[46] R. Vinuesa, S. M. Hosseini, A. Hanifi, D. S. Henningson, and P. Schlatter, “Pressure-Gradient Turbulent Boundary Layers Developing Around a Wing Section,” *Flow, Turbulence and Combustion*, vol. 99, no. 3–4, pp. 613–641, Dec. 2017, doi: 10.1007/S10494-017-9840-Z.

[47] G. Novati, H. L. de Laroussilhe, and P. Koumoutsakos, “Automating turbulence modelling by multi-agent reinforcement learning,” *Nature Machine Intelligence 2021 3:1*, vol. 3, no. 1, pp. 87–96, Jan. 2021, doi: 10.1038/s42256-020-00272-0.

[48] R. Vinuesa and S. L. Brunton, “The Potential of Machine Learning to Enhance Computational Fluid Dynamics,” pp. 1–13, 2021, [Online]. Available: http://arxiv.org/abs/2110.02085

[49] H. Eivazi, S. le Clainche, S. Hoyas, and R. Vinuesa, “Towards extraction of orthogonal and parsimonious non-linear modes from turbulent flows,” Sep. 2021, doi: 10.48550/arxiv.2109.01514.

[50] L. Guastoni *et al.*, “Convolutional-network models to predict wall-bounded turbulence from wall quantities,” *Journal of Fluid Mechanics*, vol. 928, Dec. 2021, doi: 10.1017/JFM.2021.812.

[51] Y. Bar-Sinai, S. Hoyer, J. Hickey, and M. P. Brenner, “Learning data-driven discretizations for partial differential equations,” *Proc Natl Acad Sci U S A*, vol. 116, no. 31, pp. 15344–15349, Jul. 2019, doi: 10.1073/PNAS.1814058116.

[52] B. Stevens and T. Colonius, “Enhancement of shock-capturing methods via machine learning,” *Theoretical and Computational Fluid Dynamics*, vol. 34, no. 4, pp. 483–496, Aug. 2020, doi: 10.1007/S00162-020-00531-1/FIGURES/12.

[53] C. W. Rowley and S. T. M. Dawson, “Model Reduction for Flow Analysis and Control,” *http://dx.doi.org/10.1146/annurev-fluid-010816-060042*, vol. 49, pp. 387–417, Jan. 2017, doi: 10.1146/ANNUREV-FLUID-010816-060042.

[54] K. Taira *et al.*, “Modal analysis of fluid flows: Applications and outlook,” *AIAA Journal*, vol. 58, no. 3, pp. 998–1022, Oct. 2020, doi: 10.2514/1.J058462/ASSET/IMAGES/LARGE/FIGURE19.JPEG.

[55] “The structure of inhomogeneous turbulent flows | CiNii Research.” https://cir.nii.ac.jp/crid/1571980075051475712 (accessed May 14, 2022).

[56] V. Sekar, Q. Jiang, C. Shu, and B. C. Khoo, “Fast flow field prediction over airfoils using deep learning approach,” *Physics of Fluids*, vol. 31, no. 5, p. 057103, May 2019, doi: 10.1063/1.5094943.

[57] “Airfoil data information.” http://airfoiltools.com/airfoil/ (accessed May 11, 2022).

[58] P. R. Spalart and S. R. Allmaras, “One-equation turbulence model for aerodynamic flows,” *Recherche aerospatiale*, no. 1, pp. 5–21, 1994, doi: 10.2514/6.1992-439.

[59] S. M. Nazia Fathima, R. Tamilselvi, M. Parisa Beham, and D. Sabarinathan, “Diagnosis of Osteoporosis using modified U-net architecture with attention unit in DEXA and X-ray images,” *Journal of X-Ray Science and Technology*, vol. 28, no. 5, pp. 953–973, Jan. 2020, doi: 10.3233/XST-200692.

[60] H. Kim, J. Kim, S. Won, and C. Lee, “Unsupervised deep learning for super-resolution reconstruction of turbulence,” *Journal of Fluid Mechanics*, vol. 910, 2021, doi: 10.1017/JFM.2020.1028.

[61] A. Güemes, S. Discetti, A. Ianiro, B. Sirmacek, H. Azizpour, and R. Vinuesa, “From coarse wall measurements to turbulent velocity fields through deep learning,” *Physics of Fluids*, vol. 33, no.

7, p. 075121, Jul. 2021, doi: 10.1063/5.0058346.

[62] K. Fukami, T. Nakamura, and K. Fukagata, “Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data,” *Physics of Fluids*, vol. 32, no. 9, p. 095110, Sep. 2020, doi: 10.1063/5.0020721.

[63] “The structure of inhomogeneous turbulent flows | CiNii Research.” https://cir.nii.ac.jp/crid/1571980075051475712 (accessed May 14, 2022).

[64] R. Wang, R. Walters, and R. Yu, “Incorporating Symmetry into Deep Dynamics Models for Improved Generalization,” Feb. 2020, doi: 10.48550/arxiv.2002.03061.

[65] J. C. Loiseau and S. L. Brunton, “Constrained sparse Galerkin regression,” *Journal of Fluid Mechanics*, vol. 838, pp. 42–67, Mar. 2018, doi: 10.1017/JFM.2017.823.

[66] H. Frezat, G. Balarac, J. le Sommer, R. Fablet, and R. Lguensat, “Physical invariance in neural networks for subgrid-scale scalar flux modeling,” *Physical Review Fluids*, vol. 6, no. 2, p. 024607, Feb. 2021, doi: 10.1103/PHYSREVFLUIDS.6.024607/FIGURES/17/MEDIUM.

[67] O. Mesnard and L. A. Barba, “Reproducible and Replicable Computational Fluid Dynamics: It’s Harder Than You Think,” *Computing in Science and Engineering*, vol. 19, no. 4, pp. 44–55, 2017, doi: 10.1109/MCSE.2017.3151254.

[68] L. A. Barba, “The hard road to reproducibility,” *Science (1979)*, vol. 354, no. 6308, p. 142, Oct. 2016, doi: 10.1126/SCIENCE.354.6308.142/ASSET/35E106D4-FB4C-41F4-A7AE-EDEFCD081B90/ASSETS/GRAPHIC/354_142_F1.JPEG.