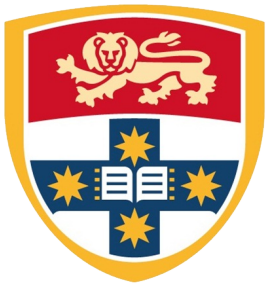
Reynolds-Averaged Turbulence Closures Using Machine Learning to Predict Open Channel Flow



THE UNIVERSITY OF SYDNEY

Rhys T. Kilian

Supervisor: Assoc. Prof. Michael Kirkpatrick

School of Aerospace, Mechanical and Mechatronic Engineering
University of Sydney

This dissertation is submitted for the degree of
Bachelor of Engineering (Mechanical)

November 2018

Abstract

Traditionally, simulating turbulent flow requires extensive computational resources to produce high-quality results using Direct Numerical Simulation (DNS) or Large Eddy Simulation (LES). While these methods yield highly accurate predictions for quantities such as the velocity of the flow, they are computationally infeasible for industrial applications. Instead, lower-fidelity Reynolds Averaged Navier-Stokes (RANS) models, such as $k-\varepsilon$, provide a compromise between accuracy and speed. However, it is well understood that these RANS models are inaccurate on flows with separation, curvature, and large pressure and buoyancy gradients [8].

Therefore, the goal of this thesis is to develop a proof-of-concept RANS turbulence model closure that would overcome the shortcomings of traditional turbulence models. Instead of using an experimental approach to inform the new turbulence model, a data-driven approach was adopted by using statistical techniques such as machine learning. The developed model proposed in this thesis is based on an Explicit Algebraic Stress (EASM) formulation of RANS, which calculates the anisotropic Reynolds stress.

The machine learning model is trained on LES data for both isothermal and stratified open channel flow. The proposed turbulence model uses a neural network, which underwent hyperparameter optimisation to form what is known as a Tensor Basis Neural Network (TBNN). This TBNN was evaluated based on three factors: accuracy, generalisability and interpretability. It is concluded that the TBNN made more accurate predictions for the simpler isothermal flow case, however, even the stratified predictions are said to be more accurate than the predictions which would be made using traditional RANS models.

Moreover, statistical inference shows the TBNN to be generalisable, with reliable average predictions for anisotropic Reynolds stress possible. Finally, the quality of the input variables is measured using Partial Dependence Plots (PDP). Here, it is found that the conditions for Galilean invariance are met and that a wall-based Reynolds number and a variable measuring the stratification effects are essential to making accurate predictions. While this study confirms that a data-driven turbulence model is viable, further variable selection and a posterior study to predict the fluid velocity is required.

Statement of Contribution

- I worked with Assoc. Prof. Michael Kirkpatrick to develop the objectives of this thesis.
- I carried out the literature review to develop a more in-depth understanding of previous data-driven turbulence models.
- I ran all the Large Eddy Simulations on PUFFIN to generate the data required.
- I completed all the data pre-processing to create the input variables for my machine learning model by writing a Python script.
- I conducted all the Exploratory Data Analysis, including the flow visualisations in ParaView and input variable assessment, using R.
- I developed and optimised the multi-input, multi-output neural network in the Python package known as keras.
- I wrote a Python script to visualise the output of my machine learning model.
- I performed the statistical analysis of my results, including writing my own Python script to create Partial Dependence Plots.
- I wrote this thesis, where the analysis, conclusions and recommendations are my own work, informed by my discussions with my supervisor.

The above points represent an accurate summary of the student's contribution.

Rhys T. KILIAN	Assoc. Prof. Michael KIRKPATRICK

Acknowledgements

First and foremost, I would like to thank my supervisor, Michael Kirkpatrick, for all his time, technical expertise, calming words and humour throughout the year. Without his help, I would still be running CFD simulations. I would also like to express how much of an impact he has had on me throughout my time at university, with his inspiring lectures leading me to where I am today.

I am also grateful for all the love and support my family have provided me, not only through thesis, but also throughout my last five years of university. To my mum, Alison, thank you for being my number one supporter in life. Your encouragement and constant questioning of when my thesis is due is something that I will not forget. To my sister, Amy, I'm still not sure you even know what my thesis topic is about, let alone what machine learning is. However, your Snapchats have always been a source of laughter when I was stuck writing my thesis. Finally, to my grandma, Beverley, thank you for always taking interest in my thesis, even though I am one of dozens of descendants you have. I also could not have completed my degree without your constant supply of desserts.

I could not have done this thesis without my loving girlfriend, Tessa. Thank you for listening to my nightly rants about thesis and university in general. I have always appreciated your words of encouragement and support when I was stressed, even though I was pretending I wasn't.

To all my friends, you have been a great help throughout the year. A big thank you to Emelia for your thesis edits, advice and coffee, which I definitely needed at times. Thank you to my roommate, Tim, for always being interested in machine learning and doing the chores around that house when I was too busy writing. Finally, I need to mention the dynamic duo of Kacy and Carl. Thank you for always getting me out of the house to exercise and supplying me with a lovely couch that I could work on.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Overview

Recent computational advances have enabled high-fidelity simulations for turbulent flows, such as Direct Numerical Simulation (DNS) and Large Eddy Simulations (LES), to become viable. However, lower fidelity Reynolds Averaged Navier-Stokes (RANS) models are still the primary tool for modelling turbulence in commercial applications [54]. Yet, commonly used RANS models, such as the $k-\varepsilon$ and $k-\omega$ methods, are known to be inaccurate on flows with three-dimensional separation, streamline curvature and large pressure and buoyancy gradients [9]. The predictive capabilities of RANS models are somewhat limited due to the assumptions made when providing a closure model for the Reynolds stress which does not have a closed form, explicit solution. The resulting discrepancy between the estimated Reynolds stress and the true Reynolds stress leads to inaccurate predictions for quantities such as the mean velocity, mean pressure, wall shear and the lift and drag forces [76].

Recently, there has been interest in providing data-driven closure models for the Reynolds stress to improve the accuracy of RANS models. Machine learning techniques, such as genetic algorithms, deep neural networks and random forests, can be used to model the Reynolds stress using data obtained from high-fidelity sources, such as DNS and LES. Studies conducted by Ling et al. [34], Wang et al. [65] and Weatheritt and Sandberg [72] have shown that data-driven closure models can improve the accuracy of traditional RANS models in a number of different applications including duct, periodic hills and open channel flow. Despite this, there is no clear consensus about how to best develop data-driven RANS closures, as considerations must be made regarding: what machine learning algorithm to use, what variables should be inputted into the algorithm, how to enforce Galilean invariance, and how to best structure the representation of the Reynolds stress.

The chosen application for this thesis is modelling open channel flow under both isothermal and stratified conditions. Stratified open channels have traditionally been used as models for river and estuarine flow where a constant heat flux is applied to the top of the channel, to mimic incident solar radiation from the Sun. This results in stratification where the temperature of the water changes at different levels of the open channel, resulting in buoyancy gradients which cannot be accurately modelled by RANS models [8]. While high-fidelity methods such as DNS can be used, the computational requirements for this turbulence modelling method are prohibitive in many situations. Therefore, there exists a need to develop an accurate, data-driven RANS model to gain a better understanding of the stratification effects since these effects result in a decrease in oxygen and nutrient transport, reducing the quality of Australian waterways [63].

1.2 Summary of Objectives

The first objective is to develop a data-driven closure model for the RANS equations which is accurate for both isothermal and stratified open channel flow conditions. This will be achieved using the Explicit Algebraic Stress Model (EASM), where the closure of the Reynolds stress could theoretically be inserted into a RANS solver. This data-driven closure should be more accurate than traditional closure models, yet provide an effective trade-off in terms of computational requirements when compared to high-fidelity methods such as DNS and LES. Further considerations must be taken into account regarding which machine learning algorithm to use, what input variables to select and how to ensure that a consistent result is obtained, independent of the coordinate system selected (i.e. Galilean invariance).

The second objective is to analyse the generalisability of such a data-driven closure model to a variety of flow conditions. This includes testing the accuracy of the turbulence model on both isothermal and stratified open channel flow. The influence of the change in fluid buoyancy, as a result of the stratification effects, will be assessed. If it is found that the data-driven closure model developed for the isothermal case is not generalisable, additional heat terms will be included to model the thermal stratification process. Furthermore, statistical inference techniques will provide an insight into the reliability of the anisotropic Reynolds stress predictions.

Finally, determining the interpretability of the turbulence model developed using the machine learning algorithm will be important to identify the most important input variables. As such, the complex interactions between the turbulent flow and stratified effects can be better understood, as well as providing a method to verify whether Galilean invariance has been enforced.

1.3 Thesis Structure

This thesis is structured into the following chapters:

- **Chapter 1**: Outline the motivation and objectives of this thesis.
- **Chapter 2**: Review of literature in the following areas.
  - Background theory for turbulence modelling.
  - Statistical methods used to quantify uncertainty in turbulence models.
  - Machine learning techniques applied to alter and develop new turbulence closures.
- **Chapter 3**: Description of the data-driven turbulence modelling framework implemented.
- **Chapter 4**: Presentation and analysis of model development results.
- **Chapter 5**: Discussion of methods used and results obtained.
- **Chapter 6**: Conclusions and future work.

Chapter 2

Background and Literature Review

2.1 Turbulence Modelling

Turbulence is a highly irregular state of fluid which is common in nature and various engineering applications. Turbulent flow is three dimensional with turbulent eddies mixing with one another, transferring heat and mass with their motion. The scale at which these eddies exist is varied, throughout time and space, making it a challenging engineering problem. However, turbulent flows are statistically determinant in the sense that the mean velocity can be found, hence making flow predictions possible. Turbulent flows can also be represented as a set of equations, known as the Navier-Stokes equations. The work of Pope [49] provides a more complete introduction of turbulence modelling.

2.1.1 Direct Numerical Simulation (DNS)

Turbulence modelling is a branch of Computational Fluid Dynamics (CFD). Using the finite volume method, turbulence is represented as a numerical simulation by dividing the fluid body into discrete control volumes and resolving the governing partial differential equations, in conservation form, over this domain [49]. The Navier-Stokes equations, which govern the motion for incompressible, Newtonian fluids can be written in tensor notation, where Einstein convention of summing over repeated indices is used, as per Equations (2.1) and (2.2).

$$\partial_\alpha u_\alpha = 0 \tag{2.1}$$

$$\partial_t u_\alpha + \partial_\beta u_\alpha u_\beta = \frac{-1}{\rho}\partial_\alpha P + \nu \partial_\beta \partial_\beta u_\alpha \tag{2.2}$$

In these equations, the term $\partial_\alpha u_\alpha$ represents the partial derivative of the velocity component, $\partial_t u_\alpha$ is the partial derivative of the velocity component with respect to time, ρ is the fluid's

density, P is the pressure and ν is the kinematic viscosity.

Nonetheless, Direct Numerical Simulation (DNS) is a method that directly solves the Navier-Stokes equations. DNS requires considerable computational power as a range of turbulent scales are required to be solved. Two metrics which are often used to quantify the scale of turbulent flow are the Kolmogorov [26] time and length scales, represented in Equations (2.3) and (2.4) respectively,

$$l_k = \left(\frac{\nu^3}{\varepsilon}\right)^{\frac{1}{4}}, \tag{2.3}$$

$$\tau_k = \left(\frac{\nu}{\varepsilon}\right)^{\frac{1}{2}}, \tag{2.4}$$

with the turbulent kinetic energy dissipation rate, ε, being defined as per Equation (2.5),

$$\varepsilon = \nu\overline{(\partial_\alpha u'_\beta)(\partial_\alpha u'_\beta)}. \tag{2.5}$$

Here, the overbar operator $\overline{(\ )}$ represents the averaging of the velocity fluctuations. Therefore, u'_β represents the fluctuating component of the velocity, where $u'_\beta = u_\beta - \bar{u}_\beta$, with $\bar{u}_\beta$ being the average velocity for that given direction.

Furthermore, Equations (2.3) and (2.4) can be non-dimensionalised,

$$l_t = \frac{k^{3/2}}{\varepsilon}, \tag{2.6}$$

$$\tau_t = \frac{k}{\varepsilon}, \tag{2.7}$$

with the turbulent kinetic energy, k, being defined by Equation (2.8),

$$k = \frac{1}{2}\overline{(u'_\alpha u'_\alpha)}. \tag{2.8}$$

Therefore, to accurately account for these small scales in the DNS method, the grid spacing Δ must be on the order of magnitude of l_t and the time step, Δt, must be on the order of magnitude of τ_t. In fact, Sandberg and Coleman [52] showed that,

$$\frac{l_t}{l_k} \approx Re^{\frac{3}{4}}, \tag{2.9}$$

where Re is the Reynolds number, which is the ratio of inertial forces to viscous forces, given in the general form in Equation (2.10),

$$Re = \frac{uL}{\nu}, \tag{2.10}$$

where L is the characteristic dimension.

Therefore, the number of grid points in one direction, such as the x-coordinate direction, is proportional to $Re^{3/4}$ (i.e. $N_x \propto Re^{3/4}$) as per Equation (2.9). Hence, when considering a three-dimensional turbulent simulation, the number of grid points is,

$$N_{xyz} \propto Re^{\frac{9}{4}}. \tag{2.11}$$

Moreover, the same exercise can be repeated to find the minimum number of time steps necessary to capture the time scale of the small eddies [52],

$$N_t \approx \frac{\tau_t}{\Delta t} \approx \frac{\tau_t}{l_k/U_0} \approx \frac{\tau_t}{,} l_t/U_0 Re^{3/4}, \tag{2.12}$$

where U_0 is defined as a large-scale reference velocity. Again, the number of time steps required is proportional to $Re^{3/4}$. As such, Equations (2.11) and (2.12) can be combined to find an estimate of the overall complexity of the turbulent scales,

$$\text{complexity} \propto N_{xyz} N_t \propto Re^3. \tag{2.13}$$

Hence, even for moderate Reynolds values, the number of grid points required to model the small turbulent scales can become prohibitively expensive. For this reason, DNS is rarely used in industrial applications where turbulent flow is present. For instance, Spalart et al. [57] estimated that a DNS of a commercial aeroplane wing would require 10^{16} points, a number which is infeasible with today's technology. This highlights the severe limitations of the DNS method and the requirements for computationally simpler models.

2.1.2 Large Eddy Simulation (LES)

To overcome the shortcomings of the DNS method a scale operator can be applied which determines what scales of turbulent eddies should be resolved and which should be modelled. This results in a significant reduction in computational cost in comparison to DNS while maintaining its high-fidelity nature. For this reason, Large Eddy Simulation (LES) is commonly used in practice and has been used in many papers to simulate the flow of thermally stratified river channels [2, 60, 68].

PUFFIN, the CFD program used to conduct the simulations in this study, solves the Navier-Stokes equations in spatially filtered Boussinesq form [25], as per Equations (2.14) and (2.15).

$$\partial_\alpha \bar{u}_\alpha = 0 \tag{2.14}$$

$$\partial_t \bar{u}_\alpha + \partial_\beta \bar{u}_\alpha \bar{u}_\beta = \frac{-1}{\rho_{ref}} \partial_\alpha \bar{P} + \frac{(\rho - \rho_{ref})}{\rho_{ref}} g_\alpha + \partial_\beta (2\nu \bar{S}_{\alpha\beta} - \partial_\beta \tau_{\alpha\beta}) \tag{2.15}$$

Equation (2.15) makes use of the ρ_{ref} term which is a fixed density reference. This is based on the assumption that density variations within the fluid are negligible for all terms, except for buoyancy term [19]. Unlike in the DNS equations, the the overbar ($\bar{}$) in Equations (2.14) and (2.15) represents the spacial filtering operator. The term $\bar{S}_{\alpha\beta}$ is the large-scale strain rate tensor given by Equation (2.16),

$$\bar{S}_{\alpha\beta} = \frac{1}{2}(\partial_\beta \bar{u}_\alpha + \partial_\alpha \bar{u}_\beta). \tag{2.16}$$

The term, $\tau_{\alpha\beta}$, in Equation (2.15), is known as the Sub-Grid Scale (SGS) stress tensor and is shown in Equation (2.17).

$$\tau_{\alpha\beta} = \overline{u_\alpha u_\beta} - \bar{u}_\alpha \bar{u}_\beta \tag{2.17}$$

However, $\tau_{\alpha\beta}$ can not be directly calculated, so instead Equation (2.17) can be modelled using Equation (2.18) which is the anisotropic component of the momentum flux [25].

$$\tau_{\alpha\beta} - \frac{1}{3}\delta_{\alpha\beta}\tau_{\gamma\gamma} = -2\nu_t \bar{S}_{\alpha\beta} \tag{2.18}$$

However, to determine the stress associated with the SGS eddies in Equation (2.18) the turbulent eddy viscosity, ν_t, is required to be found. Like the RANS method, there are a number of turbulence models which can be used to determine this term, where this study uses the dynamic Smagorinsky LES model.

Dynamic Smagorinsky Model

The dynamic Smagorinsky LES model [55] is an alteration of the standard model, where Equation (2.19) models ν_t, from the previous equation as,

$$\nu_t = (C_s \Delta)^2 |\bar{S}|. \tag{2.19}$$

In the above equation C_s is a dimensionless model coefficient, Δ is the scale threshold at which the model eddy filter is applied, and $|\bar{S}|$ is the filtered shear strain rate given by Equation (2.20),

$$|\bar{S}| = \sqrt{2\bar{S}_{\alpha\beta}\bar{S}_{\alpha\beta}}. \tag{2.20}$$

However, to overcome the shortcomings of having fixed coefficients in the standard model, these can be updated to give improved accuracy. The dynamic model allows for the coefficients to be updated as time progresses as detailed by Germano et al. [18]. Full details on the implementation of this model in PUFFIN can be found in the PUFFIN user manual [25].

2.1.3 Reynolds-Averaged Navier-Stokes (RANS)

The LES method is still computationally expensive, and as such, the Reynolds-Averaged Navier-Stokes (RANS) method is still the dominant CFD tool for commercially relevant turbulent flows [54]. RANS is favoured as the statistical averaging approach of this method is much cheaper than the scale separation process of LES. However, in this sense, the statistical averaging approach of turbulence results in modelling, rather than simulation, with only the mean flow being resolved. The RANS equations, giving the mean velocity and pressure fields, are almost identical to Equation (2.2), however, with the addition of the Reynolds stress term, $\tau_{\alpha\beta}^{rans} = \overline{u'_\alpha u'_\beta}$, as follows,

$$\partial_t \bar{u}_\alpha + \partial_\beta \bar{u}_\alpha \bar{u}_\beta = \frac{-1}{\rho}\partial_\alpha \bar{P} + \nu \partial_\beta \partial_\beta \bar{u}_\alpha - \partial_\beta \overline{u'_\alpha u'_\beta}. \tag{2.21}$$

Since the Reynolds stress term does not have a closed-form equation which can be explicitly solved, it must be approximated to close the system of equations.

Linear Eddy Viscosity Models (LEVM)

The primary closure for the of the RANS equations is the Boussinesq hypothesis, which assumes a linear relationship between the mean strain the the Reynolds stress, and can be written as,

$$\tau_{\alpha\beta}^{rans} = \frac{2}{3}k\delta_{\alpha\beta} - 2\nu_t S_{\alpha\beta}, \tag{2.22}$$

where $S_{\alpha\beta}$ is the mean strain rate as per Equation (2.23),

$$S_{\alpha\beta} = \frac{1}{2}(\partial_\beta \bar{u}_\alpha + \partial_\alpha \bar{u}_\beta) \tag{2.23}$$

and, ν_t is the eddy viscosity which is defined by the particular RANS closure model. However, in this section, the overbar ($\bar{}$) represents the Reynolds-averaged operator.

The most commonly used Linear Eddy Viscosity Model (LEVM) closures are the $k-\varepsilon$ and $k-\omega$ models, however, they are not accurate for many flow configurations, namely, three-dimensional flows with separation, streamline curvature and large pressure and buoyancy gradients [9]. This is a result of the underlying assumptions in these models, such as the Boussinesq assumption, not being valid in all cases. Consequently, there can be a large discrepancy between the true Reynolds stress and the modelled stress which results in erroneous predictions of flow characteristics [76].

The assumptions made in LEVM closure models are generally composed of a functional relationship between the mean flow properties and the turbulent quantities of interest. These were primarily informed by physical observations of fluid flow and the theoretical constraints understood at the time. However, the formulation of more advanced Reynolds stress closures via these processes is ongoing, as observed in the works of Jakirlić and Maduta [23] and Edeling et al. [16].

Reynolds Stress Modelling (RSM)

As discussed in the previous section, the linear relationship assumed by the Boussinesq hypothesis is inadequate. Therefore, to overcome these issues, the Reynolds Stress Modelling (RSM) approach uses six equations to estimate the Reynolds stress. In this way, the Reynolds stress convection, diffusion, production and redistribution are all included in the model. The full implementation of this model is not required for this study, however, Mansour et al. [38] provides an in-depth description of the six primary equations and the additional equations they each depend on.

While RMS provides superior performance over LEVM models, it is limited in other regards. For instance, RSM is more expensive than LEVM methods due to the introduction of the additional terms and equations. This negates the performance advantage of using Reynolds-averaged approach [69]. Furthermore, unlike LEVM models, the source terms in the RSM tend to dominate, resulting in numerical stiffness [69] and the RSM equations are highly non-linear which introduces another set of modelling challenges [32].

Explicit Algebraic Stress Modelling (EASM)

The Explicit Algebraic Stress Model (EASM) provides a suitable compromise between the simplistic Linear Eddy Viscosity Model and complex Reynolds Stress Model. The EASM approach is advantageous as it combines the robustness of LEVM while still maintaining the anisotropic qualities of RSM [69]. In this method, the Reynolds stress anisotropy tensor, $a_{\alpha\beta}$, is assumed to have the form given by Equation (2.24),

$$a_{\alpha\beta} = \frac{\overline{u'_\alpha u'_\beta}}{2k} - \frac{1}{3}k\delta_{\alpha\beta}. \tag{2.24}$$

Here, $\delta_{\alpha\beta}$ represents the Kronecker delta term, which is given by,

$$\delta_{\alpha\beta} = \begin{cases} 1 & \text{if } \alpha = \beta, \\ 0 & \text{if } \alpha \neq \beta. \end{cases} \tag{2.25}$$

The EASM approach traditionally models Equation (2.24) by a combination of the ten independent basis tensors, each with their own unique coefficient [49]. A derivation of these tensors is provided in later in the methodology section. Nonetheless, the EASM is the RANS model of choice to base the data-driven turbulence model off due to its relative flexibility, accuracy and ease of implementation [34, 69, 72].

2.2 Machine Learning in Turbulence Modelling

Recently, Durbin [14] has highlighted that there has been an increase in interest in utilising large data sets to find more reliable Reynolds stress closures. High-fidelity datasets are generated from DNS and LES and have recently become available due to increases in computing power. The more reliable Reynolds stress closures were developed using various statistical and computing tools which are commonly referred to as machine learning.

There have been a number of studies conducted in this field which can be broadly classified into two groups: uncertainty quantification and model development. Initially, machine learning was used for uncertainty development to gain insight into why certain turbulence models failed and aimed to correct these flaws. However, as computing power was increased, machine learning was used to develop new turbulence models and closures using existing high-fidelity data alone. This existing data is known as the training data and the goal is to minimise some cost function by modifying or adding terms within the turbulence closure.

2.2.1 Uncertainty Quantification

Uncertainty in CFD simulations arises from various sources. These include, but are not limited to, unknown or inaccurate parameters, initial conditions, boundary conditions, discretisation errors and approximations in turbulence models [15]. Therefore, data-driven methods have been used to not only measure the inadequacies of eddy-viscosity closures but also to suggest improvements to existing Reynolds stress closures.

For instance, the notion of using existing DNS data to assess traditional RANS closures and inform the creation of new ones was proposed by Parneix et al. [44]. They demonstrated that instead of using an equation for the turbulent kinetic energy dissipation rate, ε, in the $k-\varepsilon$ model, this can be instead obtained from DNS data. A similar method was used by Raiesi et al. [50] using the turbulent kinetic energy, k, data obtained from highly resolved LES in order to solve for ω in the $k-\omega$ model. While these methods improved the accuracy of flow predictions,

they each relied on existing high-fidelity data under the same flow conditions which may not be available.

2.2.2 Model Development

The process of altering existing or creating new RANS turbulence closures, using existing numerical data, is known as model development. High-fidelity data, from DNS or LES sources, can be fed into machine learning algorithms whereby a functional form of the new turbulence closure is modelled. This is an alternative approach to using experimental data and aims to use computational and statistical methods to overcome the shortcomings of traditional RANS models outlined in Section 2.1.3.

Altering Coefficients

The earliest applications of machine learning for model development involved deriving corrective coefficients for traditional Reynolds stress closures. The deficiencies of certain turbulence closures are well known and can be captured using existing high-fidelity data sources. Using these data sources, the least-squares algorithm can be used to optimise the coefficient of the eddy viscosity term in the $k-\omega$ SST turbulence model [58]. This method was found to better control the response of the eddy viscosity and vortex formation, in reference to turbo-machinery flow.

The successes of this method, when applied to the $k-\omega$ SST turbulence model for turbo-machinery flows, have also been replicated in Pichler et al. [47] and Weatheritt et al. [70]. Pichler et al. [47] used data generated from DNS to find two locally optimised coefficients: one to improve the calculation of the Reynolds stress term and the other for shear stress prediction. This study found that optimising these terms using corrective coefficients improved their understanding of the conditions under which the accuracy of RANS models deteriorate. This was confirmed by Weatheritt et al. [70] who used an evolutionary algorithm to modify the anisotropic Reynolds stress tensor using highly resolved LES data. As such, the Boussinesq assumption could be validated, with particular reference to the wake behind a turbine blade. Therefore, altering the coefficients of LEVM turbulence models, such as the $k-\omega$ SST, can be used to understand the relationship between flow parameters and to inform corrective studies.

However, the simplistic nature of the least-squares optimisation approach is problematic, particularly for complex turbo-machinery flows. Using a single data source, whether it be LES or DNS, essentially solves the flow in a frozen state [47, 70]. Therefore, the coefficients of the turbulence model in question are only optimised flow the flow conditions under which data is available.

Furthermore, only changing a small number of coefficients in the turbulence model can lead

to problems with the optimisation process. For instance, Pichler et al. [47] found that their methodology resulted in areas where the corrective coefficients produced inaccurate results as these terms were over-optimised to other areas of the flow. This suggests that care needs to be taken when developing new model terms as to ensure that the discovery of only local minima does not occur [24].

Finally, the use of only least-squares in the optimisation process is questionable. The disadvantages of the least-squares method are well known and documented [10, 30, 42], including its sensitivity to outliers and tendency to overfit. Therefore, a more robust optimisation algorithm may have been better suited to this non-linear problem.

Rather than calibrating model coefficients from existing data using an optimisation process, an inverse modelling approach can be utilised. This method is not as susceptible to the issues associated with optimisation and has shown to produce more accurate results [43]. For example, Parish and Duraisamy [43] applied a Gaussian Process (GP) machine learning approach where several corrective forms of the production term in the turbulent kinetic energy equation (using the $k-\varepsilon$ model) were generated for open channel flow. The functional correction term was found to more accurately predict flow quantities over previous studies, while also providing a measure of uncertainty.

A similar approach was utilised by Zhang and Duraisamy [80] who used neural networks to predict a similar correction factor, using high-fidelity channel flow data. This resulted in improved predictive capability, largely due to the benefits associated with neural networks which will be discussed further in Section 2.2.2. However, both these studies highlighted the drawbacks of applying machine learning models to affect only one corrective term in a turbulence model. Namely, Wu et al. [76] found that the extrapolation capabilities of this method were limited, such that the corrective factor could only be inserted into flows of similar geometries and flow conditions. This is because the corrective term only had no influence on the anisotropy of the predicted Reynolds stress tensor but could only affect the magnitude. Hence, an analysis of developing new turbulence model terms is required to gain a complete understanding of the flow using machine learning methods.

Developing New Turbulence Model Terms

More complex regression methods can be used to derive new model terms which can be inserted into existing turbulence models. These methods overcome many of the shortcomings of the approaches which only alter one coefficient of an existing model, as outlined in the previous section, and provide more accurate flow predictions. Nonetheless, there is no clear consensus as

to the optimal implementation of machine learning strategies to determine new turbulent closures. For instance, the choice of algorithm is dependent on the expected accuracy of prediction, what variables are to be included and how to enforce Galilean invariance to ensure accurate predictions.

One of the most critical decisions that needs to be made when modelling physical systems using data-driven methods is to determine what variables are to be passed onto the algorithms. Michalski [40] and Piatetsky-Shapiro [46] argue that very little structured data needs to be input into machine learning algorithms, such that if enough data is available, the algorithm should be able to work out the physical underpinnings of the data itself. This is an approach used by Ling et al. [33], Ling et al. [34] and Weatheritt and Sandberg [72] who each used only Pope's decomposition of the ten isotropic basis tensors, and five tensor invariants for an incompressible flow, to predict the Reynolds stress anisotropy tensor, $a_{\alpha\beta}$ using the Explicit Algebraic Stress Model [49]. However, rather than using theoretical approaches to determine the coefficients in the EASM, machine learning algorithms were used to determine the model terms by using various combinations of the basis tensors and invariants. Even using this small subset of possible variables, the studies conducted by these authors were shown to display significant improvements over the methods which altered existing coefficients.

On the other hand, Xiao and colleagues [65, 77] put forward the idea that data-driven turbulence models should take advantage of as much data on the underlying flow quantities as available. In a way, their models complement existing turbulence models and build in as much physical information about the flow which is available. For example, using this approach, Wang et al. [65] used 57 input variables (in comparison to the 15 of Ling et al. [34]), which included the complete set of basis tensors and invariants for the compressible case and included quantities such as the pressure gradient, gradient of turbulent kinetic energy, wall-based Reynolds number (Re_{wall}). However, the added complexity of additional variables does not always lead to improved model performance, with Weinmann and Sandberg [74] demonstrating that using a similar level of predictive accuracy could be achieved using only 5 input variables. Nonetheless, rather than modelling $a_{\alpha\beta}$ directly, Wang et al. [65] modelled the Reynolds stress anisotropy tensor discrepancy term, $\Delta\tau'$ which is given by $\Delta\tau' = \tau_{\alpha\beta}^{true} - \tau_{\alpha\beta}^{modelled}$, achieved using the methodology outlined in Figure 2.1.

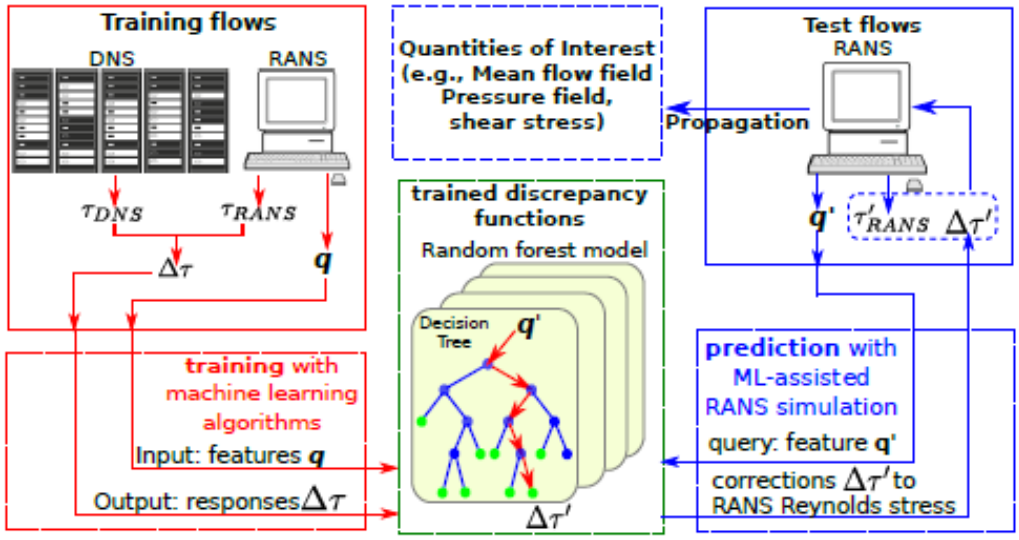


Fig. 2.1 Wang et al. [65] Reynolds stress anisotropy tensor discrepancy method

However, even methodologies which include information from as many sources as possible must ensure that a complete set of basis tensors are input into the machine learning model. Otherwise, there will be missing information, often resulting in poor model performance when attempting to predict the characteristics of the turbulent flow. The basis tensors and invariants, to be later introduced in the methodology of this study, are used to impose Galilean invariance. Galilean invariance means that Newton's laws of motion do not change in different frames of reference which is important for flow variables, such as velocity magnitude and pressure [33]. While it is not essential to maintain Galilean invariance, it is favourable as most traditional closures enforce it explicitly [33]. Therefore, the ease of implementation of invariance properties in various machine learning algorithms will often dictate the relative predictive capability of the data-developed turbulent closure. The details of several machine learning algorithms, along with an assessment of their appropriateness to meet the objectives of this study, are detailed below.

Random Forests

Random forests are a supervised machine learning technique which requires labelled 'truth' data, known as training data. This is the data in which the form of the Reynolds stress anisotropy tensor, $a_{\alpha\beta}$ can be estimated from. Random forests are collections of decision trees, where each tree uses if-then logic to categorise data points to certain branches based on their values. An example decision tree is shown in Figure 2.2.

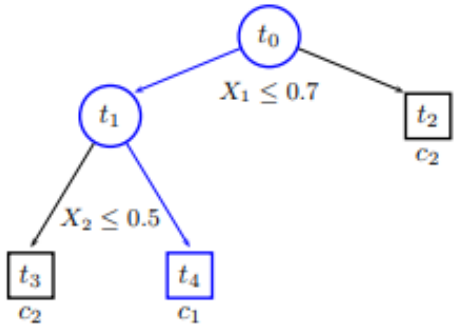


Fig. 2.2 Example decision tree. Adapted from [37]

To train the random forest, each decision tree is trained on a subset of the training data which is a process known as bagging [37]. In general, random forests are robust to over-fitting and are high-performing [3].

In terms of developing new turbulent model terms, Ling and Templeton [36] have applied random forest algorithms when modelling the anisotropic Reynolds stress to validate the isotropy and non-negativity of eddy viscosity and the linearity of the Boussinesq hypothesis. They found that the random forest algorithm: displayed good performance validating these objectives, could be generalised to other flow conditions and had a relatively straightforward implementation. This is supported by another study on predicting the Reynolds stress anisotropy tensor by Ling et al. [35] using random forests where this algorithm again displayed the ability to generalise across various flow conditions (e.g. different Reynolds numbers) and generated improved predictions when compared to LEVM closures.

However, random forests are considered 'black boxes' with complex networks of decision trees making them difficult to interpret [36]. As such, there is not a simple mathematical form which can be readily understood. Furthermore, random forests require large amounts of data for training if they are to be made generalisable to many types of flows [61]. Even with increased amounts of data, random forests do not allow Galilean invariance to be enforced [35, 61]. Finally, there is limited research to understanding whether turbulence models, using random forests, will converge to a final solution and whether this type of algorithm can be used to predict other flow quantities of interest such as heat fluxes and wall stresses [35].

Evolutionary Algorithms

Evolutionary algorithms aim to produce a simple, yet accurate, mathematical equation to represent high-fidelity turbulence data [28]. Unlike random forests, no functional form of the data is initially assumed. Therefore, the entire domain of possible solutions is searched by the algorithm.

The formulation of evolutionary algorithms is complex, however, the methodology undertaken by Weatheritt and Sandberg [71] is shown in Figure 2.3.

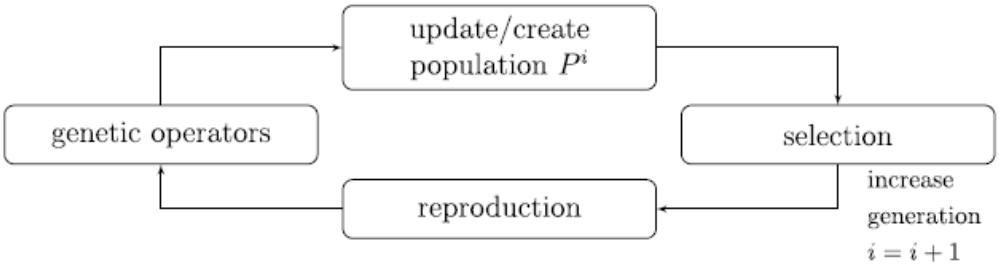


Fig. 2.3 Flow chart of the Weatheritt and Sandberg [71] evolutionary algorithm

Here, a population P^i of possible solutions is randomly generated using a collection of variables and mathematical symbols. By applying selection criteria, only the 'fittest' solutions are reproduced and slightly modified by additional genetic operators, whereby the process starts again. This process repeats until a satisfactory representation of the data is obtained.

The evolutionary approach has shown to have numerous advantages. Firstly, its construction means that a mathematical representation of the turbulence closure is obtained, allowing the researcher to understand the underlying relationships [71, 72]. Furthermore, it is straightforward to embed Galilean invariance into the algorithm by using the isotropic basis tensors and five invariants which were introduced in a previous section [69]. Finally, the methodology employed by a evolutionary algorithm is flexible enough to be applied to varying geometries [72], for hybrid RANS/LES methodologies [69, 73] and can be readily implemented into existing CFD codes [72].

Yet, the relatively simplistic nature of evolutionary algorithms means that the complexities of turbulent flow cannot be completely captured using an interpretable mathematical form [72]. Methodologies employed by Ling et al. [34] and Wu et al. [76] using neural networks generally yield more accurate results. Moreover, the construction of evolutionary algorithms means that they are susceptible to optimising to only a local minima [51]. This was an issue reported by Weatheritt and Sandberg [71] and therefore, the modeller is required to be aware of these deficiencies.

Neural Networks

Another machine learning algorithm which has gained popularity for developing new RANS turbulence closures are neural networks, which are sometimes referred to as deep learning algorithms. Neural networks transform input variables (input layer) through several layers of

non-linear interactions (hidden layers) to produce an output (output layer) [31]. This algorithm is widely used in applications such as voice recognition [21], image classification [29] and finance [62] because of its ability to represent highly complex, non-linear relationships within data. This feature allowed Ling et al. [33] to embed invariance properties into their model, improving the performance over random forest implementations. Ling et al. [34] again showed that embedding Galilean invariance using a neural network is possible, as per Figure 2.4.

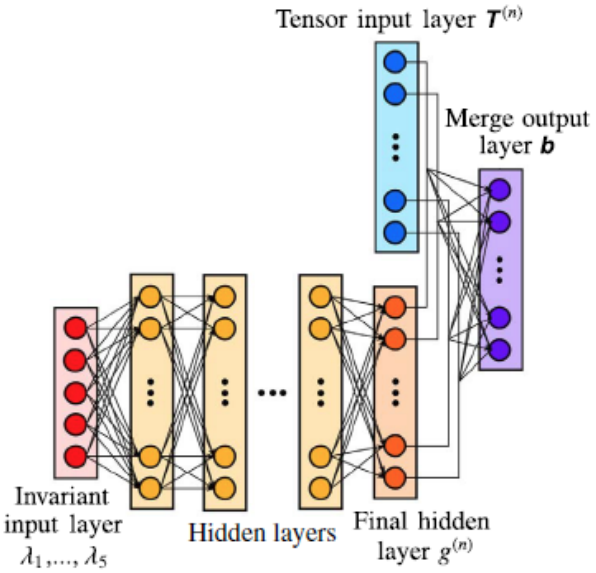


Fig. 2.4 Schematic of Ling et al. [34] neural network with Galilean invariance enforced

Like random forests, neural network are also are black boxes algorithms. However, this issue can be overcome by using a technique known as Partial Dependence Plotting (PDP), used to determine the marginal effect of an input variable on the output, which is employed in this work. Nonetheless, since the neural network algorithm has been demonstrated to yield high performance and also provides a flexible implementation structure, this is the algorithm of choice for the following analysis. The details of the implementation of the neural network, including a description of how the algorithm works, is provided in the methodology section.

2.3 Summary of Literature Review

Upon reviewing the literature, it is clear that there is a need to develop more accurate RANS models to better understand stratified open channels. Traditional RANS methods, including Linear Eddy Viscosity Models, are heavily dependent on assumptions, and therefore are unsuitable for use in this investigation. While high-fidelity methods, such as Direct Numerical Simulation, could instead be implemented, its computational requirement is far too costly. Thus, demonstrating the need for new, data-driven RANS models.

An exploration into the earliest implementations of data-driven methods was conducted. Firstly, uncertainty in existing CFD simulations can be quantified by analysing existing databases of fluid data. The data can be used to inform the alteration single terms within a given RANS model, to improve the accuracy of future simulations. This proposed method is an early form of turbulence model development. While it has been shown that the altering coefficients improves predictive performance of RANS models, this method is somewhat limited as it: requires large sources of existing data, has an upper threshold of accurate as only a small number of model terms are changed, and it is likely to over-optimise and therefore can only be used on a select number of flow cases.

For these reasons, the final branch of data-driven turbulence modelling, where new turbulence model terms are found directly, will instead be used in this work. The term of focus will be the (anisotropic) Reynolds stress to close the RANS equations. Based on the work of others [34, 66, 72], the new turbulence model developed will follow the Explicit Algebraic Stress Model. As discussed, the input variables to the machine learning algorithm will include the ten isotropic basis tensors and the five invariants, and if they are properly implemented, will ensure Galilean invariance is enforced. Based on a range of machine learning algorithms, a neural network will be adopted in this work. Neural networks have superior predictive performance and are flexible enough to correctly implement an EASM turbulence model. While this algorithm is a black-box, techniques exist to allow the identification of the most important input variables.

Chapter 3

Methodology

This section details the framework used to create the machine learning model for the anisotropic Reynolds stress, including the data and pre-processing required to generate the input variables, the machine learning algorithms used to construct the predictive model and the criteria used to evaluate the success of the final model.

3.1 Problem Definition

The experimental setups proposed by Ling et al. [34] and Weatheritt and Sandberg [72], outlined in the literature review, were adapted for this thesis. Specifically, the anisotropic Reynolds stress tensor in Equation (3.1),

$$a_{\alpha\beta} = \frac{\overline{u'_\alpha u'_\beta}}{2k} - \frac{1}{3}k\delta_{\alpha\beta}, \tag{3.1}$$

is the target (dependent) variable of the investigation. Theoretically, this representation of the Reynolds stress tensor could be inserted into the RANS equations and solved using the CFD package of choice.

To produce a machine learning model which best represents the anisotropic Reynolds stress tensor, the Explicit Algebraic Stress Model (EASM) was used as a starting point. As described in Section 2.1.3, the anisotropic Reynolds stress tensor can be written as a function of the normalised strain and rotation tensors of the form,

$$a_{\alpha\beta} = g^{(n)}(S_{\alpha\beta}, R_{\alpha\beta}), \tag{3.2}$$

where $S_{\alpha\beta}$ is given by,

$$S_{\alpha\beta} = \frac{k}{\varepsilon} \times \frac{1}{2}(\partial_\beta \bar{u}_\alpha + \partial_\alpha \bar{u}_\beta), \tag{3.3}$$

and $R_{\alpha\beta}$ is given by,

$$R_{\alpha\beta} = \frac{k}{\varepsilon} \times \frac{1}{2}(\partial_\beta \bar{u}_\alpha - \partial_\alpha \bar{u}_\beta). \tag{3.4}$$

This study focuses on the flow of water through an open channel. Since water is an incompressible fluid, Pope [49] showed that the representation of Equation (3.2) can be decomposed into a linear combination of 10 isotropic basis tensors under the Caley-Hamilton theory,

$$a_{\alpha\beta} = \sum_{n=1}^{10} g^{(n)}(I_1, \ldots, I_5) T^{(n)}_{\alpha\beta}, \tag{3.5}$$

where the 10 isotropic basis tensors ($T^{(n)}_{\alpha\beta}$) are given by,

$$T^{(1)}_{\alpha\beta} = S_{\alpha\beta} \tag{3.6a}$$
$$T^{(2)}_{\alpha\beta} = S_{\alpha\beta}R_{\alpha\beta} - R_{\alpha\beta}S_{\alpha\beta} \tag{3.6b}$$
$$T^{(3)}_{\alpha\beta} = S^2_{\alpha\beta} - \frac{1}{3}I \cdot Tr(S^2_{\alpha\beta}) \tag{3.6c}$$
$$T^{(4)}_{\alpha\beta} = R^2_{\alpha\beta} - \frac{1}{3}I \cdot Tr(R^2_{\alpha\beta}) \tag{3.6d}$$
$$T^{(5)}_{\alpha\beta} = R_{\alpha\beta}S^2_{\alpha\beta} - S^2_{\alpha\beta}R_{\alpha\beta} \tag{3.6e}$$
$$T^{(6)}_{\alpha\beta} = R^2_{\alpha\beta}S_{\alpha\beta} + S_{\alpha\beta}R^2_{\alpha\beta} - \frac{2}{3}I \cdot Tr(S_{\alpha\beta}R^2_{\alpha\beta}) \tag{3.6f}$$
$$T^{(7)}_{\alpha\beta} = R_{\alpha\beta}S_{\alpha\beta}R^2_{\alpha\beta} - R^2_{\alpha\beta}S_{\alpha\beta}R_{\alpha\beta} \tag{3.6g}$$
$$T^{(8)}_{\alpha\beta} = S_{\alpha\beta}R_{\alpha\beta}S^2_{\alpha\beta} - S^2_{\alpha\beta}R_{\alpha\beta}S_{\alpha\beta} \tag{3.6h}$$
$$T^{(9)}_{\alpha\beta} = R^2_{\alpha\beta}S^2_{\alpha\beta} + S^2_{\alpha\beta}R^2_{\alpha\beta} - \frac{2}{3}I \cdot Tr(S^2_{\alpha\beta}R^2_{\alpha\beta}) \tag{3.6i}$$
$$T^{(10)}_{\alpha\beta} = R_{\alpha\beta}S^2_{\alpha\beta}R^2_{\alpha\beta} - R^2_{\alpha\beta}S^2_{\alpha\beta}R_{\alpha\beta}. \tag{3.6j}$$

Furthermore, the 5 scalar invariants (I_n) of Equation (3.5) are defined as,

$$I_1 = Tr(S^2_{\alpha\beta}) \tag{3.7a}$$
$$I_2 = Tr(R^2_{\alpha\beta}) \tag{3.7b}$$
$$I_3 = Tr(S^3_{\alpha\beta}) \tag{3.7c}$$
$$I_4 = Tr(R^2_{\alpha\beta}S_{\alpha\beta}) \tag{3.7d}$$
$$I_5 = Tr(R^2_{\alpha\beta}S^2_{\alpha\beta}). \tag{3.7e}$$

Hence, the goal of the machine learning model is to use the 5 scalar invariants $(I_1, \ldots, I_5)$ as an input, to find the 10 scalar coefficients $(g^{(1)}, \ldots, g^{(10)})$. In addition, the research conducted by Wang et al. [67, 65] showed that Pope's definition in Equation (3.5) can be extended to include further variables to improve the accuracy of the predictive model while still retaining Galilean invariance. Nonetheless, the scalar coefficients, which are a function of the invariants and any other relevant inputs, are then linearly combined with the 10 isotropic basis tensors $(T^{(1)}_{\alpha\beta}, \ldots, T^{(10)}_{\alpha\beta})$ to find the anisotropic Reynolds stress tensor as per Equation (3.5).

Finally, a key goal of any data-driven turbulence modelling approach is satisfying Galilean invariance. As described by Pope [49] any Reynolds stress anisotropy tensor which satisfies Equation (3.5) will also satisfy the condition of Galilean invariance. Therefore, this definition of the problem will meet a key objective of this work.

3.2 Data Generation

3.2.1 CFD Setup

Predictive models developed using machine learning methods require large amounts of high-quality data. Therefore, several PUFFIN [25] large eddy simulations (LES) were run to generate high-fidelity flow data through an open-channel under various conditions. Namely, two broad classes of simulations were considered: isothermal open channel flow and stratified open channel flow. The following section outlines the numerical set-up of the Computational Fluid Dynamics simulation for these two cases.

Domain

The domain in question is an open channel flow. The schematic of the domain is shown in Figure 3.1.

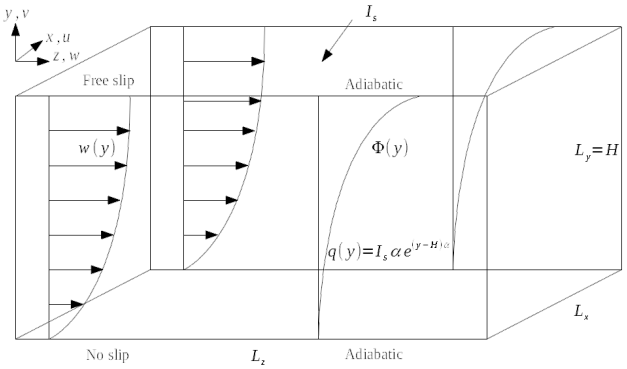


Fig. 3.1 Schematic of flow. Adapted from Williamson et al. [75].

The domain is identical for the isothermal and stratified open channel simulations conducted in this study. The geometry was bounded by:

$$\begin{aligned} \frac{-\pi}{2} \leq L_x \leq \frac{-\pi}{2} \text{ (cross-stream)} \\ 0 \leq L_y \leq 1 \text{ (vertical)} \\ -\pi \leq L_z \leq \pi \text{ (stream-wise).} \end{aligned}$$

Continuous Form Equations

Both the isothermal and stratified flow simulations in this thesis use the PUFFIN code developed by Kirkpatrick [25]. The governing equations for the Large Eddy Simulation (LES) are written in filtered Boussinesq form as:

$$\partial_\alpha \bar{u}_\alpha = 0, \tag{3.8}$$

$$\partial_t \bar{u}_\alpha + \partial_\beta \bar{u}_\alpha \bar{u}_\beta = -\frac{1}{\rho_0}\partial_\alpha \bar{P} + \delta_{\alpha 2} g \frac{\phi}{\phi_0} + \partial_\beta \left(2\nu \bar{S}_{\alpha\beta}\right) - \partial_\beta \tau_{\alpha\beta}, \tag{3.9}$$

$$\partial_t \phi + \partial_\beta \bar{u}_\beta \phi = \partial_\beta \left(\frac{\nu}{Pr}\partial_\beta \phi\right) - \partial_\beta \gamma_\phi. \tag{3.10}$$

In these equations, $\bar{u}_\alpha$ are the filtered velocity components of the fluid, ϕ is the temperature perturbation, ϕ_0 is the reference temperature of the fluid and Pr is the Prandtl number which is given in the general form,

$$Pr = \frac{c_p \mu}{k}, \tag{3.11}$$

where c_p is the specific heat, μ is the dynamic viscosity and k is the thermal conductivity of the fluid.

Nonetheless, Equations (3.8)-(3.10) are spatially filtered, as denoted by the bar operator, removing high wave-number modes that the numerical model cannot resolve. As a result, the filter yields the Sub-Filter Scale (SFS) turbulent stresses and fluxes as defined in Equations (3.12) and (3.13).

$$\tau_{\alpha\beta} = (\overline{u_\alpha u_\beta} - \bar{u}_\alpha \bar{u}_\beta) \tag{3.12}$$

$$\gamma_\phi = (\overline{\phi u_\beta} - \bar{\phi}\bar{u}_\beta) \tag{3.13}$$

Here, the SFS turbulent stresses and fluxes are the unresolved scales of motion and are parameterised using a dynamic mix model. For instance, the model for the SFS stress is written as,

$$\tau_{\alpha\beta} = (\overline{\bar{u}_\alpha \bar{u}_\beta} - \overline{\bar{u}_\alpha}\overline{\bar{u}_\beta}) - 2C\Delta^2|\bar{S}|\bar{S}_{\alpha\beta}C_B. \tag{3.14}$$

In this model, $|\bar{S}| = \sqrt{2\bar{S}_{\alpha\beta}\bar{S}_{\alpha\beta}}$, C is a dimensionless coefficient, Δ is the filter width and C_B is a stability function to incorporate statically stable and unstable stratification effects.

Moreover, in this simulation, the radiative heating effects, for the stratified flow, are represented using the Beer-Lambert Law. This is equated as,

$$q(y) = I_s \alpha e^{(y-H)\alpha}. \tag{3.15}$$

Here, H is the height of the open channel in the vertical direction. The isothermal flow case had no radiative heat effects applied to the top of the open channel.

Boundary Conditions

The open channel has a no-slip, adiabatic wall at the lower boundary, and a free-slip, adiabatic boundary at the top surface. Furthermore, there are also periodic boundaries in the stream-wise and also in the cross-flow directions.

The boundary conditions at the bottom of the open channel ($y = 0$) and at the top of the channel ($y = 1$) are as per Equations (3.16) and (3.17).

$$y = 0 : \bar{u} = \bar{v} = \bar{w}; \frac{\partial \bar{\phi}}{\partial y} = 0 \tag{3.16}$$

$$y = 1 : \frac{\partial \bar{w}}{\partial y} = \frac{\partial \bar{u}}{\partial y} = 0; \bar{v} = 0; \frac{\partial \bar{\phi}}{\partial y} = 0 \tag{3.17}$$

Furthermore, the Reynolds number was set to $Re_t = 225$ for both the isothermal and stratified flow cases.

Grid and Time Step Size

The mesh was constructed on a Cartesian grid which was non-uniform and staggered. This was as per Table 3.1.

Table 3.1 Grid Used

	cells	**min**	**max**
Δx	64	$-\pi/2$	$\pi/2$
Δy	82	0	1
Δz	120	$-\pi$	π

Since the vertical (y-)direction is the most important, several fine regions of cells were added. This was done in order to accurately model the turbulent structures, in particular, the viscous sub-layer, close to the boundaries of the open channel. This is shown in Table 3.2.

Table 3.2 Fine Regions in the Grid

	cells	**y-min**	**y-max**
Fine region 1	1	0.000	0.003
Fine region 2	20	0.400	0.800
Fine region 3	1	0.992	1.000

Time Steps

The simulations, for both the isothermal and stratified flow cases, were run for a total of 50-time units (where a time-unit is a non-dimensional time measurement). However, the results were only recorded as a single snapshot every 1-time unit, between (and inclusive of) time units 30 and 50. This resulted in 21 different data files between this time period.

Discretisation

The Large Eddy Simulation equations were discretised using a finite volume formulation. The advection scheme for the momentum and scalar equations are 4^{th} order accurate in space. The time discretisation is 2^{nd} order accurate.

3.2.2 Modelling Variables

The processing of learning the anisotropic Reynolds stress based on a set of input variables is known as a supervised learning task. A supervised learning task maps a set of input variables to an output variable in a functional form. However, supervised learning problems require ‘labelled’ training data which essentially means that the target (dependent) variable is required to be known before the machine learning process can take place.

In this study, the three-dimensional flow through the open channel can be simplified to only the y-z plane, where the y-coordinate is the vertical direction and the z-coordinate is the direction of the flow. As a result, Equation (3.1), which defines the anisotropic Reynolds stress tensor, now becomes a_{yz}. This target variable is no longer a tensor, but instead is a scalar value which represents the anisotropic Reynolds stress on the y-z plane only. Prior to training the machine learning model, a_{yz} was calculated using the velocity field data from the open channel simulations using a Python script. Here, the velocity components were averaged across 10 non-dimensional time-units on the y-z planes.

Moreover, the input variables of this supervised learning task would be the 10 isotropic basis tensors (see Equations (3.6a)-(3.6j)) and the 5 invariants (see Equations (3.7a)-(3.7e)). These were again calculated using a Python script after first calculating the normalised strain and rotation tensors. It should be noted, that the calculation of the 10 isotropic basis tensors produced a (3×3) tensor, with each element representing the respective principal and plane directions in three-dimensional space. However, since the flow through the open channel can be simplified to the y-z plane, the other directions were excluded from the dataset.

Furthermore, the results obtained by Wang et al. [67] demonstrate it is possible to extend the Explicit Algebraic Stress Model (EASM) to include additional variables to improve the accuracy of predicting the anisotropic Reynolds stress. It is theorised that a wall-based Reynolds number would function as a good predictor for a_{yz} as this variable would provide a non-dimensional method of quantifying the distance from the lower channel wall. Since the variable is non-dimensional, it is applicable to open channel flows of varying geometries. Modifying the definition as found in Wang et al. [67], this study defines the wall-based Reynolds number as,

$$Re_{wall} = \frac{ky}{2\nu}, \tag{3.18}$$

where y is the distance from the wall of the open-channel, k is the turbulent kinetic energy and ν is the kinematic viscosity.

Finally, the last input variable used in this investigation would aim to allow the machine learning algorithm to distinguish between isothermal and stratified flow. This was decided to be the gradient of the non-dimensional temperature fluctuation field, ϕ. Thus, the variable had the form,

$$\text{phi\_grad} = \frac{\phi_{y+1} - \phi_{y-1}}{2\Delta y}, \tag{3.19}$$

where Δy is the grid size.

In summary, the output and input variables are listed in Table 3.3.

Table 3.3 Summary of Modelling Variables

	Variable	**Description**
Output/Dependent	a_{yz}	Anisotropic Reynolds stress on the y-z plane
Input/Independent	$T_{yz}^{(n)}$ for n = 1-10	10 isotropic basis tensors on the y-z plane
	I_n for n = 1-5	5 invariants
	Re_{wall}	Wall-based Reynolds number
	phi_grad	Gradient of non-dimensional temperature fluctuation

3.2.3 Training, Validation and Test Sets

Throughout the following sections, reference will be made to the training, validation and test sets of data. The training set is a subset of the overall dataset on which the machine learning model is trained on (i.e. to learn the function mapping the input variables to the output variable). The validation set is used for model selection, which is the process of selecting the 'best' model from a range of potential models. Finally, the test set is used to provide an unbiased final evaluation of the predictive model. The test dataset is only used at the very end, after the machine learning process is complete, and is used to provide an insight as to how the predictive model would perform on unseen data (e.g. under different flow conditions).

Due to computational limitations, the size of the complete dataset, including the train, validation and test sets, was reduced to 200,000 points via a random sampling process implemented in Python. Of these 200,000 data points, 100,000 were taken from the isothermal open channel LES and the other 100,000 from the stratified LES.

Moreover, the 200,000 data points were split according to a 56/24/20 partition. This means that 56% of the total dataset was allocated to the training set, 24% for the validation set and 20% for the test set. Again, the data was randomly split as to not induce any sampling bias.

The relationship between the three datasets; training, validation and test, are outlined in the following steps:

1. The complete dataset (i.e. combined isothermal and stratified LES data) is randomly sampled to create a 200,000 subset. 100,000 points are from the isothermal LES and 100,000 from the stratified LES.

2. The randomly sampled subset is split into 3 parts as per Table 3.4.

Table 3.4 Summary of Datasets

Sample	Sample Size	Purpose
Train	112,000	Create several candidate models with different structures
Validate	48,000	Select the 'best' of the candidate models
Test	40,000	Final evaluation of the model to see how well it will perform on unseen data

3. Several machine learning models are built on the *training* data. The model building process is outlined in Section 3.3.

4. Once each predictive model has been built on the training data, its performance is checked on the *validation* data. The model which performs the 'best' on the validation data is selected, as it is assumed that this model would perform the best on unseen data. The criteria by which the 'best' model is determined are outlined in Section 3.4.

5. The final model is then checked on the *test* data, which is a final sanity check of the model. The result on the test data should not differ too much from the validation result. If the results are vastly different, the process starts at Step 3 again.

3.3 Modelling

There are a number of machine learning algorithms which can be used to create a functional mapping between the input variables and output variable in a supervised regression problem. Several of these algorithms were introduced in the literature review with their relative advantages and disadvantages highlighted in context. With these in mind, this thesis focuses on a branch of machine learning algorithms known as neural networks (commonly referred to as deep learning). The reasons why this algorithm was selected, in favour of the alternatives, are as follows:

1. **Flexibility of architecture**: a goal of this investigation is to develop a methodology to predict the anisotropic Reynolds stress with embedded Galilean invariance. This requires the implementation of the EASM which can be readily achieved using open-source neural network software packages.

2. **Accuracy**: as discussed in the literature review, neural networks often outperform other types of algorithms when large amounts of data are available. As accuracy is a key consideration of the final predictive model, this algorithm type was selected.

3. **Open-source support**: within the Python community, one of the most popular open-source implementations of neural networks is the software package Keras [6]. Keras is a high-level package which allows developers to quickly prototype their ideas.

Therefore, this section provides a more thorough introduction to neural networks, the exact structure of the network used in this investigation, and the steps taken to optimise the accuracy of the final predictive model.

3.3.1 Neural Networks

A neural network is a type of machine learning algorithm which was inspired by the mechanisms of the human brain. A neural network is composed of interconnected layers of neurons, which is the basic unit of the algorithm. These neurons receive an input and compute an output based on a series of weights and an activation function. The basic structure of a neural network is shown in Figure 3.2.

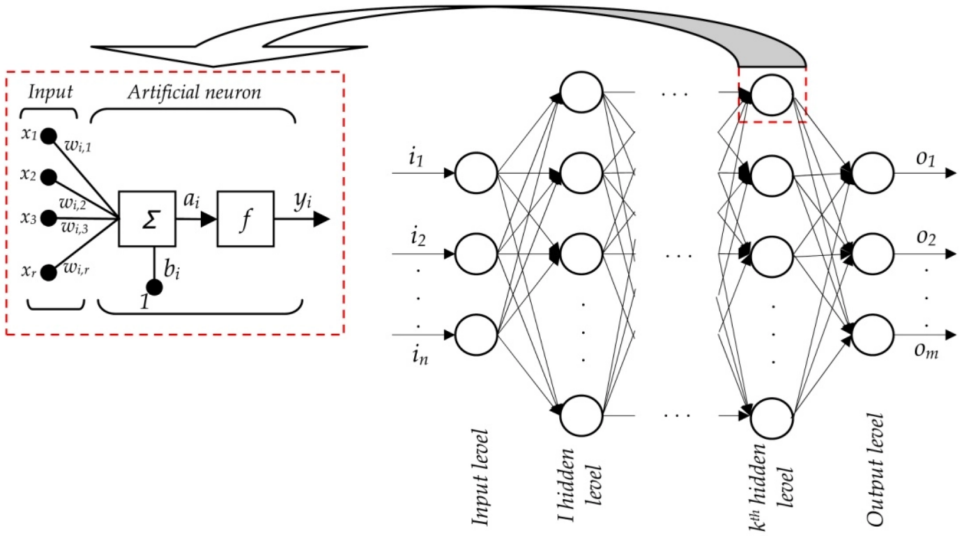


Fig. 3.2 Neural Network Structure. Adapted from Despotovic [11].

As shown in Figure 3.2, a neural network can be broken up into three layers:

1. **Input layer**: where the input variables, including the isotropic basis tensors, invariants, wall-based Reynolds number and temperature variable, is passed into the network.

2. **Hidden layers**: where the intermediate processing takes place between the input and output layers, allowing complex non-linear relationships in the underlying data to be found.

3. **Output layer**: produces the output in the desired format, which in this study, is a continuous number representing a_{yz}.

Furthermore, each neuron has several components which are used to mathematically represent the relationships in the data as shown in Figure 3.3.

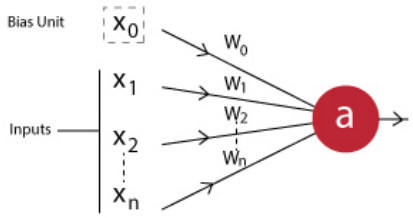


Fig. 3.3 Individual Neuron Structure. Adapted from Jain [22].

As shown in Figure 3.3 each individual neuron has the following components:

1. **Inputs** ($x_1, x_2, \ldots, x_n$): these are either directly from the input layer or the output from a previous hidden layer.
2. **Bias** (x_0): a constant value which is added to the activation function. This is similar to adding an intercept on a linear line of best fit.
3. **Weights** ($w_0, w_1, w_2, \ldots, w_n$): coefficients by which each input is multiplied by .
4. **Output** (a): the output of the neuron which is either the input to the next layer or the final output of the model.

These four components are combined as per the relationship defined in Equation (3.20),

$$a = f(\sum_{i=0}^{N} w_i x_i), \tag{3.20}$$

where f is defined as the activation function. This is a function by which the sum of the weights and inputs (including the bias) forms the input, thus producing the output a. The activation function is what allows complex non-linear relationships to be represented by the neural network model. In this investigation, the ReLU (Rectified Linear Unit) activation function is used, which is often the default activation function for neural networks for regression purposes [78], and is defined as,

$$f(\sum_{i=0}^{N} w_i x_i) = max(0, f(\sum_{i=0}^{N} w_i x_i)). \tag{3.21}$$

3.3.2 Tensor Basis Neural Network

Based on the definition of the problem outlined in Section 3.1, the neural network used in this thesis was an adaptation of the Tensor Basis Neural Network (TBNN) used by Ling et al. [34]. A generalised structure of the TBNN used in this investigation is shown in Figure 3.4. The

TBNN structure is advantageous as it installs Galilean invariance directly into the output of the predictive model because it mimics the Explicit Algebraic Stress Model formulation.

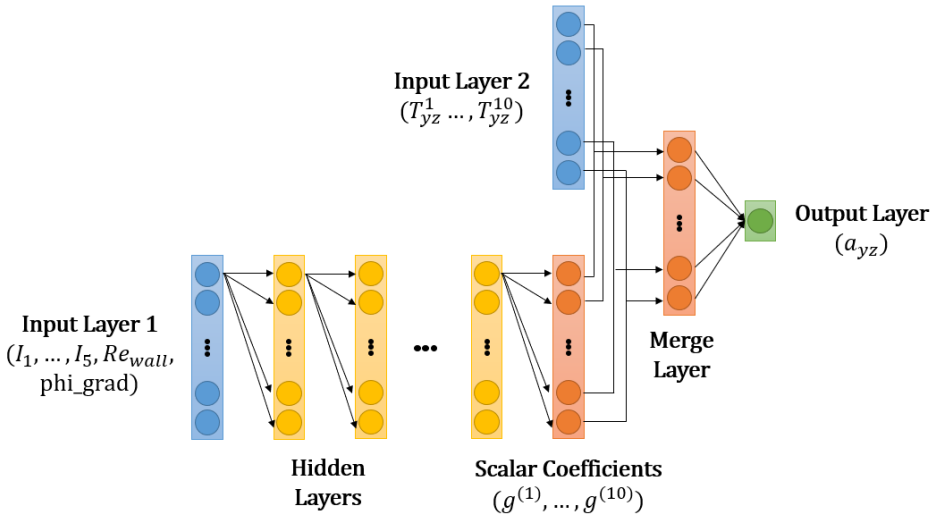


Fig. 3.4 TBNN Structure. Adapted from Ling et al. [34] and updated for this study.

The goal of the TBNN structure is to take the variables from input layer 1, passing them through the hidden layers where the complex, non-linear relationships are found in the form of the ten scalar coefficients. These are then linearly combined with the isotropic basis tensors, in input layer 2, to find the prediction for a_{yz}.

Now, since the open channel flow problem in this study only has only a single dimension on the y-z plane, the Explicit Algebraic Stress Model, with our two additional variables, is expanded as follows:

$$\begin{aligned} a_{yz} &= \sum_{n=1}^{10} g^{(n)}(I_1,\ldots,I_5,Re_{wall},phi\_grad)T_{yz}^{(n)} \\ &= g^{(1)}T_{yz}^{(1)} + g^{(2)}T_{yz}^{(2)} + \ldots + g^{(10)}T_{yz}^{(10)}. \end{aligned} \tag{3.22}$$

Therefore, it is possible to observe the similarities of Equation (3.22) to the general form of multiple linear regression (MLR) in the following equation:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_{p-1} x_{i,p-1} + \varepsilon_i, \tag{3.23}$$

where y_i is the dependent variable, $x_{i,p}$ are the input variables, and ε_i is the error term. Using this example, the scalar coefficients of the EASM ($g^{(n)}$) are equivalent to the MLR model coefficients (β_p) and the isotropic basis tensors ($T_{yz}^{(n)}$) are equivalent to the MLR dependent variables ($x_{i,p}$).

Moreover, since Figure 3.4 presents only a generalised structure of the TBNN, the exact structure needs to be found, such that the machine learning model is as accurate as possible. The process of determining the structure of a machine learning model is known as hyperparameter optimisation and the methodology used in this process is described in the following section.

3.3.3 Hyperparameters

In machine learning, a hyperparameter is a parameter of a model which needs to be selected prior to the training process. In the case of neural networks, some of the hyperparameters which must first be selected include those which determine the structure of the network (e.g. number of hidden layers, number of neurons in each layer) and those which determine how the network is trained (e.g. optimisation algorithm). For this study, the following hyperparameters were considered:

- **Layer structure**: the number of neurons in each layer and the number of layers.
- **Data transformation**: making the input data more normal (i.e. following a normal distribution) to improve training speed and predictive accuracy [56].
- **Batch size**: a subset of the entire training data which is fed through the network after which the parameter update process occurs.
- **Epochs**: the number of times the entire set of training data is passed through the network during the training process. Therefore, there are several batches for each epoch.
- **Optimisation algorithm**: determines what values for the parameters (those which change during the training process, unlike hyperparameters which do not) would make the machine learning model as accurate as possible.
- **Initial network weights**, $(w_0, w_1, \ldots, w_n)$: changing the initial values of the weights.
- **Dropout**: a regularisation technique to ensure the model is not overfit to the training data, thus increasing its generalisability.

Further discussion on these hyperparameters will be provided in context with the results of this investigation.

3.3.4 Hyperparameter Optimisation

The process of finding the optimal set of hyperparameters to minimise the error is known as hyperparameter optimisation [53]. In this study, the combination of hyperparameters that minimises the Root Mean Squared Error (RMSE) on the validation data, to be explained in

Section 3.4.1, is deemed to be the 'optimal' permutation of hyperparameters. To select the hyperparameters, a manual search process was conducted, which still remains the most common method of hyperparameter optimisation [7].

3.4 Model Evaluation

The success of the machine learning model to predict the anisotropic Reynolds stress on the y-z plane will be evaluated on the set of three criteria: accuracy, generalisability and interpretability. These are listed in order of decreasing importance and are individually discussed in the following subsections.

3.4.1 Accuracy

The primary objective of this study is to produce a predictive model which best estimates the true anisotropic Reynolds stress on the y-z plane, as provided by the LES of open channel flow. The model for a_{yz} could then theoretically be inserted into a RANS solver which would ideally produce results that are more accurate than traditional RANS modelling approaches. As accuracy is a primary concern for any Computational Fluid Dynamics project, it is adopted as the number one measure of success for the machine learning model in this work.

The predictions of the machine learning model, a_{yz}, come in two forms: instantaneous and mean/median. The instantaneous predictions for a_{yz} are taken for each individual point within the open channel at any given time unit. While instantaneous predictions are useful for the training and validation phases of the machine learning process, it is more common in CFD applications, particularly Reynolds *Averaged* Navier-Stokes turbulence models, to report summarised statistics. For each given y-coordinate and over all time units, the mean/median values of the instantaneous predictions a_{yz} of can be calculated.

Nonetheless, in statistical regression problems, a loss function is used to quantify the predictive error of the outputs of a machine learning model. While there are a number of loss functions which could be used, a commonly used loss function in regression is the Mean Squared Error (MSE). The MSE measures the average of the squared differences between the predicted values and the true values. It is defined as,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3.24}$$

where the terms $1/n\sum_{i=1}^{n}$ represent the *mean* and $(y_i - \hat{y}_i)^2$ represent the *squared errors*. Noting that y_i in Equation (3.24) represents the anisotropic Reynolds stress on the y-z plane obtained

from the LES data (i.e. $y_i = a_{yz}^{LES}$) and $\hat{y}_i$ is the anisotropic Reynolds stress on the y-z plane obtained from the predictive model (i.e. $\hat{y}_i = a_{yz}^{pred}$).

Furthermore, taking the square root of the MSE yields the Root Mean Squared Error,

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{3.25}$$

The use of the RMSE is advantageous as it is in the same units as a_{yz}, which is Ns^2/m^4.

The RMSE can be reported for both instantaneous and mean/median predictions for a_{yz}. In this study, the following metrics will be reported:

- **RMSE Training Combined**: this is calculated based on a combination of the isothermal and stratified training datasets. As previously described, the training data is used only in the first step to estimate the various parameters (weights, bias, etc.) of the neural network, prior to the validation data. The data this metric is calculated on is instantaneous.
- **RMSE Validation Combined**: this is also a combination of the isothermal and stratified datasets. However, this metric uses the validation data and is therefore the primary metric for selecting the final machine learning model. The data this metric is calculated on is instantaneous.
- **RMSE Validation Isothermal - Mean**: this metric is calculated by taking the mean value of the instantaneous predictions for a given y-coordinate on the validation data for the isothermal flow case. These predictions are similar to the averaged statistics as calculated by a RANS model which uses a similar averaging process.
- **RMSE Validation Isothermal - Median**: this metric is calculated by taking the median value of the instantaneous predictions for a given y-coordinate on the validation data for the isothermal flow case. This is used as an alternative method to the averaging process, in the hope of reducing the effect of outliers.
- **RMSE Validation Stratified - Mean**: same as the isothermal case, except the data, is replaced with the mean stratified flow predictions.
- **RMSE Validation Stratified - Median**: same as the isothermal case, except the data is replaced with the median stratified flow predictions.

Furthermore, the Mean Absolute Percentage Error (MAPE) will be reported for the final model evaluation on the test data (only). MAPE is defined as,

$$MAPE = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{y_i - \hat{y}_i}{y_i} \right) \times 100. \tag{3.26}$$

The MAPE can be roughly interpreted as the average percentage each prediction of a_{yz} is incorrect by. Since the machine learning model is explicitly trained to minimise the RMSE, the MAPE statistic will only be reported for the test dataa when evaluating the final model, to allow for easier interpretation of the performance.

3.4.2 Generalisability

The machine learning process should produce a model which is accurate on the training, validation and test datasets, for both the isothermal and stratified flow cases. The training data is a subset of the overall dataset on which values for the parameters of the Tensor Basis Neural Network (TBNN) are determined. The validation set is used to select the model, with a particular permutation of hyperparameters. Finally, the test set is used to estimate how good the model would perform in the real-world on unseen data. A predictive model which is only accurate on the training dataset is said to be *overfit* as it has essentially memorised the training data to produce overly optimistic predictions. Ideally, the machine learning model should have high accuracy on the training, validation and test data, that is, it is said to have high generalisability.

Using the properties of the MSE it can be decomposed into what is known as the bias-variance trade-off. While the full derivation is beyond the scope of this thesis, it is possible to use the MSE to derive an equation of the form,

$$MSE = E[(y - \hat{y})^2] = \sigma_\varepsilon^2 + Bias^2[\hat{y}] + Variance[\hat{y}], \tag{3.27}$$

where the terms are defined as follows:

- **Irreducible error (σ_ε^2)**: this is the underlying noise in the data which cannot be improved by further training.
- **Bias**: the lack of flexibility of a particular machine learning algorithm to determine the relationships in the data.
- **Variance**: how sensitive the machine learning algorithm is to a particular dataset. A model which high variance will produce less repeatable results if a different subset of the data were to be used.

It should be noted that the decomposition for MSE has been used as it has a more interpretable result than if the RMSE was decomposed.

The level of bias or variance of a particular predictive model is generally a function of the complexity of the model. For instance, a more complex TBNN model would have a greater number of neurons and more layers than a simpler model. In general, models which are more complex have a higher variance and lower bias and therefore tend to *overfit* the data, memorising the underlying structure of the training including the useless noise. Overly complex models are not generalisable to different datasets, which is not desirable for this study. On the other hand, simpler models tend to *underfit* the data, resulting in an inaccurate model on average. The bias-variance trade-off and its relationship to model complexity are shown in Figure 3.5.

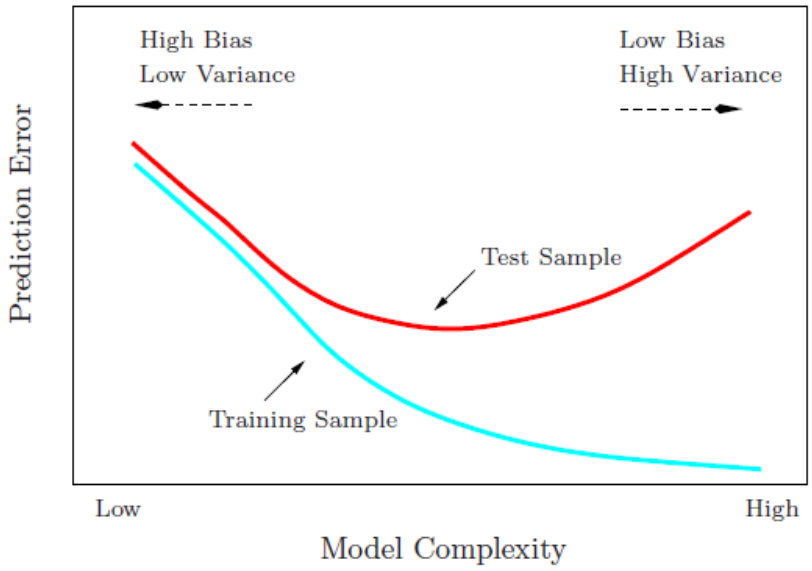


Fig. 3.5 Bias-Variance Trade-off. Adapted from Hastie et al. [20]

Figure 3.5 shows that as the complexity of the model is increased, the prediction error (i.e. RMSE) will decrease on the training data. Therefore, when training the TBNN it is expected that increasing the number of layers and neurons in the network, for example, will improve the accuracy of the model. However, as the complexity of the predictive model is increased, there exists an inflection point for the predictive error on the test sample which represents the real-world unseen data. This is why a validation dataset is used, as it used as a method of estimating test set error, allowing the ideal model complexity to be found. Therefore, the model which performs the best on the validation dataset is used as the final model on the test set. This model should not be overfitted on the training sample and should be generalisable to the test sample.

Finally, as discussed in the introduction, and throughout the literature review, a key outcome for the developed turbulence model should be Galilean invariance. This is the property by which a_{yz}

is not dependent on any coordinate frame of reference. For instance, if the open channel was rotated such that the z-direction was the vertical direction and the x-direction was the streamwise direction, then the machine learning model should accurately predict a_{xz} as it is not coordinate dependent. This form of generalisability will be confirmed by checking that the assumptions of the Explicit Algebraic Stress Model (EASM) are upheld.

3.4.3 Interpretability

Interpretability is the final metric on which the predictive model for the anisotropic Reynolds stress will be evaluated on. One of the main drawbacks with neural network machine learning models, as outlined in the literature review, is its black-box nature. Thus, the coefficients on the parameters in the model cannot be directly interpreted. As a result, associative relationships between the inputs and outputs are not readily available (e.g. "if the first invariant, I_1, is increased by one unit, then we expect that the anisotropic Reynolds stress, a_{yz}, will increase/decrease by X units").

However, a sensitivity analysis can be conducted in the form of Partial Dependence Plotting (PDP). The goal of this analysis would be to understand how the inputs of this machine learning model (i.e. invariants, isotropic basis tensors, wall-based Reynolds number and temperature fluctuation gradient) affect the anisotropic Reynolds stress output, and which of these variables are the most important.

3.5 Limitations

Before presenting the results of this investigation, it is first important to understand the limitations of the methodology proposed throughout this chapter. The first limitation which should be discussed is the hyperparameter selection process outlined in Section 3.3.3. When the hyperparameters are selected, in a process known as hyperparameter optimisation, the proposed methodology is to vary them independently in a manual, iterative approach. While a manual, iterative search is still a widely used method for selecting hyperparameters [4], it is generally better to use a method known as grid search.

Hyperparameter grid search is a method by which distinct permutations of models are constructed, each with different hyperparameters. In such a way, the hyperparameters are not tuned independently which often leads to more accurate results [4]. However, this method is extremely computationally intensive due to a large number of permutations which are available. For instance, if the 7 hyperparameters outlined each had 5 different options, there would be $P(7,5) = 2,520$, different models to train and validate. Furthermore, the Python package used

to train the neural networks in this study, *keras*, does not support grid search when there are multiple inputs and outputs in the TBNN configuration. For these reasons, the manual search method for hyperparameter was deemed to be an acceptable compromise.

In addition, the results presented in the following section are classified as a 'priori' study, using the definitions outlined in Ling et al. [35] and Weatheritt and Sandberg [72]. A 'priori' study is one that the methodology only extends to estimate the anisotropic Reynolds stress, rather than other flow variables such as the velocity and pressure of the fluid. Calculation of these additional flow variables, after inserting the data-developed turbulence model into a RANS solver, would be conducted in a 'posterior' study. As a result, direct comparison to other turbulence models, such as $k-\varepsilon$, is not available as the most commonly used models use an implicit formulation for the Reynolds stress. Nonetheless, the problem outlined in Ling et al. [35] is similar to this study, therefore, their results should provide a reasonable proxy.

Finally, to get a complete understanding of the generalisability of the model developed, it should be tested on several distinct datasets. This may include open channel flows with different Reynolds numbers or varying amounts of stratification. However, the decision was made that statistical techniques, such as confidence intervals and bootstrapping, would provide a reasonable estimate for the generalisation performance of the machine learning model. These techniques are based on large sample assumptions and computational resampling respectively, and therefore replicate multiple investigations without the need for large computational requirements for generating several Large Eddy Simulation datasets.

Chapter 4

Results and Analysis

4.1 Flow Visualisation

4.1.1 Isothermal

After running the Large Eddy Simulation (LES) for the isothermal open channel flow case, the results were visualised using the ParaView software [1]. Figure 4.1 shows the w-velocity at time unit $t = 30$ and Figure 4.2 shows the vorticity at $t = 30$, where vorticity is a measure of the local rotation of the fluid.

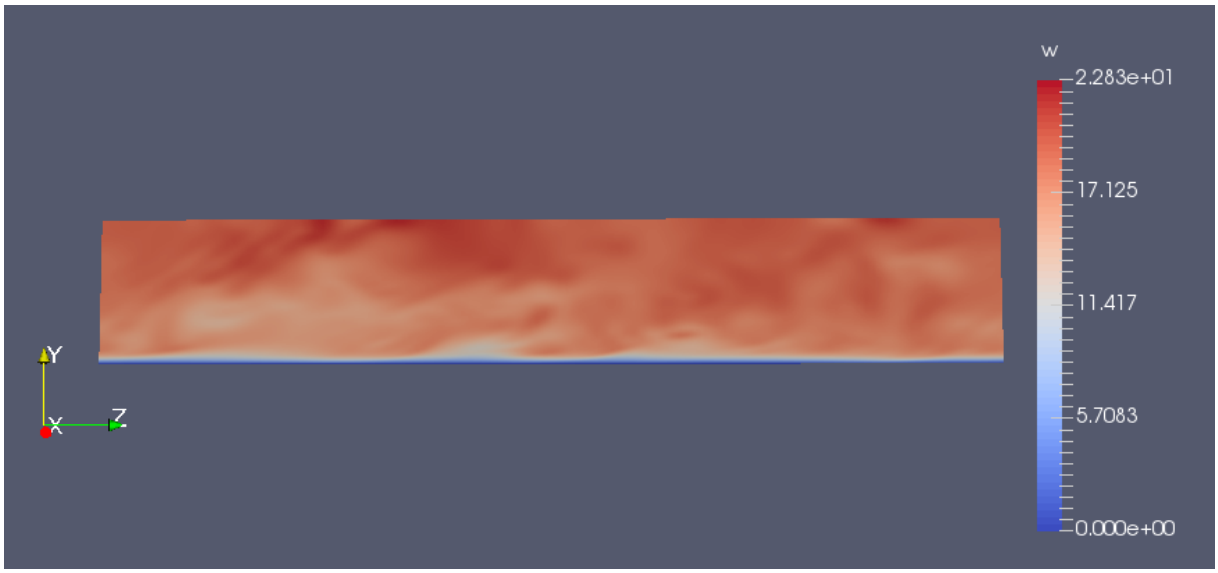


Fig. 4.1 Isothermal: w-velocity at t = 30

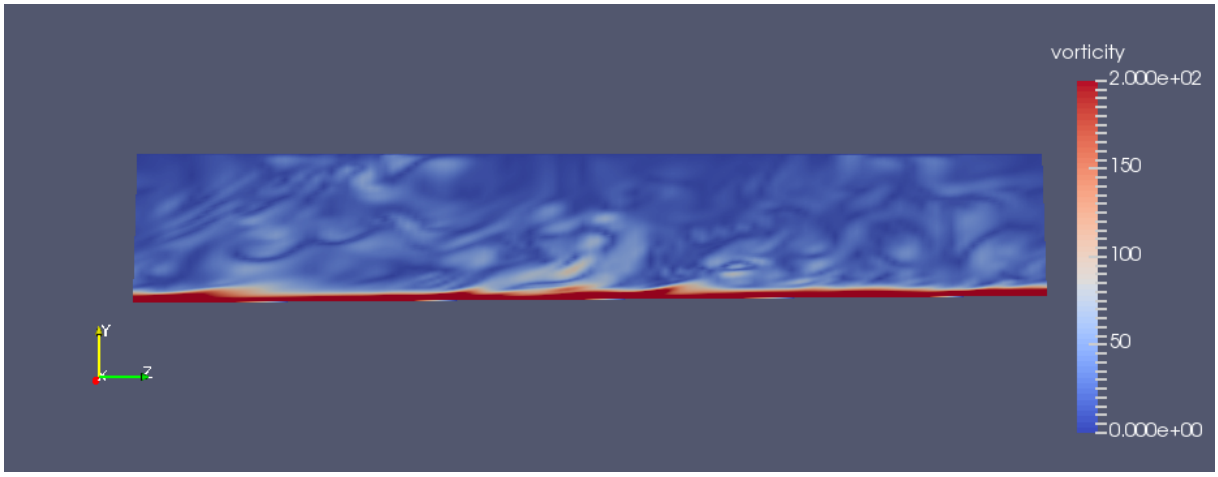


Fig. 4.2 Isothermal: Vorticity at t = 30

Further flow visualisations at $t = 40$ and $t = 50$ time units can be found in Appendix A.1.

4.1.2 Stratified

Likewise, Figures 4.3-4.4 show the w-velocity and vorticity for the stratified flow case at $t = 30$.

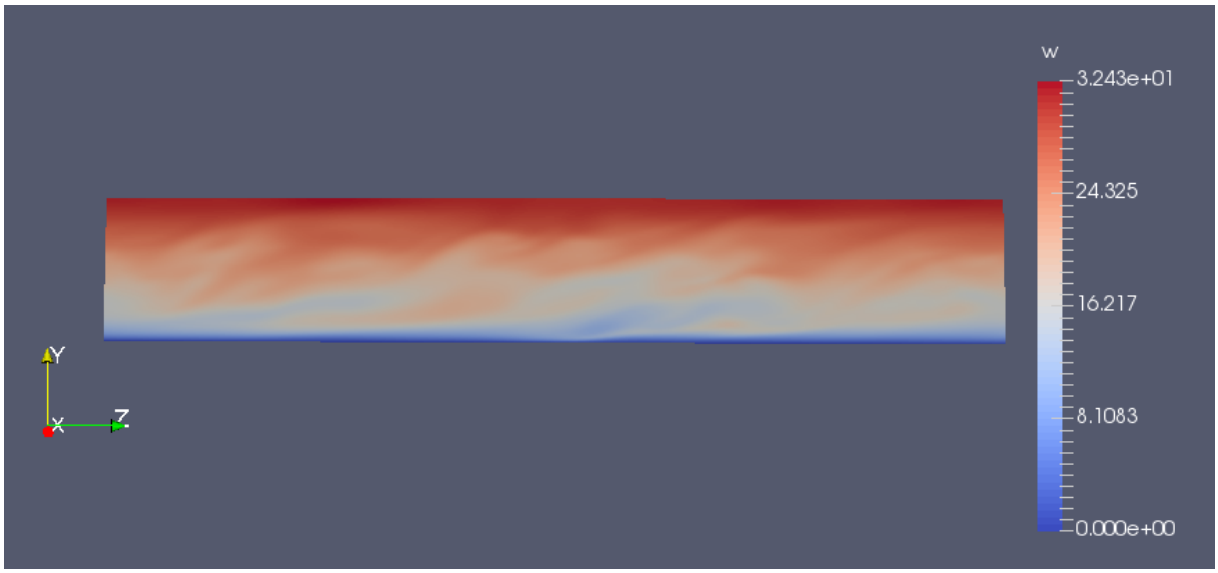


Fig. 4.3 Stratified: w-velocity at t = 30

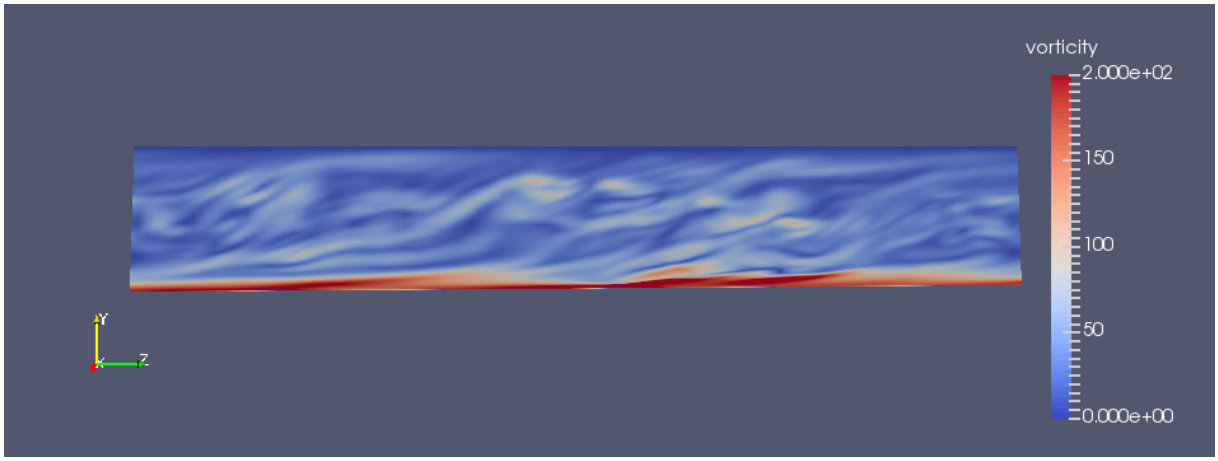


Fig. 4.4 Stratified: Vorticity at t = 30

Since the stratified flow case also varies in temperature, Figure 4.5 shows how the non-dimensional temperature fluctuation, ϕ, varies through the channel at $t = 30$.

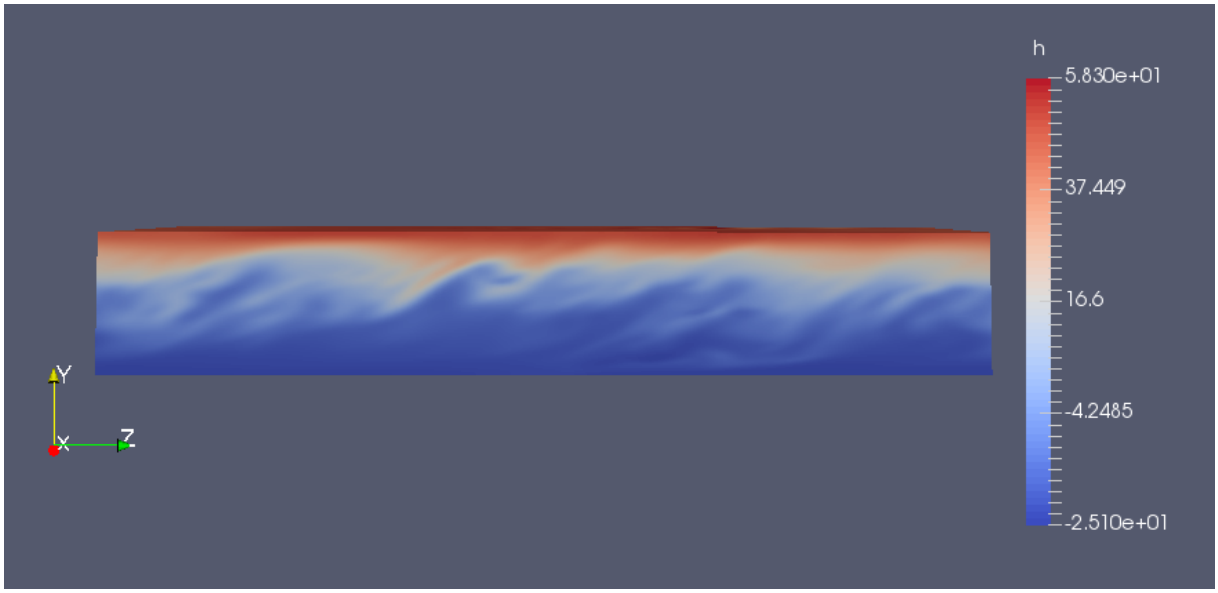


Fig. 4.5 Stratified: ϕ at t = 30

4.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the first stage conducted in any machine learning project. EDA is used to: gain a high-level understanding of the dataset, identify outliers, and predict which variables might be the most useful in the analysis. The dependent variable, a_{yz}, will be first considered individually in a univariate analysis. Then the variables which might be most important for the machine learning algorithm will be identified using a correlation plot. The

independent variables deemed to be the most likely to contribute to the accuracy of the algorithm will then be analysed univariately, as well as bivariately in relation to the dependent variable.

4.2.1 Analysis of Anisotropic Reynolds Stress

Firstly, Figure 4.6 shows how the anisotropic Reynolds stress on the y-z plane varies with the vertical (y-)direction. This has been separated by isothermal and stratified data.

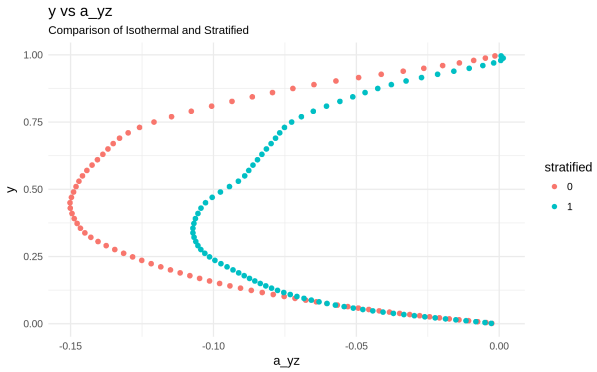


Fig. 4.6 y vs a_{yz}
Separated by Isothermal and Stratified

Here, Figure 4.6 shows that the minimum value of a_{yz} occurs approximately $y = 0.4$ for the isothermal (red) and at $y = 0.3$ for the stratified (blue) flows. Clearly, the isothermal flow has a smaller minimum value. Nonetheless, the maximum values of a_{yz} occur at the wall ($y = 0$) and at the free surface ($y = 1$) where $a_{yz} \approx 0$.

Unsurprisingly, the scatter plot in Figure 4.6 suggests a much more complex flow pattern for the stratified flow. The red stratified curve is less smooth than the blue isothermal curve, in particular around the middle of the flow. These differences may be due to the complex interaction effects of the buoyancy-driven motions (caused by temperature differences) and turbulent eddies, thus creating usual patterns for a_{yz} which may prove to be a challenge for the machine learning model.

4.2.2 Correlation with Inputs

As there are 17 input variables in the data; the five invariants, the ten isotropic basis tensors, the wall-based Reynolds number, and the gradient of non-dimensional temperature fluctuation, it is important to first understand which of these variables will be most relevant in predicting a_{yz}. Without first analysing the bivariate relationships between the input and output, it is possible to use a correlation matrix as a *priori* study. The input variables with the highest correlation to the

anisotropic Reynolds stress will likely be the most important for the machine learning model and will inform the variables analysed in this section. Figure 4.7 shows a correlation heat map, which is a plot showing the correlation between variables in matrix form.

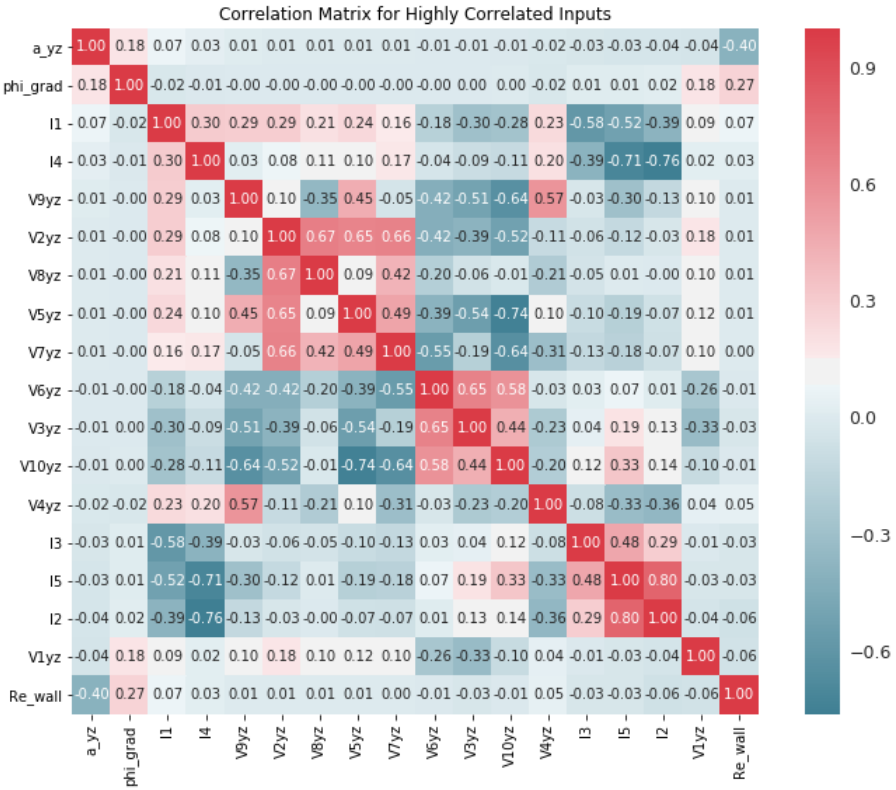


Fig. 4.7 Input Correlation to a_{yz} in descending order

Interestingly, Figure 4.7 shows the input variables with the greatest magnitude of correlation to a_{yz} are the first set of inputs to the Tensor Basis Neural Network (TBNN) as per Figure 3.4 in Section 3.3.2. They are, in decreasing order of magnitude: Re_{wall}, phi_grad, I_1, I_2, I_5, I_3, I_4. This is significant as these variables will undergo the non-linear transformation that the TBNN provides to map the underlying relationships in the data. The non-linear transformations of these input variables will find the scalar coefficients, $g^{(n)}$, which will be linearly combined with the ten isotropic basis tensors to find the prediction for the anisotropic Reynolds stress.

Nonetheless, Figure 4.7 shows that the strength of these correlations is quite weak. Thus, a more complex TBNN might be required to find the underlying relationships within the data, which in turn may lead to overfitting issues as per the bias-variance trade-off previously described.

4.2.3 Univariate Analysis

A univariate analysis for the input variables was conducted based on the findings in the previous section. In this section, only the input variables of Re_{wall}, phi_grad and I_1 were analysed as they had the highest linear correlation with a_{yz}. The univariate analysis of the remaining input variables, including the ten isotropic basis tensors, can be found in Appendix A.

Nonetheless, Figure 4.8 shows how the wall-based Reynolds number changes with the vertical direction, separated for the isothermal (red) and stratified (red) flow cases. Close to the lower boundary of the channel, the values are almost identical. However, as the y-coordinate increases, and the heat effects which originate from $y = 1$ become stronger, the values for Re_{wall} depart. Furthermore, Figure 4.9 shows how phi_grad varies with the y-direction. However, since the isothermal flow case has constant phi_grad = 0, there are only values for the stratified case.

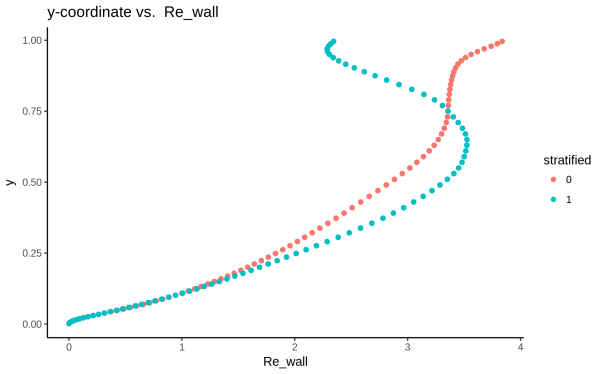


Fig. 4.8 y-coordinate vs. Re_{wall}

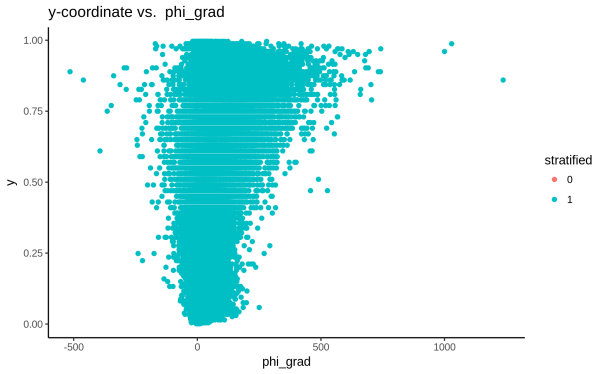


Fig. 4.9 y-coordinate vs. phi_grad

Moreover, an interesting pattern emerges for the first of five invariants, I_1. Here, Figure 4.10 shows an hourglass shape for both the isothermal (red) and stratified (blue) cases. This shape is a result of outliers being present at $y \approx 1$ for both the isothermal and stratified flow cases, as well as at $y \approx 0.4$ for the isothermal flow and $y \approx 0.3$ for the stratified flow. The hourglass pattern was also true for the other four invariants, and was shown to have a significant effect on the accuracy of the machine learning model, which will be described in the discussion in Section 5.3.

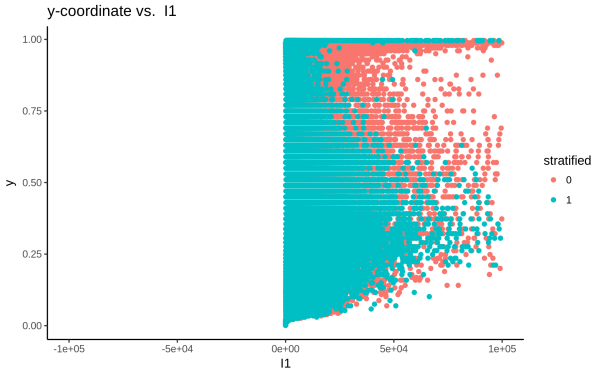


Fig. 4.10 y-coordinate vs. I_1

4.2.4 Bivariate Analysis

Following the univariate analysis of the input variables in the previous section, this study can be extended to the bivariate case. Here, the dependent variable, a_{yz}, will be plotted against the same three input (independent) variables from the previous section with the aim of finding clear linear or non-linear relationships.

The two variables Re_{wall} and phi_grad are plotted with a_{yz} as shown in Figures 4.11 and 4.12 respectively. In these figures, there is no clear linear or non-linear relationship which can be drawn between a_{yz} and these two variables.

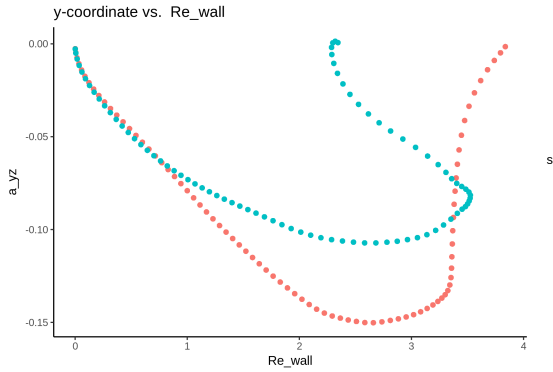


Fig. 4.11 a_{yz} v.s. Re_{wall}

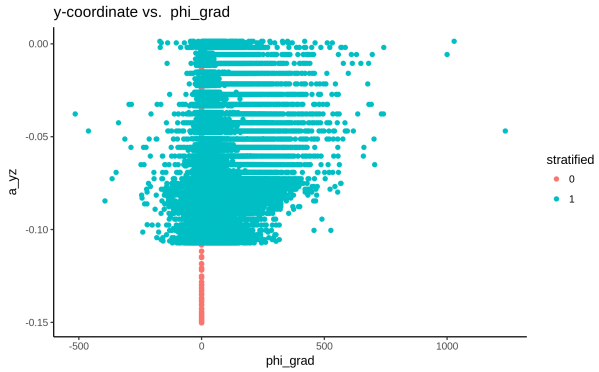


Fig. 4.12 a_{yz} v.s. *phi_grad*

On the other hand, the bivariate plot for I_1 in Figure 4.13 highlights an interesting hourglass shape, which is very similar to the univariate case previously.

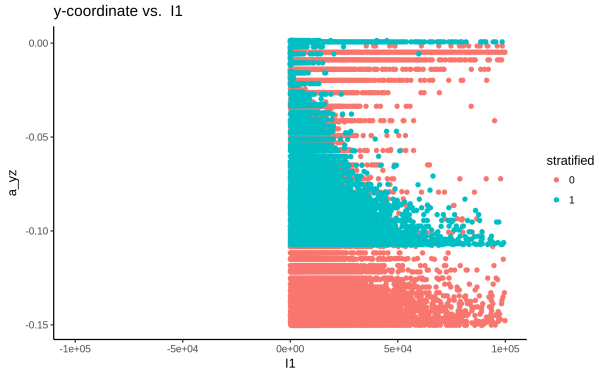


Fig. 4.13 a_{yz} v.s. I_1

This time, however, the outliers of I_1 occur at two locations on this plot. Firstly, at approximately $a_{yz} = 0$, which as previously shown, was at the upper ($y \approx 1$) and lower ($y \approx 0$) boundaries of the open channel. The other is at the respective minimums of the isothermal (red) and stratified (blue) flow cases. When comparing to the univariate plot in the previous section (Figure 4.10), this again occurs at $y \approx 0.4$ for the isothermal flow and $y \approx 0.3$ for the stratified flow. Again, these outliers for I_1, which is also true for the other four invariants, was shown to have a significant effect on the accuracy of the developed turbulence model, as discussed in Section 5.3.

4.3 Modelling

As described in the methodology, there are a number of hyperparameters that must be selected for the neural network machine learning algorithm. Informally, the hyperparameters are all the different levers which can be pulled by the modeller to make the final machine learning model as accurate as possible.

The various accuracy metrics reported throughout the modelling phase were outlined in the methodology (see Section 3.4.1). Each of these RMSE metrics will provide insight into the performance of any given model, each with different hyperparameter values. However, as previously described, the single model, which minimises the RMSE on the combined (isothermal + stratified) validation data will be deemed to be the most appropriate for final evaluation.

4.3.1 Baseline Model

The baseline model is used as the starting point for the hyperparameter selection process. The values of the hyperparameters for the baseline model were the default values for a neural network trained by the *keras* Python package. These values were assumed to be a good starting point for the selection process and are shown in Table 4.1.

Table 4.1 Baseline Model

	Model 1
Layer structure	50, 32, 10
Data transformation	StandardScaler
Epochs	200
Batch size	32
Optimiser	adam
Weight initilisation	glorot_uniform
Dropout regularisation	none

Using the baseline model, the results are as per Table 4.2. Here, the metric by which the models will be primarily evaluated and selected against, RMSE Validation Combined, is highlighted in bold. If any model has an RMSE which is lower than a value of 0.0142, then the value of the hyperparameters of this particular model will become the new best model. This is because the goal is to minimise the RMSE on the validation data, therefore, a model which is more accurate will have a lower RMSE.

Table 4.2 Baseline Model Results

	Model 1
RMSE Training Combined	0.0077
RMSE Validation Combined	**0.0142**
RMSE Validation Isothermal - Mean	0.0124
RMSE Validation Isothermal - Median	0.0149
RMSE Validation Stratified - Mean	0.0052
RMSE Validation Isothermal - Median	0.0036

Furthermore, the mean (orange) and median (green) predictions of a_{yz} for any given y-coordinate were compared against the LES values. This is shown in Figure 4.14 for the isothermal predictions and on Figure 4.15 for the stratified predictions. These were both done using the validation data only.

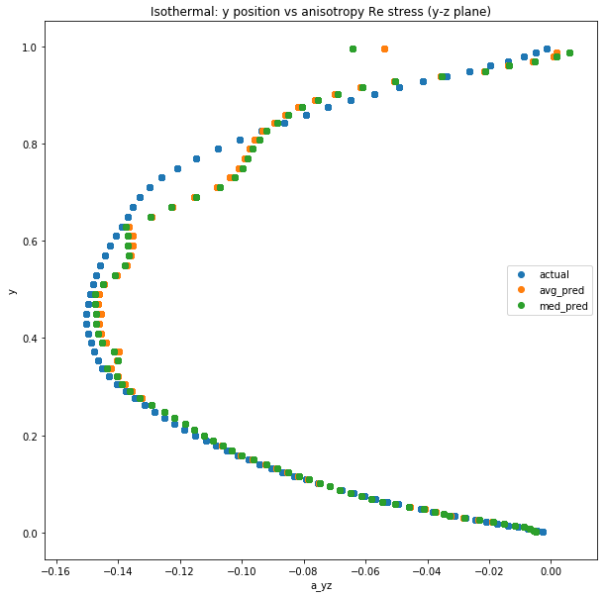


Fig. 4.14 Isothermal: Average and Median Validation Predictions - Model 1

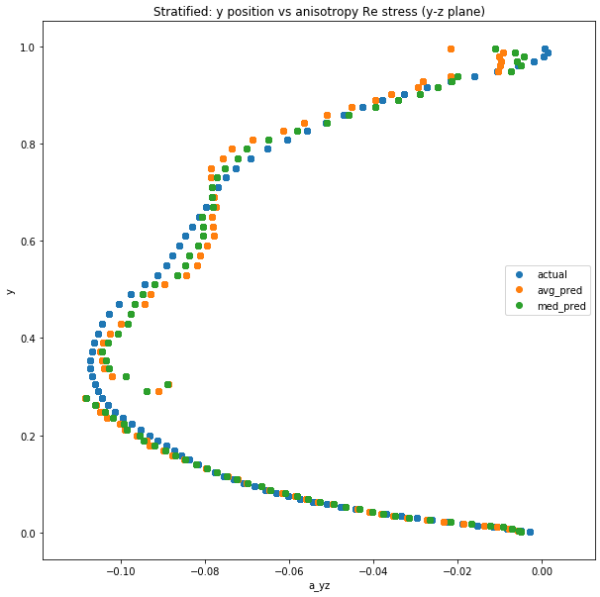


Fig. 4.15 Stratified: Average and Median Validation Predictions - Model 1

If the predictions from the machine learning model were completely accurate, there would be no difference either the orange or green predictions when compared to the blue LES values. However, in both cases, further hyperparameter optimisation is clearly required to produce a more accurate machine learning model.

4.3.2 Layer Structure

The layer structure hyperparameter refers to the number of layers within the neural network and the number of neurons for each given layer. For the Tensor Basis Neural Network (TBNN) there are two distinct sections of the network: the section used to determine the scalar coefficients of the Explicit Algebraic Stress Model (EASM) and the section used to linearly combine the scalar coefficients with the isotropic basis tensors. Further details of the TBNN are provided in Section 3.3.2.

Based on this definition, it is only possible to change the structure immediately following input layer 1. This is the input layer with the variables of the five invariants, wall-based Reynolds number and the gradient of non-dimensional temperature fluctuation, used to determine the scalar coefficients of the EASM. According to the work conducted by Cheng et al. [5], there are a number of considerations when deciding how many layers and neurons in a neural network. For instance, a neural network with only a single layer, with sufficiently many neurons, can approximate any function given enough data. However, having too many neurons will result in the neural network model memorising the training data leading to overfitting.

To make the model more generalisable, additional layers can be added to the network. On the other hand, a network with many layers and many neurons creates a model which is overly complex. As such, the time it takes to train such a model becomes prohibitively large. The various permutations of the models tested, varying only the structure immediately following the first input layer of the TBNN, is shown in Tables 4.3 and 4.4. Noting that a layer structure of '50, 50, 10' means there are three layers with 50 neurons in the first hidden layer, 50 in the second and 10 in the final (output) layer.

Table 4.3 Layer Structure Models 2-6

	Model 2	**Model 3**	**Model 4**	**Model 5**	**Model 6**
Layer structure	50, 50, 10	75, 50, 10	75, 75, 10	50, 32, 16, 10	50, 32, 16, 12, 10
Data transformation	StandardScaler	StandardScaler	StandardScaler	StandardScaler	StandardScaler
Epochs	200	200	200	200	200
Batch size	32	32	32	32	32
Optimiser	adam	adam	adam	adam	adam
Weight initilisation	glorot_uniform	glorot_uniform	glorot_uniform	glorot_uniform	glorot_uniform
Dropout regularisation	none	none	none	none	none

Table 4.4 Layer Structure Models 7-10

	Model 7	**Model 8**	**Model 9**	**Model 10**
Layer structure	32, 20, 16, 10	50, 50, 50, 10	32, 32, 32, 10	75, 75, 75, 10
Data transformation	StandardScaler	StandardScaler	StandardScaler	StandardScaler
Epochs	200	200	200	200
Batch size	32	32	32	32
Optimiser	adam	adam	adam	adam
Weight initilisation	glorot_uniform	glorot_uniform	glorot_uniform	glorot_uniform
Dropout regularisation	none	none	none	none

Based on these models with various structures, the results are shown in Table 4.5.

Table 4.5 Layer Structure Results

	Model 2	**Model 3**	**Model 4**	**Model 5**	**Model 6**	**Model 7**	**Model 8**	**Model 9**	**Model 10**
RMSE Training Combined	0.0083	0.0067	0.0074	0.0061	0.0062	0.0081	0.0058	0.0067	**0.0056**
RMSE Validation Combined	0.3559	0.0235	0.1348	0.0088	0.0441	0.0383	**0.0061**	0.1034	0.0111
RMSE Validation Isothermal - Mean	0.0260	0.0262	0.0268	0.0037	0.0507	0.0453	**0.0021**	0.0043	0.0094
RMSE Validation Isothermal - Median	0.0102	0.0262	0.0141	0.0037	0.0508	0.0453	**0.0020**	0.0046	0.0109
RMSE Validation Stratified - Mean	0.0198	0.0156	0.0092	0.0055	0.0351	0.0279	**0.0032**	0.0101	0.0036
RMSE Validation Isothermal - Median	0.0212	0.0158	0.0083	0.0047	0.0352	0.0279	**0.0024**	0.0034	0.0029

The best performing model for the model selection metric, RMSE validation combined, is Model 8 as it has the lowest score ($RMSE = 0.0061$). This is an improvement of 57% from the baseline model in the previous section. Furthermore, this model performs the best for each of the other metrics, except the combined training RMSE where Model 10 performs the best. Model 10 is an example of a model which has been overfit to the training data. While it performs the very well on the data that the machine learning model is trained on ($RMSE = 0.0056$), it does not generalise very well to the validation data where the RMSE approximately doubles ($RMSE = 0.0111$).

Moreover, the mean and median predictions for Model 8 can also be visualised in comparison to the LES results. This is shown in Figure 4.16 for the isothermal flow and Figure 4.17 for the stratified flow.

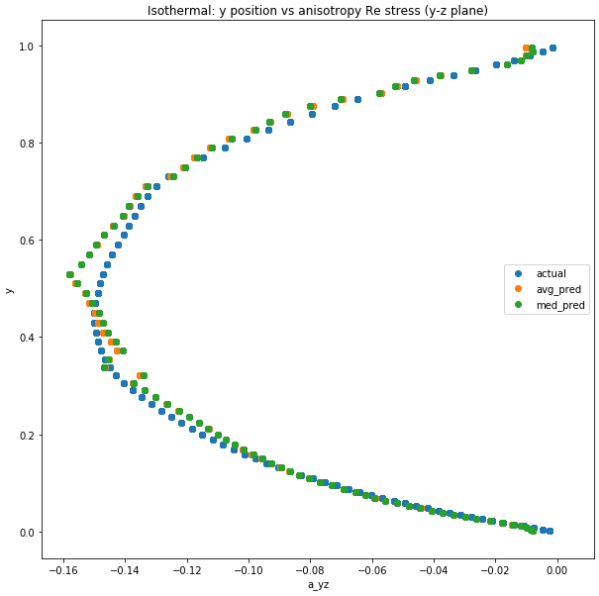


Fig. 4.16 Isothermal: Average and Median Validation Predictions - Model 8

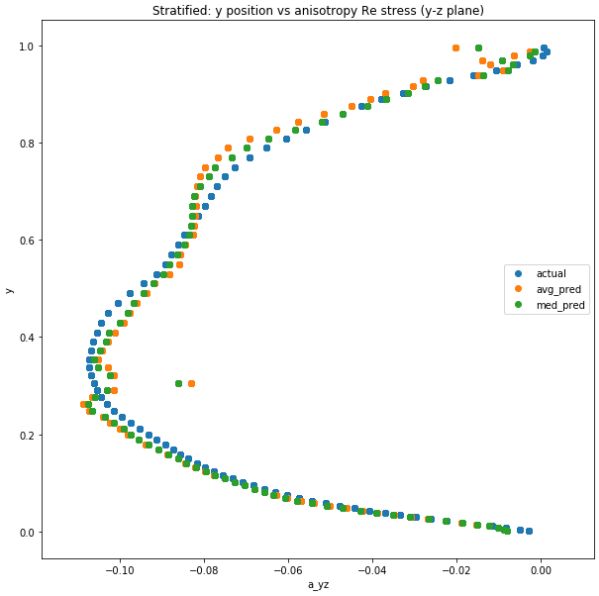


Fig. 4.17 Stratified: Average and Median Validation Predictions - Model 8

When compared to the baseline model in Figures 4.14 and 4.15 of the previous section, there is a visual improvement of the mean and median predictions of a_{yz}. While there is minimal change closer to the lower boundary of the channel, the mean (orange) and median (green) predictions of a_{yz} at the centre are each closer to the LES (blue) values. However, there is clearly still room for improvement with large discontinuities for both the isothermal and stratified plots at approximately $y = 0.3$ and $y = 1.0$.

Nonetheless, since Model 8 is the best performing model, and is an improvement over the baseline as it has a lower validation RMSE for the combined data, its layer structure will be used for all following models. This layer structure is 50, 50, 50, 10.

4.3.3 Data Transformation

By making the input data more normally distributed, it is possible to improve the accuracy and training speed of a neural network [56]. Data transformation was achieved using one of five preprocessing classes in the Python package *scikit-learn* [45]:

1. **StandardScaler**: standardise input variables by subtracting the mean and dividing to by the variance. This results in a variable having approximately a mean of 0 and a variance of 1.

2. **MinMaxScaler**: rescale an input variable between a range of 0 and 1. This is an alternative to StandardScaler.

3. **MaxAbsScaling**: rescale the input variable between a range of $-\infty$ and 1. This does not shift the data closer together like the MinMaxScaler and therefore sparsity is preserved.

4. **RobustScaler**: subtract the median and divide by the interquartile range for any given input variable. This method of scaling is less affected by outliers.

5. **NoScaling**: no transformation is done to the input variables.

Using the updated TBNN structure from the previous section, the models tested to evaluate the input data transformations are as per Table 4.6.

Table 4.6 Data Transformation Models

	Model 8	**Model 11**	**Model 12**	**Model 13**	**Model 14**
Layer structure	50, 50, 50, 10	50, 50, 50, 10	50, 50, 50, 10	50, 50, 50, 10	50, 50, 50, 10
Data transformation	StandardScaler	MinMaxScaler	MaxAbsScaler	RobustScaler	NoScaling
Epochs	200	200	200	200	200
Batch size	32	32	32	32	32
Optimiser	adam	adam	adam	adam	adam
Weight initilisation	glorot_uniform	glorot_uniform	glorot_uniform	glorot_uniform	glorot_uniform
Dropout regularisation	none	none	none	none	none

The results in Table 4.7 show that Model 8, which uses the StandardScaler, is the best performing model for all metrics. Interestingly, it was not possible to compute the RMSE for models 13 and 14, which used the RobustScaler and NoScaling respectively. These data transformation methods produced predictions for a_{yz} which were close to infinity, resulting in a model which was far too inaccurate to be able to calculate the RMSE scores.

Table 4.7 Data Transformation Results

	Model 8	**Model 11**	**Model 12**	**Model 13**	**Model 14**
RMSE Training Combined	**0.0058**	0.0096	0.0261	NA	NA
RMSE Validation Combined	**0.0061**	0.0120	0.0267	NA	NA
RMSE Validation Isothermal - Mean	**0.0021**	0.0059	0.0274	NA	NA
RMSE Validation Isothermal - Median	**0.0020**	0.0059	0.0277	NA	NA
RMSE Validation Stratified - Mean	**0.0032**	0.0109	0.0108	NA	NA
RMSE Validation Isothermal - Median	**0.0024**	0.0104	0.0097	NA	NA

Since the Model 8 is again the most accurate model, the plot of the anisotropic Reynolds stress on the y-z plane for each given y-position is unchanged from Figures 4.16 and 4.17 in the previous section. Hence, it is concluded that that the StandardScaler data transformation is the most appropriate for further analysis.

4.3.4 Epochs

When training a model, the training data is passed through the neural network many times. Each time the entire training set is passed through the network it is referred to as one epoch. Generally, with each epoch, the combined training RMSE decreases down to some optimal level. At this threshold, increasing the number iterations results in only marginally increased performance on the validation data. Therefore, the goal is to find the number of epochs where this threshold is met on the validation data. The various values of the number of epochs are shown in Table 4.8.

Table 4.8 Epochs Models

	Model 8	**Model 15**	**Model 16**
Layer structure	50, 50, 50, 10	50, 50, 50, 10	50, 50, 50, 10
Data transformation	StandardScaler	StandardScaler	StandardScaler
Epochs	200	300	500
Batch size	32	32	32
Optimiser	adam	adam	adam
Weight initilisation	glorot_uniform	glorot_uniform	glorot_uniform
Dropout regularisation	none	none	none

The results are shown in Table 4.9 below. The best performing model is Model 15, which uses 300 epochs and produces a RMSE of 0.0057 for the combined validation data. It appears that 300 epochs is the threshold where further model training does not yield an improved result. Model

16, which uses 500 epochs, results in only a marginal improvement for the combined training RMSE, whereas, the combined validation RMSE actually increases.

Table 4.9 Epochs Results

	Model 8	**Model 15**	**Model 16**
RMSE Training Combined	0.0058	0.0054	**0.0053**
RMSE Validation Combined	0.0061	**0.0057**	0.0168
RMSE Validation Isothermal - Mean	0.0021	**0.0014**	0.0147
RMSE Validation Isothermal - Median	0.0020	**0.0016**	0.0159
RMSE Validation Stratified - Mean	0.0032	**0.0029**	0.0044
RMSE Validation Isothermal - Median	0.0024	**0.0022**	0.0038

Using Model 15, the plots in Figure 4.18 for the isothermal predictions and Figure 4.19 can be created. These plots clearly demonstrate a large improvement over Model 8 (Figures 4.16 and 4.17) showing the influence of the number of epochs on the accuracy of the machine learning model. The mean and median predictions for the isothermal validation data in Figure 4.18 almost mirror the blue LES values for a_{yz}. While not perfect, Figure 4.19 showing the stratified predictions for Model 15 visually shows fewer discontinuities at the centre of the flow, with the issues the utmost layer almost eliminated.

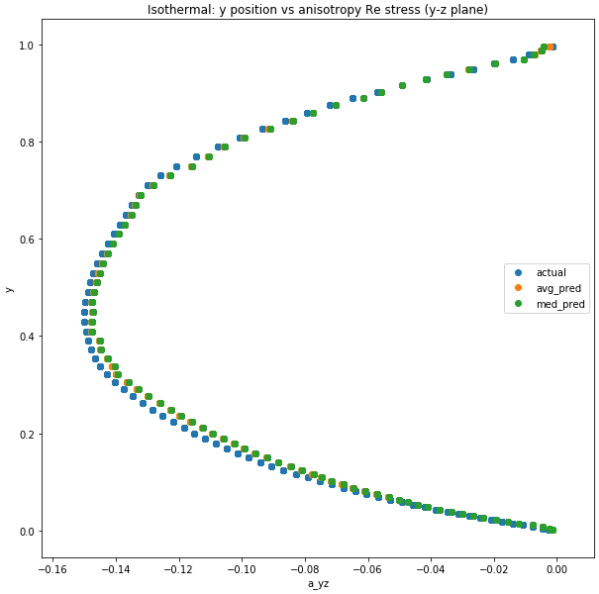


Fig. 4.18 Isothermal: Average and Median Validation Predictions - Model 15

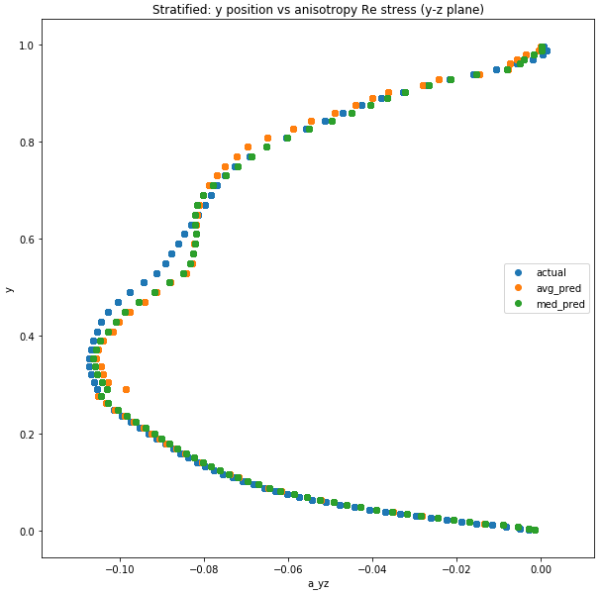


Fig. 4.19 Stratified: Average and Median Validation Predictions - Model 15

Hence, based on the results in this section, 300 epochs is selected as the optimal number for further training.

4.3.5 Batch Size

The batch size is the number of data points which are passed through a neural network at any given time during the training process. These are the samples which are used by the optimisation algorithm to determine how the output is related to the inputs (e.g. weights/bias between the neurons). In general, the batch size is less than the total training dataset. Smaller batch sizes require less memory and are faster to train. However, reducing the batch size can result in a less accurate estimation of the gradient of the error metric (i.e. the goal is to minimise the mean squared error which is achieved by using an optimisation process known as gradient descent) [22]. The various models trialled, with varying batch sizes, is shown in Table 4.10.

Table 4.10 Batch Size Models

	Model 15	**Model 17**	**Model 18**
Layer structure	50, 50, 50, 10	50, 50, 50, 10	50, 50, 50, 10
Data transformation	StandardScaler	StandardScaler	StandardScaler
Epochs	300	300	300
Batch size	32	100	16
Optimiser	adam	adam	adam
Weight initilisation	glorot_uniform	glorot_uniform	glorot_uniform
Dropout regularisation	none	none	none

The results are shown in Table 4.11. The model which performs the best is again Model 15 which uses the default batch size of 32. Even though Model 17 fits the best to the training data, it is overfit and therefore it does not generalise well to the combined validation data. Therefore, a batch size of 32 was selected.

Table 4.11 Batch Size Results

	Model 15	**Model 17**	**Model 18**
RMSE Training Combined	0.0054	**0.0050**	0.0055
RMSE Validation Combined	**0.0057**	0.0126	0.0087
RMSE Validation Isothermal - Mean	**0.0014**	0.0080	0.0037
RMSE Validation Isothermal - Median	**0.0016**	0.0078	0.0037
RMSE Validation Stratified - Mean	**0.0029**	0.0084	0.0054
RMSE Validation Isothermal - Median	**0.0022**	0.0075	0.0051

4.3.6 Optimiser

An optimiser, or optimisation algorithm, is used to determine the internal parameters of a neural network during the training process. While the technicalities of the various optimisers are beyond the scope of this thesis, each of these algorithms generally converges to different results. Therefore, it is important to try several different variations of optimisation algorithms to ensure the most accurate result is obtained. As such, Table 4.12 outlines the various models, each with a different optimiser, used in the hyperparameter selection process.

Table 4.12 Optimiser Models

	Model 15	**Model 19**	**Model 20**	**Model 21**
Layer structure	50, 50, 50, 10	50, 50, 50, 10	50, 50, 50, 10	50, 50, 50, 10
Data transformation	StandardScaler	StandardScaler	StandardScaler	StandardScaler
Epochs	300	300	300	300
Batch size	32	32	32	32
Optimiser	adam	nadam	AdaGrad	AdaDelta
Weight initilisation	glorot_uniform	glorot_uniform	glorot_uniform	glorot_uniform
Dropout regularisation	none	none	none	none

The results for the use of various optimisers is shown in Table 4.13. Clearly, the results highlight that the use of the *adam* optimisation algorithm in Model 15 is the optimal choice. This is unsurprising as this optimiser generally performs well on many types of problems and therefore was selected as the default optimisation algorithm for this reason [22].

Table 4.13 Optimiser Results

	Model 15	**Model 19**	**Model 20**	**Model 21**
RMSE Training Combined	**0.0054**	0.0067	0.0852	0.0181
RMSE Validation Combined	**0.0057**	0.0441	1.4095	0.1467
RMSE Validation Isothermal - Mean	**0.0014**	0.0515	0.0665	0.0144
RMSE Validation Isothermal - Median	**0.0016**	0.0515	0.0283	0.0137
RMSE Validation Stratified - Mean	**0.0029**	0.0347	0.0104	0.0066
RMSE Validation Stratified - Median	**0.0022**	0.0347	0.0122	0.0056

Therefore, the *adam* optimisation algorithm in Model 15 will be selected from this list of candidate models.

4.3.7 Weight Initilisation

As described in Section 3.3.1, a key component of neural networks are the weights between the neurons. Prior to the training phase, initial values must be provided to the weights which dictate where on the optimisation process starts. Changing the starting location on this hyperplane will affect the minimum (local) error the optimisation algorithm will find [22]. Hence, different starting weights will influence the final RMSE obtained. Table 4.14 outlines the different weight initialisations trialled in this study. Again, the details of the different weight initialisations are beyond the scope of this thesis.

Table 4.14 Weight Initilisation Models

	Model 15	**Model 22**	**Model 23**	**Model 24**
Layer structure	50, 50, 50, 10	50, 50, 50, 10	50, 50, 50, 10	50, 50, 50, 10
Data transformation	StandardScaler	StandardScaler	StandardScaler	StandardScaler
Epochs	300	300	300	300
Batch size	32	32	32	32
Optimiser	adam	adam	adam	adam
Weight initilisation	glorot_uniform	random_normal	lecun_uniform	he_normal
Dropout regularisation	none	none	none	none

Table 4.15 shows the RMSE results for the various weight initialisations. Based on the primary selection metric, RMSE of the combined validation data, Model 15 which uses the *glorot_uniform* initialisation is the chosen hyperparameter value. However, it is interesting to see Model 23, which uses *lecun_uniform*, is the highest performer for several metrics. Unfortunately, however, it appears as if this model has overfit to the training data, particularly for the isothermal flow, and therefore performs poorly for this flow condition.

Table 4.15 Weight Initilisation Results

	Model 15	**Model 22**	**Model 23**	**Model 24**
RMSE Training Combined	0.0054	0.0054	**0.0051**	0.0064
RMSE Validation Combined	**0.0057**	0.0072	0.0209	0.0444
RMSE Validation Isothermal - Mean	**0.0014**	0.0020	0.0020	0.0517
RMSE Validation Isothermal - Median	**0.0016**	0.0020	0.0019	0.0518
RMSE Validation Stratified - Mean	**0.0029**	0.0036	**0.0029**	0.0343
RMSE Validation Stratified - Median	0.0022	0.0022	**0.0021**	0.0344

Nonetheless, Model 15, which uses the *glorot_uniform* initialisation is selected for the final phase of hyperparameter selection of dropout regularisation.

4.3.8 Dropout Regularisation

Finally, dropout regularisation is a technique used to reduce the complexity of neural network models with the hope of preventing overfitting to the training data. Dropout works by randomly ignoring neurons in the network, meaning that they no longer contribute to the final prediction of the output variable. Doing so prevents the neural network from becoming over-reliant on individual neurons which should prevent overfitting from occurring. Table 4.16 shows the different combinations of dropout regularisation attempted in this thesis. For example, the notation '0.2, 0.2, 0, 0' implies that a neuron in either the first or second hidden layer of the network is dropped with a probability of 20%.

Table 4.16 Dropout Regularisation Models

	Model 15	**Model 25**	**Model 26**	**Model 27**
Layer structure	50, 50, 50, 10	50, 50, 50, 10	50, 50, 50, 10	50, 50, 50, 10
Data transformation	StandardScaler	StandardScaler	StandardScaler	StandardScaler
Epochs	300	300	300	300
Batch size	32	32	32	32
Optimiser	adam	adam	adam	adam
Weight initilisation	glorot_uniform	glorot_uniform	glorot_uniform	glorot_uniform
Dropout regularisation	none	0.2, 0.2, 0, 0	0.2, 0, 0, 0	0, 0.2, 0, 0

Interesting, Table 4.17 below shows that dropout regularisation yields no performance improvement over the previous best model. The lack of performance gain is likely an indication that the TBNN structure of Model 15 is not overly complex and should be applicable enough to various types of fluid flow conditions.

Table 4.17 Dropout Regularisation Results

	Model 15	**Model 25**	**Model 26**	**Model 27**
RMSE Training Combined	**0.0054**	0.0201	0.0080	0.0146
RMSE Validation Combined	**0.0057**	31.8053	0.0127	0.0154
RMSE Validation Isothermal - Mean	**0.0014**	2.4210	0.0115	0.0111
RMSE Validation Isothermal - Median	**0.0016**	0.0323	0.0114	0.0111
RMSE Validation Stratified - Mean	**0.0029**	0.0159	0.0101	0.0133
RMSE Validation Stratified - Median	**0.0022**	0.0155	0.0091	0.0131

Therefore, no dropout regularisation will be used, with Model 15 having the best performance with the lowest combined validation RMSE.

4.4 Model Evaluation

Based on the model selection findings in Section 4.3, Model 15 was chosen as the model which best represents a_{yz}. The full details of the implementation of this model are outlined in Appendix B; including the final selection of hyperparameters in Table B.1, the number of parameters in the model in Figure B.1 and a visualisation of the TBNN structure in Figure B.2.

Nonetheless, this particular model selected is finally evaluated on the test data. The test data is used as the best representation for unseen 'real-world' fluid data and is used only once to evaluate the model's final performance. On the other hand, the training and validation data is used many times, such as in Section 4.3, and therefore does not provide a good estimate of the model's real-world performance. In this investigation, the machine learning model for the anisotropic Reynolds stress on the y-z plane is evaluated with respect to its instantaneous predictions, mean and median predictions, its generalisability and its interpretability.

4.4.1 Performance Metrics

When the predictions for a_{yz} are obtained on the test data the performance metrics are as per Table 4.18. Noting that the mean absolute percentage error (MAPE) has been included, now that model selection has been finalised.

Table 4.18 Final Performance Metrics on Test Data

	RMSE	**MAPE (%)**
Combined - Instantaneous	0.0067	30.16
Isothermal - Mean	0.0025	16.11
Isothermal - Median	0.0025	16.68
Stratified - Mean	0.0030	19.45
Stratified - Median	0.0023	14.22

Table 4.18 highlights the success of the model training process. Firstly, the results in Table 4.18, which use the test data, are similar to those obtained when the validation data was used in Table 4.17. The similarity in results is an indication that the machine learning model is not overfit to the training or validation data, and thus will generalise to fluid flow simulations under different conditions.

Secondly, the accuracy of the machine learning model is much greater than originally expected. For instance, if the mean predictions for a_{yz} are considered, as this is the flow quantity most of interest, it will be on average off by 16.11% for isothermal flow and 19.45% for stratified flow. Considering that the LES took several days to run, the predictions from the machine learning model, which took approximately 6 seconds to obtain, this result represents a good trade-off between speed and accuracy.

A comparison of the machine learning model to predict a_{yz} in this thesis will be made against traditional RANS approaches in Section 5.1.1.

4.4.2 Instantaneous Predictions

Using the machine learning model, the instantaneous predictions (blue) for a_{yz} can be obtained using the isothermal and stratified test data and compared to the LES (orange) values. This is shown in Figures 4.20 and 4.21 respectively.

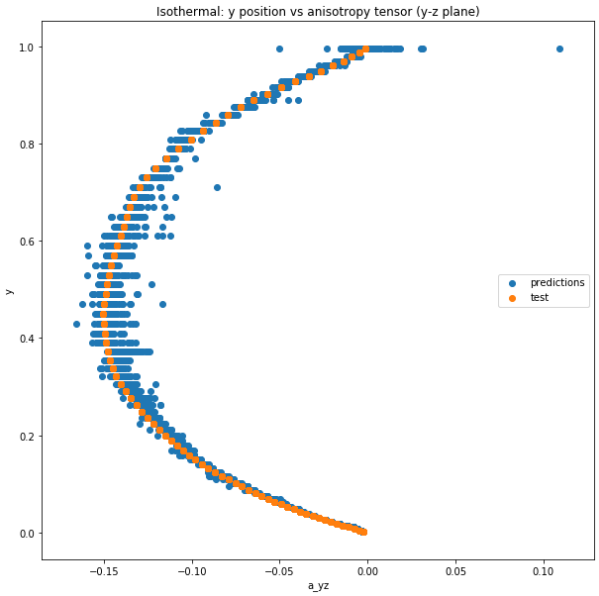


Fig. 4.20 Isothermal: Instantaneous Test Predictions - Model 15

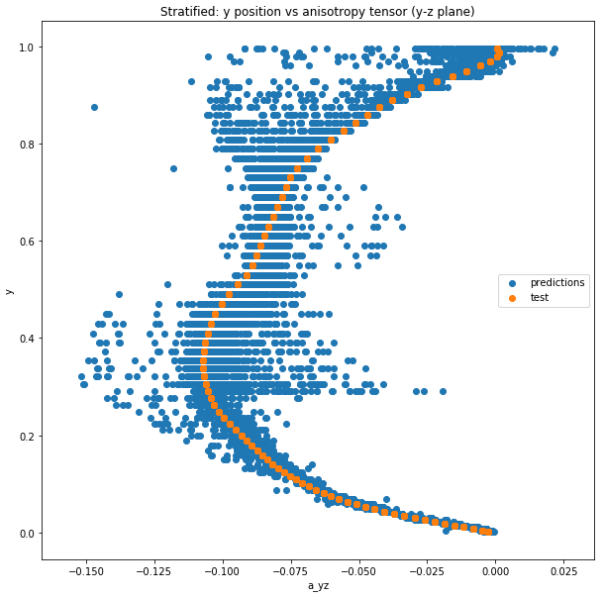


Fig. 4.21 Stratified: Instantaneous Test Predictions - Model 15

As expected, the variance of the predictions is particularly large at the centre of the flow channel ($0.3 < y < 0.8$) and at the top layer ($y = 1.0$) for both the isothermal and stratified plots. This was a similar issue observed throughout the previous section in which the model's hyperparameters were selected.

Unsurprisingly, Figure 4.21 shows the instantaneous predictions for the stratified flow is significantly less accurate than the isothermal flow in Figure 4.20. The greater inaccuracy is due to the more complex nature of the introduction of heat effects to the flow simulation, thus creating density variation within the flow. As a result, the machine learning model finds it more difficult to obtain reliable flow predictions for the stratified case.

Nonetheless, the reason why the blue predictions of a_{yz} don't exactly match the orange LES values, in either case, is due to the LES data on which the machine learning model is trained on. Throughout the Large Eddy Simulation, there is also random fluctuations caused by numerical solver issues and via turbulent eddies within the flow. The random fluctuations disturb the instantaneous predictions, but these disturbances should be removed when an averaging/median process is applied, as described in 4.4.3.

Moreover, there is also a handful of positive instantaneous predictions for a_{yz} which should in fact be negative. Again, positive predictions for a_{yz} is likely caused by disturbances in the input LES data on which the model was trained on. Regardless, the positive predictions should be removed after taking the mean or median of the predictions for a given y-coordinate. Alternatively, one can always 'clip' at zero, filtering the positive predictions, which is done with the dynamic Smagorinsky model for LES.

4.4.3 Mean and Median Predictions

Following the instantaneous predictions for the isothermal and stratified flows in the previous section, the mean and median predictions for any given y-coordinate is calculated. This process is done to remove the noise in the predictions and hence why the mean/median values are the primary summary statistics reported in the literature. Figure 4.22 shows the mean (orange) and median (green) predictions of a_{yz}, in comparison to the LES values (blue) for the isothermal case. This is repeated in Figure 4.23 for the stratified flow.

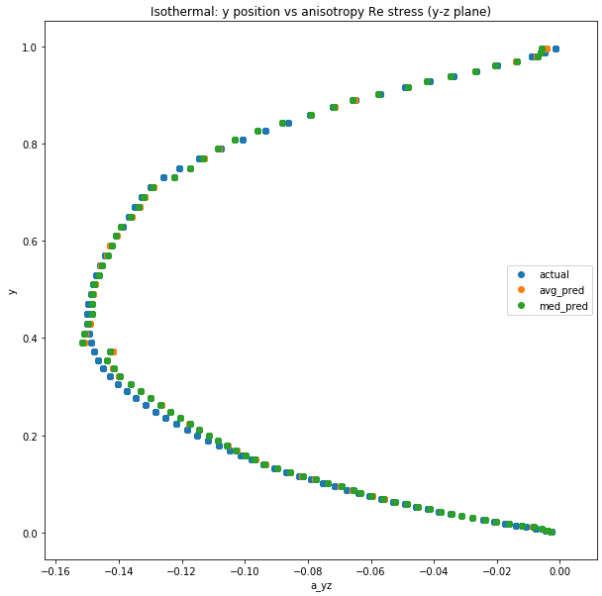


Fig. 4.22 Isothermal: Mean and Median Test Predictions - Model 15

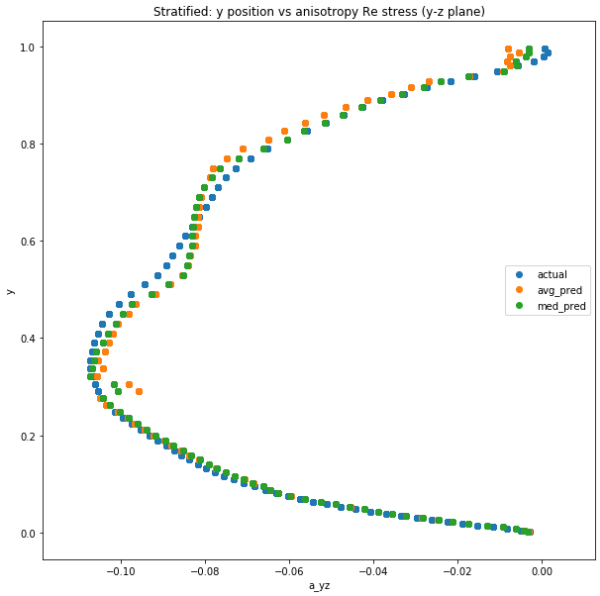


Fig. 4.23 Stratified: Mean and Median Test Predictions - Model 15

In general, the machine learning model does a good job at predicting the anisotropic Reynolds stress on the y-z plane for all levels of the isothermal flow in Figure 4.22. However, a discrepancy is shown at approximately $y = 0.4$ where a discontinuity is present. This discontinuity was also at this y-coordinate through the model training phase and is investigated further in Section 5.3.

Moreover, the stratified flow predictions of the final model on the test data also show good agreement as shown in Figure 4.23, albeit, being less accurate than in the simpler isothermal case. Additionally, there is again issues with predicting the anisotropic Reynolds stress, particularly at $y \approx 0.3$.

Nonetheless, for the remainder of the thesis, only the mean predictions for a_{yz} will be considered. This is because there is a mathematically convenient formula to calculate the standard error which is used in finding the confidence intervals for the next section and also because it is more common in literature to report averaged statistics, for example, from the Reynolds Averaged Navier-Stokes equations.

4.4.4 Confidence Interval on Mean Predictions

If the mean predictions from the previous section are used, then a 95% confidence interval can be constructed for each given y-coordinate. For instance, Figure 4.24 shows orange bands which represent the 95% confidence region for the isothermal flow predictions on the test data. This is interpreted, using statistical language, as the region where we would expect 95% of our

isothermal predictions of a_{yz} to fall in between if we were to repeat the experiment many times on unseen 'real-world' data. The confidence bands are repeated for the stratified flow predictions in Figure 4.25.

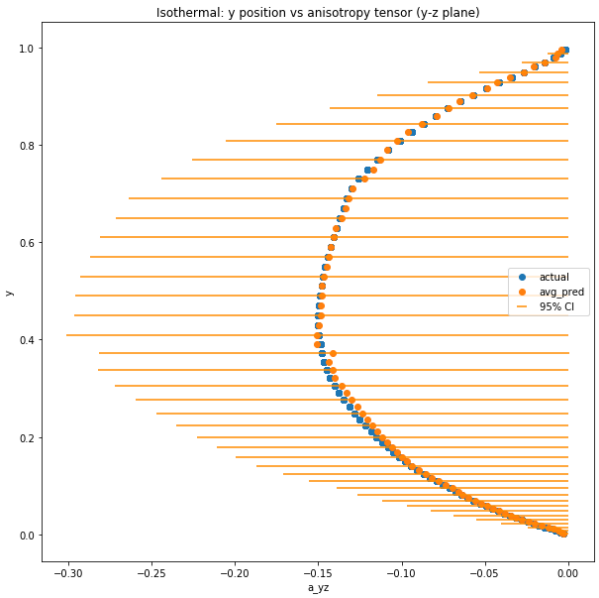


Fig. 4.24 Isothermal: Confidence Interval Test Predictions - Model 15

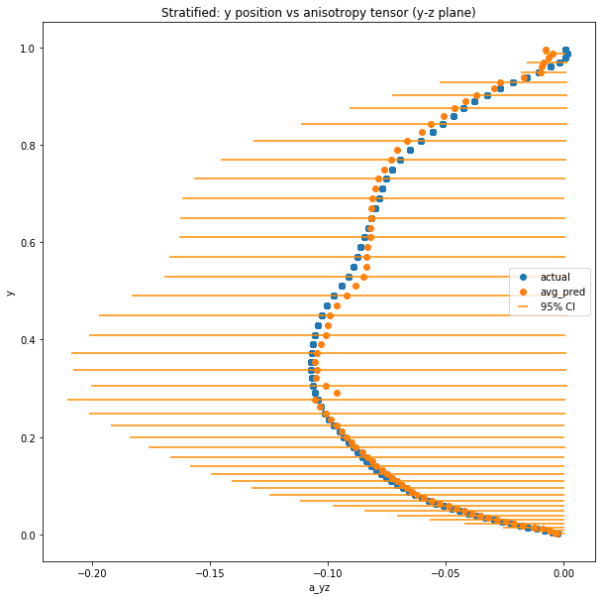


Fig. 4.25 Stratified: Confidence Interval Test Predictions - Model 15

Clearly, in both the isothermal and stratified flow cases, there is more confidence in the predictions at the top and bottom of the isothermal channel as shown by the narrower confidence bands. This was also highlighted in the instantaneous predictions in Section 4.4.2 where the range of predictions was much smaller at the extreme ends of the channel ($y = 0$ and $y = 1$). Furthermore, none of confidence intervals extends into the positive region of a_{yz} in Figures 4.24 and 4.25. This is an indication that the predictions for the machine learning model, after they are averaged, would unlikely be positive, which would otherwise be an error.

As expected, however, the 95% confidence bands are proportionally larger in the stratified case in Figure 4.25. The reason being is that there is more uncertainty when making stratified predictions for a_{yz} due to its more complex flow properties (i.e. heat effects). Moreover, the largest confidence band in Figure 4.25 is at approximately $y = 0.3$ which is the location of the discontinuity. This highlights that the machine learning model struggles to find reliable predictions at this y-coordinate.

4.4.5 Confidence Interval on RMSE

Using a computational method known as bootstrapping, a 95% confidence interval on the Root Mean Squared Error (RMSE) on the test data can be generated. Bootstrapping is a method

whereby the test data is iteratively resampled with replacement and the resulting statistic is used as a method of evaluating how generalisable the machine learning model is. For instance, the 95% confidence intervals in Table 4.19 below can be interpreted by imagining if this experiment was repeated 100 times and the results collected, then we would expect 95 of those experiments to have an outcome somewhere between the intervals in this table.

Table 4.19 95% Confidence Interval of RMSE

	RMSE	**95% CI**
Combined - Instantaneous	0.0067	(0.0064, 0.0069)
Isothermal - Mean	0.0025	(0.0014, 0.0034)
Stratified - Mean	0.0030	(0.0022, 0.0037)

There are two important interpretations which can be drawn from the results in Table 4.19. Firstly, the lower and upper ranges of the confidence interval estimates are proportionally narrow. The narrowness is an indication that the final machine learning model presented is generalisable to a wide range of flows, both isothermal and stratified. Secondly, however, as none of the confidence intervals in Table 4.19 contain the value of 0, it is not possible to conclude that the machine learning model is statistically the same as the Large Eddy Simulation (LES). This is under a null hypothesis of the machine learning model and LES are equivalent (i.e. $H_0 : RMSE = 0$).

For reference, the distributions of the errors, as generated by the bootstrapping process, are shown in Appendix A.6.

4.4.6 Sensitivity Analysis

As previously described, a goal of the analysis was to determine the influence of the input variables on the output variable, a_{yz}. This is achieved with the use of Partial Dependence Plots (PDP) which show the marginal effect of an input variable on the output. The marginal effect is determined by holding all other input variables constant at their mean value and allowing only the one input variable being analysed to vary, checking to see its influence on the output variable. Furthermore, a rug plot has been added to the base of each PDP. A rug plot simply shows the density of the input variable, with areas of high concentration (many straight lines forming a solid block) indicating the most important regions.

For instance, Figure 4.26 shows the PDP plot for the variable which represents the second of ten isotropic basis tensors, $T^{(2)}_{yz}$. The plot shows that for increasing values of the input variable $T^{(2)}_{yz}$, the value of a_{yz} linearly increases on average when all other variables are held constant. This is the expected result because as per the definition of the Explicit Algebraic Stress Model

(EASM) in Equation (3.5) of Section 3.1, the isotropic basis tensors are linearly combined with the scalar coefficients. Thus, Figure 4.26 confirms that the assumptions of the EASM are upheld, meaning that the machine learning model is Galilean invariant - a key requirement for any turbulence model.

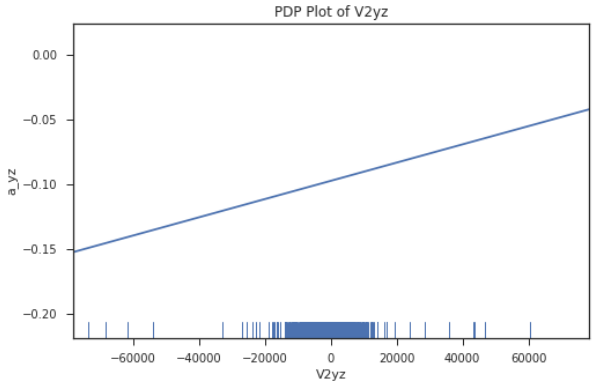


Fig. 4.26 PDP of $T_{yz}^{(2)}$

On the other hand, I_1 in Figure 4.27, appears have no marginal effect on a_{yz}. While there are regions where the relationship is decreasing, the rug plot at the bottom of the figure indicates that this variable does not take on any of these values where this decreasing relationship holds. Instead, the flat line highlights there is no marginal effect. As a result, the machine learning model does not use this variable independently, instead it relies on complex interactions between I_1 and the other input variables, to find the underlying patterns in the LES data.

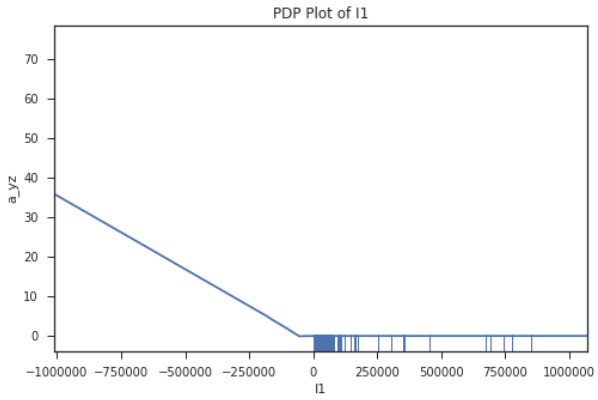


Fig. 4.27 PDP of I_1

Finally, the PDP plot of the phi_grad variable in Figure 4.28, has an interesting interpretation. In particular, the plot shows a sharp drop in the marginal effect of phi_grad on a_{yz} at approximately 0. This is likely caused by the isothermal data which has a value of phi_grad = 0 for all points in the open channel as it has no heat effects. As a result, it appears as if this variable is

particularly important to the machine learning model when distinguishing between the isothermal and stratified flow predictions for a_{yz}. Hence, for all future studies on stratified flow prediction using machine learning, this variable should be included.

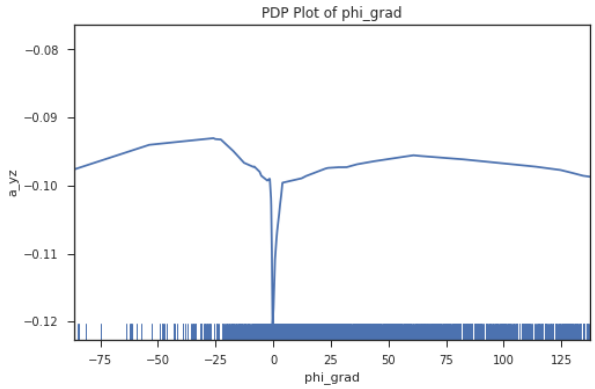


Fig. 4.28 PDP of phi_grad

The remainder of the PDP plots, for the other input variables, are shown in Appendix A.5.

4.4.7 Summary of Results

Overall, the model selection process found that Model 15 yielded the lowest Root Mean Squared Error on the combined (isothermal + stratified) validation data. For this reason, this model was further evaluated on the test data. Here, it was shown that this machine learning model, after taking the mean predictions of a_{yz} for each y-coordinate, had good predictive accuracy for both the isothermal and stratified flow cases. However, further investigation is required to explain the discontinuities at the centre of the flow channel. Furthermore, it was found that the selected machine learning model for the anisotropic Reynolds stress was not overfit to the training data and thus is generalisable. The model is also generalisable in the sense that the PDP plots show that the assumptions of the Explicit Algebraic Stress Model are upheld. Therefore, Galilean invariance is enforced. Finally, when modelling stratified open channels with data-driven techniques, the inclusion of the phi_grad variable, measuring the heat effects, is recommended for all future studies.

Chapter 5

Discussion

5.1 Comparison to Other Turbulence Models

5.1.1 Traditional RANS Approaches

The stated objective of this thesis was to develop a data-driven closure model for the RANS equations that would be more accurate than traditional closure approaches. This is to overcome the deficiencies of tradition eddy-viscosity RANS models, discussed in the literature review, which makes them unsuitable for the flows tested in this thesis.

While traditional RANS models were not directly tested in this thesis, as outlined in the limitations of the methodology, the work conducted by Ling et al. [34] can be considered as a proxy investigation. For instance, their paper also used a Tensor Basis Neural Network (TBNN), using a smaller set of input variables, to model the anisotropic Reynolds stress of isothermal flow through a square duct. They found that the Mean Absolute Percentage Error (MAPE) of their machine learnt turbulence model was 44% and 30% lower than that of the Linear Eddy Viscosity Model (LEVM) and Quadratic Eddy Viscosity Model (QEVM) respectively. In fact, Table 5.1 compares the ratio between the RMSE and the (absolute) maximum value of the anisotropic Reynolds stress, for both these studies, where a lower ratio indicates a more accurate model.

Table 5.1 Result Comparison to Ling et al. [35]

	Isothermal Duct (Ling)	**Isothermal Channel (Thesis)**
RMSE	0.0800	0.0025
Max. $a_{\alpha\beta}$ (Absolute)	0.6000	0.1450
Ratio	0.1333	0.0172

Since this study has a lower ratio, this is evidence suggesting that the average prediction error for the machine learning model in this work is more accurate than theirs. Hence, the machine learning model for the anisotropic Reynolds stress in this thesis should be more accurate than LEVM and QEVM by at least the same amount.

Furthermore, Ling et al. [34] tested a more complex flow case which was the flow over a wavy wall using the same machine learning model. Again, they found the machine learning turbulence model had improvement of the MAPE of 56% and 27% of LEVM and QEVM respectively. Hence, if the assumption is made that the TBNN structure in this thesis is equivalent to the one proposed in Ling et al. [34], it is safe to assume that the turbulence model trained using machine learning provides an improvement over traditional RANS approaches.

In terms of modelling the anisotropic Reynolds stress for stratified open-channel flow, the problems associated with traditional RANS closures, such as $k-\varepsilon$, are well understood and well documented [9, 39, 64, 79]. For example, Craft et al. [9] concluded that RANS models using eddy viscosity assumptions are insufficient to model buoyancy damping and turbulent mixing as a result of the stratified heat effects. This was supported by the work of McGuirk and Papadimitriou [39], Uitenbogaard [64] and Zeman and Lumley [79] who each independently summarised the $k-\varepsilon$ turbulence model as being 'inadequate' for stratified flows, such as the one tested in this thesis.

This thesis proposes that it is indeed possible to capture stratification effects within the flow when a data-driven RANS closure is developed. While the root mean squared errors are not as low as in the isothermal channel case, this thesis acts as a proof of concept for more advanced machine learning turbulence models for buoyancy-driven flows.

5.1.2 Data-Driven Approaches

It is difficult to draw direct comparisons regarding accuracy when comparing the turbulence model in this work to other data-driven models. This is because other studies each tested various flow geometries under different conditions, each with alternative problem definitions. However, the ratios shown in Table 5.1 in the previous section suggest that the machine learning turbulence model in this study is comparable, if not an improvement, to the work conducted by Ling et al. [34].

There are two potential reasons for improvement, the first of which is that the isothermal open channel flow in this thesis is a simpler case than isothermal duct flow in their research. While this may be the case, the large discrepancy between the two ratios in Table 5.1 should

account for this. Thus, it is more likely that the extension of the methodology in this thesis to include the additional variables of a wall-based Reynolds number and the gradient of the heat effects, which were not included in Ling et al. [35], improved the accuracy of the turbulence model presented here.

Moreover, the selection of input variables and the machine learning algorithm is critical to the overall accuracy. For instance, Weatheritt and Sandberg [72] limited themselves to only the first three of ten isotropic basis tensors (i.e. $T^{(1)}_{\alpha\beta}, T^{(2)}_{\alpha\beta}, T^{(3)}_{\alpha\beta}$) and the first two invariants (i.e. I_1, I_2) which were modelled using a genetic algorithm. The resulting equation for the anisotropic Reynolds stress had only 9 terms in it, which is vastly simpler to the model in this thesis which had 6,012 terms. While the number of terms in the Reynolds stress closure model is by no means a measure of success, it does put into context why the results of Weatheritt and Sandberg [72] struggled to make accurate flow predictions for the anisotropic Reynolds stress in all situations tested in their paper.

The level of complexity of the resulting model in this thesis was comparable of that to Wu et al. [76], but with the addition of the wall-based Reynolds number and the gradient of the heat effects. Nonetheless, the work conducted by Wang et al. [67] extended Wu et al. [76] by including 47 additional variables, which were a further decomposition of the five invariants and ten isotropic basis tensors. This was based on the justification that the variables in Wu et al. [76] were "not necessarily rich enough to represent all possible polynomial invariants of the local mean flow variables" [67]. However, it is concluded that this is indeed not the case, and a smart selection of other flow variables (such as the wall based Reynolds number), rather than adding as many variables as possible, can yield a comparable performance increase.

Finally, it was important to ensure that Galilean invariance had been enforced. Studies conducted by Ling et al. [33] demonstrate that the use of the explicit algebraic stress model with input variables of the invariants and isotropic basis tensors make it Galilean invariant. Moreover, studies conducted by Wu et al. [76] demonstrate that adding other variables such as a wall-based Reynolds number did not prevent this condition from holding. Therefore, based on the methodology and the results of the sensitivity analysis of this thesis, it is reasonable to assume that Galilean invariance would indeed hold.

5.2 Turbulent Boundary Layers

The results obtained indicate that the inaccuracies of modelling the anisotropic Reynolds stress on the y-z plane (a_{yz}) are related to the turbulent boundary layers. The boundary layer is the region

of a flow over which the velocity is slowed, relative to the initial and free steam velocities, by frictional forces. For flow in an open channel, the turbulent boundary layers can be approximated using the results from the flow over a flat plate. The turbulent velocity profile can be broadly classified into four layers, starting from the region closest to the channel wall: viscous sub-layer, buffer zone and log-law region, and the outer layer.

For the isothermal open-channel case, the regions of turbulent flow can be imposed on the mean predictions for a_{yz}, as per Figure 5.1. Likewise, these regions can be plotted on the mean predictions for the anisotropic Reynolds stress for the stratified flow case using the test data, as per Figure 5.2. The calculation of these turbulent boundary layers, for both the isothermal and stratified cases, are detailed in Appendix C.

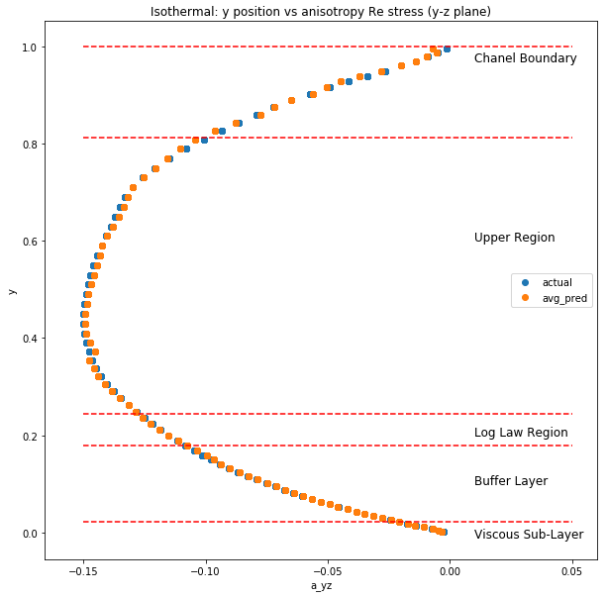


Fig. 5.1 Isothermal Boundary Layers

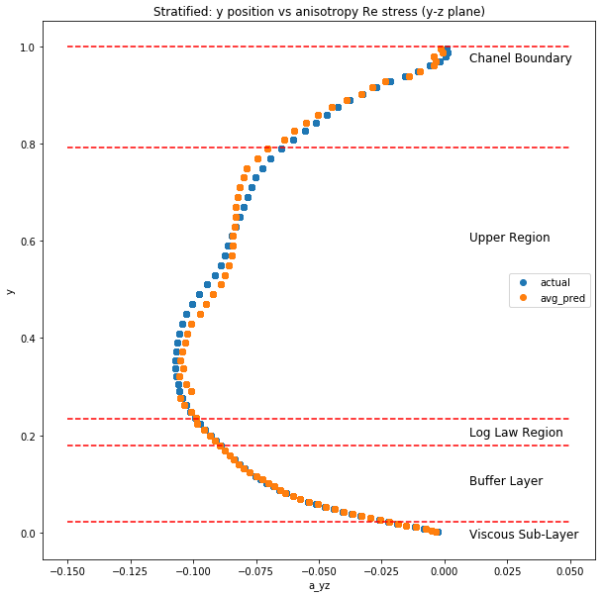


Fig. 5.2 Stratified Boundary Layers

As shown in Figure 5.1 above, the anisotropic Reynolds stress has generally been accurately modelled for all turbulent layers for the isothermal case. However, the largest discrepancy for the isothermal case is at the channel boundary. This is at the free surface at which there is theoretically an interface between the water and the atmosphere. According to Deusebio et al. [12], who conducted direct numerical simulation (DNS) on an open channel flow, gravity waves are likely to "develop and dominate closer to the upper boundary". Since there were no input variables measuring the effects of gravitational waves, the machine learning model may have difficulty accurately modelling the phenomena.

Moreover, there may be modelling inaccuracies at the free surface due to the boundary conditions imposed by the LES. The difficulty of accurately modelling the free surface of any open-channel,

both isothermal and stratified, is well documented in literature [17, 41, 59]. Therefore, numerical issues when the CFD solver tries to impose the boundary condition may also be contributing to the discrepancies at this location.

Regarding the stratified open channel case, Figure 5.2 shows that the machine learning model for a_{yz} decreases in accuracy in the upper region. This may be explained by the work of Taylor et al. [60], who found that turbulence in stratified channels can be classified into three regions: unstratified, buoyancy-affected and buoyancy dominated. In the Large Eddy Simulation used to generate the data, a constant heat flux was applied at the free surface. As a result, the heat effects do not reach the inner turbulent regions of the channel (i.e. viscous sub-layer, buffer and log law), and hence can be reasonably modelled as isothermal. This is why Figures 5.1 and 5.2 are both accurately modelled for the lower regions of the flow.

On the other hand, the upper region of the flow in Figure 5.2 is classified as buoyancy-affected [60]. Komori et al. [27] found that in this region of the stratified channel, there is a complex interaction between the turbulent eddies and the stratification effects. Here, "fluctuating motions become close to wavelike mentions" due to the "intermittent buoyancy-driven motions". The seemingly random fluid motions are extremely difficult to model [13], hence why the machine learning model becomes more inaccurate in this region.

Finally, the utmost turbulent layer, close to the channel boundary (free surface), is classified as buoyancy dominated [60]. In this region of the flow, the fluctuations caused by stratification and the turbulent eddy structures are dampened, creating a more simple flow [13]. This is why the machine learning model in this region becomes more accurate.

5.3 Discontinuity in Predictions

Throughout the model selection process and the final evaluation of the model, reference was made to the clear discontinuities of the pattern of mean a_{yz} along the y-coordinate. In particular, the isothermal flow case generally had a discontinuity at $y \approx 0.4$ (see Figure 5.1 above for example) and the stratified flow case at $y \approx 0.3$ (see Figure 5.2 above). This systematically repeated throughout this thesis, for both the validation and test data sets, indicating these inaccuracies have a root cause.

Upon further analysis, it is clear that the discontinuities are related to the input data on which the machine learnt turbulence model was trained on. In general, the discontinuity occurs at the y-coordinate at which the anisotropic Reynolds stress on the y-z plane is at its minimum

value. When analysing the exploratory data analysis it reveals the clear presence of outliers in this location. For instance, observing Figures A.33 through A.49 in Appendix A.3 shows an hourglass pattern. The input variables have outliers at both their minimum and maximum values in each of the isothermal and stratified cases. While the numerical issues relating to modelling the free surface were discussed in the previous section, hence explaining the outliers when $a_{yz} \approx 0$ (maximum value), the outliers at the minimum value indicate an issue with the data generation process using the CFD software PUFFIN.

When examining the mesh of the Large Eddy Simulations (LES), which dictates the control volumes over which the equations of fluid flow are solved, for both the isothermal and stratified cases, it became clear this was the cause of the discontinuities of the mean a_{yz} plot. For instance, the grid size of the mesh at $y \approx 0.4$ had a clear loss of resolution (i.e. the grid size increases at this point) for the isothermal case. Here, the grid size changed from $\Delta y = 0.018$ to $\Delta y = 0.02$. Likewise, the mesh used to calculate the LES for the stratified flow case also increased its grid size at its discontinuity location of $y \approx 0.3$ from $\Delta y = 0.015$ to $\Delta y = 0.018$. Therefore, it is proposed that the sudden loss of resolution affects the calculation of the input and output variables, particularly those involving gradients which used first-order approximations. As a result, the machine learning algorithm is clearly sensitive to the changes of the grid, hence why discontinuities are systematically present at these y-locations.

Chapter 6

Conclusion

6.1 Research Outcomes

The primary objective of this thesis was to develop a proof-of-concept Reynolds Averaged Navier-Stokes (RANS) turbulence model closure that would be applicable to both isothermal and stratified open channel flows. This model closure was developed using data-driven techniques, such as machine learning, and provided an improvement over traditional RANS turbulence models, such as $k-\varepsilon$ and $k-\omega$. The developed model is based on an Explicit Algebraic Stress (EASM) formulation of the Reynolds Averaged Navier-Stokes equations, and finds a functional representation of the anisotropic Reynolds stress on the y-z plane, a_{yz}.

The proposed methodology and obtained results have demonstrated that it is indeed possible to use machine learning to develop a turbulence model, based on the data from Large Eddy Simulations (LES). Specifically, an adaptation of the neural network algorithm was proposed, known as a Tensor Basis Neural Network (TBNN). The selection of the hyperparameters in the TBNN was informed by a model selection process. Here, the TBNN was fit on the training data, a combination of instantaneous LES results for both the isothermal and stratified open-channel cases. After the model was trained, the quality of its predictions for a_{yz} were measured on the combined validation data. The model which had the lowest validation root mean squared error (RMSE) was selected for further evaluation on three criteria of success: accuracy, generalisability and interpretability.

The results obtained on the test dataset, used as a representation of 'real-world' unseen data, demonstrated good accuracy for both the isothermal and stratified flows. After the average value of a_{yz} was taken for a given y-coordinate, similar to that of a traditional RANS model, the RMSE was 0.0025 and the Mean Absolute Percentage Error (MAPE) was 16.11%. The more complex stratified flow case had a larger RMSE of 0.0030 and a MAPE of 19.45%. Based on further

analysis, it was found that inaccuracies may be due to gravitational waves and numerical issues at the free surface, loss of resolution in the LES mesh, and issues modelling the complex interaction between turbulent eddies and buoyancy effects. Nonetheless, the work conducted by Ling et al. [35] suggests that the TBNN proposed in this work will still be markedly more accuracy than traditional RANS models.

Furthermore, the data-driven turbulence closure presented in this thesis was shown to have good generalisability. This was achieved using statistical techniques, such as bootstrapping, to generate confidence intervals for the predictions and RMSE. While the individual predictions may have wide confidence intervals, suggesting uncertainty in the instantaneous predictions, after taking the averaged values a_{yz} for a given y-coordinate, the 95% confidence intervals for the RMSE of the isothermal and stratified cases are (0.0014, 0.0034) and (0.0022, 0.0037) respectively.

Finally, Partial Dependence Plots (PDP) were used to qualitatively describe the importance of the input variables when making flow predictions. It was found that the linear formulation EASM was upheld, thus confirming that Galilean invariance was enforced. Moreover, the PDP analysis also demonstrated the relative impact that the wall-based Reynolds number and gradient of the heat variable had on the predictive accuracy of both the isothermal and stratified channel flows.

6.2 Future Work

Based on the findings in this thesis, several recommendations are put forward to better meet the accuracy, generalisability and interpretability objectives. While this thesis has shown that data-driven methods, such as machine learning, to develop turbulence closures is indeed possible, further work could be conducted to displace traditional RANS closures as the favoured methods for turbulence modellers.

6.2.1 Additional Flow Input Variables

This thesis made use of 17 input variables: the 10 isotropic basis tensors and 5 invariants as proposed by Pope [49]; a wall-based Reynolds number; and a variable, known as phi_grad, which measured the heat effects of stratification. While the results of this study have been shown to accurately capture the inherent relationships within the LES data, the flexibility of the structure of the TBNN would allow for further data to increase the predictive accuracy.

Based on the suggestions made in the discussion section, it was theorised that a source of

inaccuracy at the free surface layer was gravity waves. To overcome the inaccuracies at this boundary, Ponce and Simons [48] and Nakayama and Yokojima [41] proposed the addition of the Froude number and a dimensionless wave number to more accurately model these effects. Additionally, to more accurately model the buoyancy-affected regions of the flow [60], the study conducted by Komori et al. [27] concluded that the Richardson number was a 'significant parameter for representing the buoyancy effects'. Finally, the further decomposition of the 10 basis tensors and 5 invariants of Pope [49] could be conducted. This was done by Wang et al. [65], which resulted in 47 additional variables and a marginal improvement in their data-driven model accuracy.

6.2.2 Reynolds Stress Modelling

The TBNN structure of the turbulence model proposed in this thesis is based on the EASM. For a relatively simple flow, such as the open channel flow in this study, it can be broken down into a one-dimensional problem where there is only one output quantity (i.e. a_{yz}). By the definition proposed in the methodology section of this thesis, the EASM essentially becomes a generalised linear model as previously described. However, it is well understood that turbulence modelling is a complex, non-linear problem and the definition proposed by the EASM may be overly simplified.

An alternative to this approach is the use of Reynolds Stress Modelling (RSM). As described in the literature review, the RSM, which is another type of RANS closure, uses six non-linear equations to model the Reynolds stress. While this would be considerably more difficult to model using machine learning, it is hypothesised that this type of closure would be more accurate than the simplistic EASM.

6.2.3 Posterior Study

The goal of this thesis was to develop a data-driven closure for the RANS equations using machine learning. This explicitly modelled the anisotropic Reynolds stress on the y-z plane (a_{yz}). However, like most CFD studies, it is also desirable to predict further quantities of interest, such as flow velocity, vorticity and pressure.

This could be achieved by inserting the developed turbulence closure into a RANS solver, such as those offered by open-source CFD software *openFOAM*. This posterior study has been conducted by Weatheritt and Sandberg [72], Ling et al. [35] and Wang et al. [65] and has primarily been used to predict the velocity field. As such, direct comparisons to traditional RANS models, such as the $k-\varepsilon$ model, would be made possible.

References

[1] Ahrens, J., Geveci, B., and Law, C. (2005). ParaView: An End-User Tool for Large Data Visualization.

[2] Armenio, V. and Sarkar, S. (2002). An investigation of stably stratified turbulent channel flow using large-eddy simulation. *Journal of Fluid Mechanics*, 459:1–42.

[3] Banfield, R. E., Hall, L. O., Bowyer, K. W., Bhadoria, D., Kegelmeyer, W. P., and Eschrich, S. (2004). A Comparison of Ensemble Creation Techniques. In *Multiple Classifier Systems*, Lecture Notes in Computer Science, pages 223–232. Springer, Berlin, Heidelberg.

[4] Bergstra, J. and Bengio, Y. (2012). Random Search for Hyper-parameter Optimization. *J. Mach. Learn. Res.*, 13:281–305.

[5] Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., and Shah, H. (2016). Wide & Deep Learning for Recommender Systems. *arXiv:1606.07792 [cs, stat]*. arXiv: 1606.07792.

[6] Chollet, F. and others (2015). Keras.

[7] Claesen, M. and De Moor, B. (2015). Hyperparameter Search in Machine Learning. *arXiv:1502.02127 [cs, stat]*. arXiv: 1502.02127.

[8] Craft, T. J., Ince, N. Z., and Launder, B. E. (1996a). Recent developments in second-moment closure for buoyancy-affected flows. *Dynamics of Atmospheres and Oceans*, 23(1):99–114.

[9] Craft, T. J., Launder, B. E., and Suga, K. (1996b). Development and application of a cubic eddy-viscosity model of turbulence. *International Journal of Heat and Fluid Flow*, 17(2):108–115.

[10] Dennis Jr., J. E. (1973). SOME COMPUTATIONAL TECHNIQUES FOR THE NONLINEAR LEAST SQUARES PROBLEM. In Byrne, G. D. and Hall, C. A., editors, *Numerical Solution of Systems of Nonlinear Algebraic Equations*, pages 157–183. Academic Press.

[11] Despotovic, D. T. a. V. (2012). Artificial Intelligence Techniques for Modelling of Temperature in the Metal Cutting Process. *Metallurgy - Advances in Materials and Processes*.

[12] Deusebio, E., Schlatter, P., Brethouwer, G., and Lindborg, E. (2011). Direct numerical simulations of stratified open channel flows. *Journal of Physics: Conference Series*, 318(2):022009.

[13] Dong, Y. H. and Lu, X. Y. (2005). Direct numerical simulation of stably and unstably stratified turbulent open channel flows. *Acta Mechanica*, 177(1):115–136.

[14] Durbin, P. A. (2018). Some Recent Developments in Turbulence Closure Modeling. *Annual Review of Fluid Mechanics*, 50(1):77–103.

[15] Edeling, W. N., Cinnella, P., Dwight, R. P., and Bijl, H. (2014). Bayesian estimates of parameter variability in the k–epsilon turbulence model. *Journal of Computational Physics*, 258:73–94.

[16] Edeling, W. N., Iaccarino, G., and Cinnella, P. (2017). A return to eddy viscosity model for epistemic UQ in RANS closures. *arXiv:1705.05354 [physics]*. arXiv: 1705.05354.

[17] Faure, J.-B., Buil, N., and Gay, B. (2004). 3-D Modeling of unsteady free-surface flow in open channel. *Journal of Hydraulic Research*, 42(3):263–272.

[18] Germano, M., Piomelli, U., Moin, P., and Cabot, W. H. (1991). A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765.

[19] Gray, D. D. and Giorgini, A. (1976). The validity of the boussinesq approximation for liquids and gases. *International Journal of Heat and Mass Transfer*, 19(5):545–551.

[20] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, NETHERLANDS.

[21] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

[22] Jain, A. (2016). Fundamentals of Deep Learning - Starting with Artificial Neural Network.

[23] Jakirlić, S. and Maduta, R. (2015). Extending the bounds of 'steady' RANS closures: Toward an instability-sensitive Reynolds stress model. *International Journal of Heat and Fluid Flow*, 51:175–194.

[24] Jamil, M. and Yang, X.-S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194.

[25] Kirkpatrick, M. P. (2013). The PUFFIN Manual: An Engineering and Environmental Fluid Dynamics Simulation Model.

[26] Kolmogorov, A. N. (1991). The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds Numbers. *Proceedings: Mathematical and Physical Sciences*, 434(1890):9–13.

[27] Komori, S., Ueda, H., Ogino, F., and Mizushina, T. (1983). Turbulence structure in stably stratified open-channel flow. *Journal of Fluid Mechanics*, 130:13–26.

[28] Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4(2):87–112.

[29] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

[30] Kuo, A. D. (1998). A Least-Squares Estimation Approach to Improving the Precision of Inverse Dynamics Computations. *Journal of Biomechanical Engineering*, 120(1):148–159.

[31] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[32] Leschziner, M. and Lien, F.-S. (2002). Numerical Aspects of Applying Second-Moment Closure to Complex Flows. In Launder, B. E. and Sandham, N. D., editors, *Closure Strategies for Turbulent and Transitional Flows*, pages 153–187. Cambridge University Press, Cambridge.

[33] Ling, J., Jones, R., and Templeton, J. (2016a). Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*, 318:22–35.

[34] Ling, J., Kurzawski, A., and Templeton, J. (2016b). Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics; Cambridge*, 807:155–166.

[35] Ling, J., Ruiz, A., Lacaze, G., and Oefelein, J. (2016c). Uncertainty Analysis and Data-Driven Model Advances for a Jet-in-Crossflow. *Journal of Turbomachinery*, 139(2):021008–021008–9.

[36] Ling, J. and Templeton, J. (2015). Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty. *Physics of Fluids*, 27(8):085103.

[37] Louppe, G. (2014). Understanding Random Forests: From Theory to Practice. *arXiv:1407.7502 [stat]*. arXiv: 1407.7502.

[38] Mansour, N. N., Kim, J., and Moin, P. (1988). Reynolds-stress and dissipation-rate budgets in a turbulent channel flow. *Journal of Fluid Mechanics*, 194:15–44.

[39] McGuirk, J. and Papadimitriou, C. (1985). Buoyant surface layers under fully entraining and internal hydraulic jump conditions. In *5th Symp. on Turbulent Shear Flows, Cornell University*, pages 22 – 41, Cornell University.

[40] Michalski, R. S. (1986). Understanding The Nature Of Learning: Issues And Research Directions. In *Machine Learning: An Artificial Intelligence Approach*, pages 3–25. Morgan Kaufmann.

[41] Nakayama, A. and Yokojima, S. (2003). Modeling Free-Surface Fluctuation Effects for Calculation of Turbulent Open-Channel Flows. *Environmental Fluid Mechanics*, 3(1):1–21.

[42] Osborne, M. R., Presnell, B., and Turlach, B. A. (2000). A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20(3):389–403.

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

[44] Parneix, S., Laurence, D., and Durbin, P. A. (1998). A Procedure for Using DNS Databases. *Journal of Fluids Engineering*, 120(1):40–47.

[45] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[46] Piatetsky-Shapiro, G. (1991). Knowledge Discovery in Real Databases: A Report on the IJCAI-89 Workshop. *AI Mag.*, 11(5):68–70.

[47] Pichler, R., Sandberg, R. D., Michelassi, V., and Bhaskaran, R. (2016). Investigation of the Accuracy of RANS Models to Predict the Flow Through a Low-Pressure Turbine. *Journal of Turbomachinery*, 138(12):121009–121009–12.

[48] Ponce, V. M. and Simons, D. B. (1977). Shallow Wave Propagation in Open Channel Flow. *Journal of the Hydraulics Division*, 103(12):1461–1476.

[49] Pope, S. B. (2000). *Turbulent Flows*. Cambridge University Press, Cambridge.

[50] Raiesi, H., Piomelli, U., and Pollard, A. (2011). Evaluation of Turbulence Models Using Direct Numerical and Large-Eddy Simulation Data. *Journal of Fluids Engineering*, 133(2):021203–021203–10.

[51] Rocha, M. and Neves, J. (1999). *Preventing Premature Convergence to Local Optima in Genetic Algorithms via Random Offspring Generation*.

[52] Sandberg, R. D. and Coleman, G. N. (2010). A Primer on Direct Numerical Simulation of Turbulence – Methods, Procedures and Guidelines. Technical report, University of Southampton, Aerodynamics & Flight Mechanics Research Group.

[53] Schaer, R., Müller, H., Depeursinge, A., Schaer, R., Müller, H., and Depeursinge, A. (2016). Optimized Distributed Hyperparameter Search and Simulation for Lung Texture Classification in CT Using Hadoop. *Journal of Imaging*, 2(2):19.

[54] Slotnick, J. (2014). CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences. Technical report.

[55] Smagorinsky, J. (1963). General circulation experiments with the primitive equations. *Monthly Weather Review*, 91(3):99–164.

[56] Sola, J. and Sevilla, J. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems - IEEE Journals & Magazine. *IEEE Transactions on Nuclear Science*, 44(3):1464 – 1468.

[57] Spalart, P., Jou, W.-H., Strelets, M., and Allmaras, S. (1997). Comments on the Feasibility of LES for Wings, and on a Hybrid RANS/LES Approach.

[58] Spalart, P. R., Shur, M. L., Strelets, M. K., and Travin, A. K. (2015). Direct Simulation and RANS Modelling of a Vortex Generator Flow. *Flow, Turbulence and Combustion*, 95(2-3):335–350.

[59] T. Swean, J., Leighton, R., Handler, R., and Swearingen, J. (1991). Turbulence modeling near the free surface in an open channel flow. In *29th Aerospace Sciences Meeting*, Reno,NV,U.S.A. American Institute of Aeronautics and Astronautics.

[60] Taylor, J. R., Sarkar, S., and Armenio, V. (2005). Large eddy simulation of stably stratified open channel flow. *Physics of Fluids*, 17(11):116602.

[61] Tracey, B., Duraisamy, K., and Alonso, J. (2013). Application of Supervised Learning to Quantify Uncertainties in Turbulence and Combustion Modeling. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics.

[62] Trippi, R. R. and Turban, E., editors (1992). *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance*. McGraw-Hill, Inc., New York, NY, USA.

[63] Turner, L. and Erskine, W. D. (2005). Variability in the development, persistence and breakdown of thermal, oxygen and salt stratification on regulated rivers of southeastern Australia. *River Research and Applications*, 21(2-3):151–168.

[64] Uitenbogaard, R. (1988). Measurement of turbulent fluxes in a steady, stratified mixing layer. In *3rd Int. Symp. on Refined Flow Modelling and Turbulence Measurement*, Tokyo.

[65] Wang, J.-X., Wu, J., Ling, J., Iaccarino, G., and Xiao, H. (2017a). A Comprehensive Physics-Informed Machine Learning Framework for Predictive Turbulence Modeling. *arXiv:1701.07102 [physics]*. arXiv: 1701.07102.

[66] Wang, J.-X., Wu, J.-L., and Xiao, H. (2017b). A Physics Informed Machine Learning Approach for Reconstructing Reynolds Stress Modeling Discrepancies Based on DNS Data. *Physical Review Fluids*, 2(3). arXiv: 1606.07987.

[67] Wang, J.-X., Wu, J.-L., and Xiao, H. (2017c). Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Physical Review Fluids*, 2(3):034603.

[68] Wang, L. and Lu, X.-Y. (2005). Large eddy simulation of stably stratified turbulent open channel flows with low- to high-Prandtl number. *International Journal of Heat and Mass Transfer*, 48(10):1883–1897.

[69] Weatheritt, J. (2015). The Development of Data Driven Approaches to Further Turbulence Closures.

[70] Weatheritt, J., Pichler, R., Sandberg, R. D., Laskowski, G., and Michelassi, V. (2017). Machine Learning for Turbulence Model Development Using a High-Fidelity HPT Cascade Simulation. page V02BT41A015.

[71] Weatheritt, J. and Sandberg, R. D. (2016). A novel evolutionary algorithm applied to algebraic modifications of the RANS stress-strain relationship. *Journal of Computational Physics*, 325:22–37.

[72] Weatheritt, J. and Sandberg, R. D. (2017a). The development of algebraic stress models using a novel evolutionary algorithm. *International Journal of Heat and Fluid Flow*, 68:298–318.

[73] Weatheritt, J. and Sandberg, R. D. (2017b). Hybrid Reynolds-Averaged/Large-Eddy Simulation Methodology from Symbolic Regression: Formulation and Application. *AIAA Journal*, 55(11):3734–3746.

[74] Weinmann, M. and Sandberg, R. (2009). Suitability of Explicit Algebraic Stress Models for Predicting Complex Three-Dimensional Flows. In *19th AIAA Computational Fluid Dynamics*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics.

[75] Williamson, N., Armfield, S. W., Kirkpatrick, M. P., and Norris, S. E. (2015). Transition to stably stratified states in open channel flow with radiative surface heating. *Journal of Fluid Mechanics*, 766:528–555.

[76] Wu, J., Sun, R., Laizet, S., and Xiao, H. (2017). Representation of Reynolds Stress Perturbations with Application in Machine-Learning-Assisted Turbulence Modeling.

[77] Xiao, H., Wu, J. L., Wang, J. X., Sun, R., and Roy, C. J. (2016). Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier–Stokes simulations: A data-driven, physics-informed Bayesian approach. *Journal of Computational Physics*, 324:115–136.

[78] Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv:1505.00853 [cs, stat]*. arXiv: 1505.00853.

[79] Zeman, O. and Lumley, J. L. (1979). Buoyancy effects in entraining turbulent boundary layers: a second order closure study.

[80] Zhang, Z. J. and Duraisamy, K. (2015). Machine Learning Methods for Data-Driven Turbulence Modeling. In *22nd AIAA Computational Fluid Dynamics Conference*, AIAA AVIATION Forum.

Appendix A

Further Results and Analysis

This Appendix contains the additional visualisations and plots introduced the results and analysis section of this study. Firstly, Appendix A.1 contains the flow visualisations for the isothermal and stratified open channel flow cases in terms of the the w-velocity, vorticity and the temperature fluctuation (ϕ). Appendix A.2 shows further Kernel Density Estimate (KDE) plots for the input variables. These plots show the distribution of the input variables and can be used to diagnose issues with outliers. Moreover, Appendix A.3 displays scatter plots, with a_{yz} plotted against the input variables. A similar series of plots is shown in Appendix A.4, with the y-coordinate of the open channel instead being plotted on the y-axis. Appendix A.5 displays the remainder of the Partial Dependence Plots (PDP) which were used to determine the most important input variables. Finally, Appendix A.6 shows the distribution of the bootstrapped statistics which were used to find the 95% confidence intervals.

A.1 Flow Visualisations

A.1.1 Isothermal

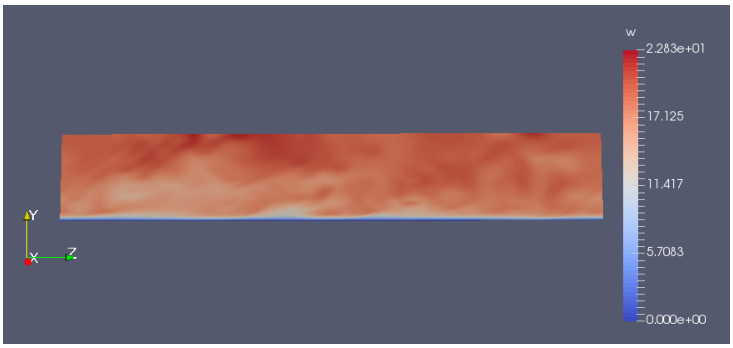


Fig. A.1 Isothermal: w-velocity at t = 30

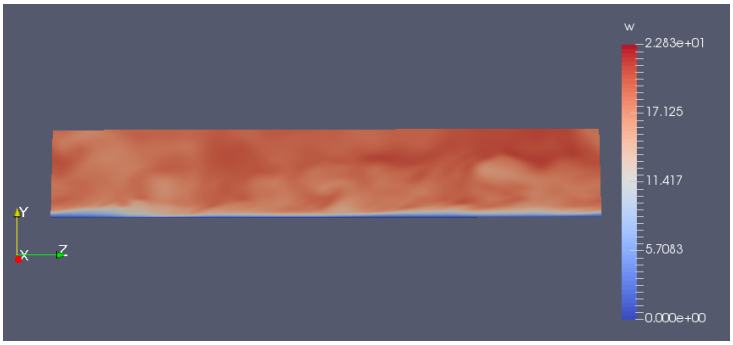


Fig. A.2 Isothermal: w-velocity at t = 40

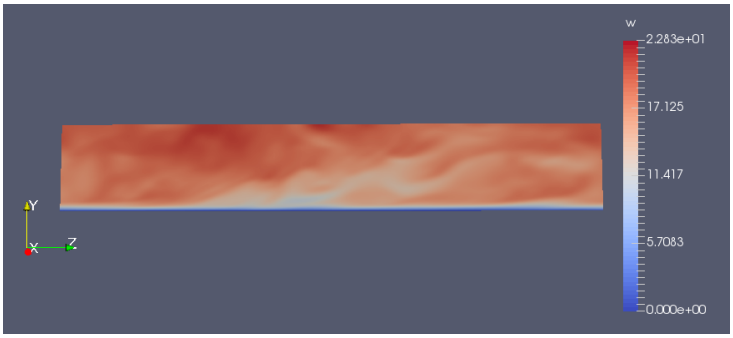


Fig. A.3 Isothermal: w-velocity at t = 50

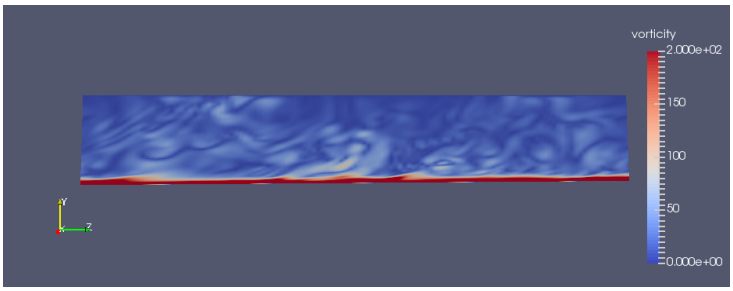


Fig. A.4 Isothermal: Vorticity at t = 30

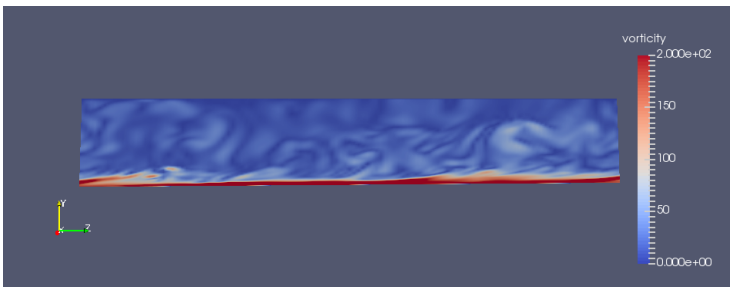


Fig. A.5 Isothermal: Vorticity at t = 40

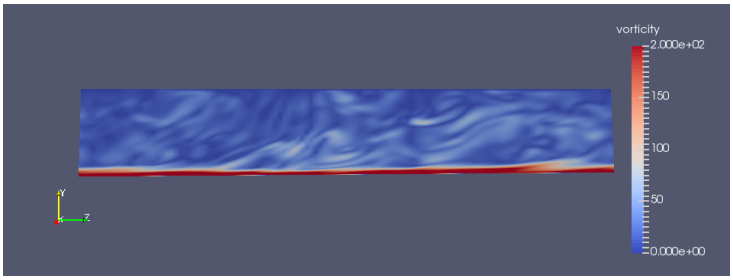


Fig. A.6 Isothermal: Vorticity at t = 50

A.1.2 Stratified

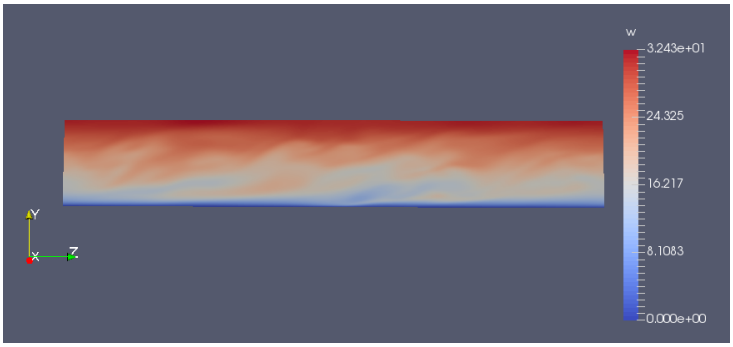


Fig. A.7 Stratified: w-velocity at t = 30

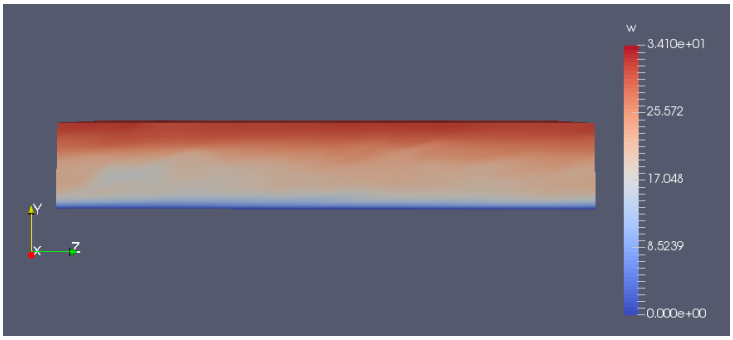


Fig. A.8 Stratified: w-velocity at t = 40

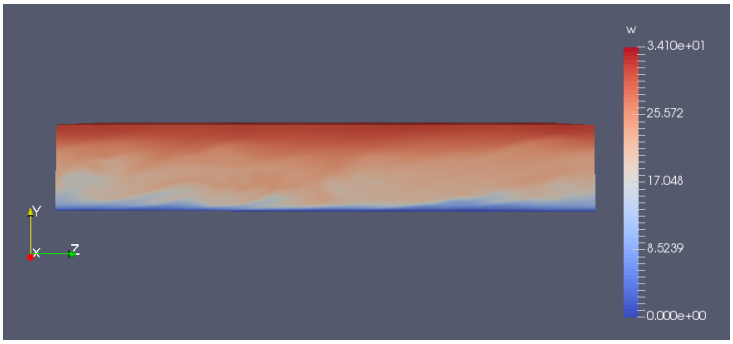


Fig. A.9 Stratified: w-velocity at t = 50

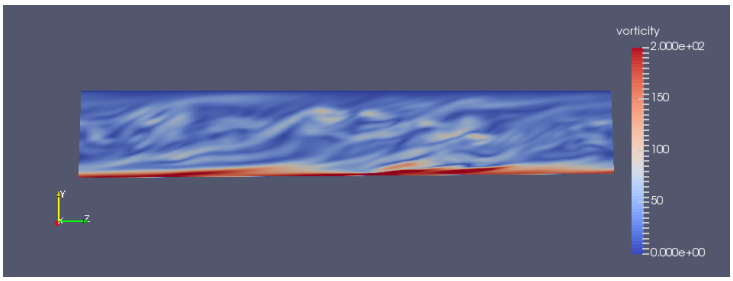


Fig. A.10 Stratified: Vorticity at t = 30

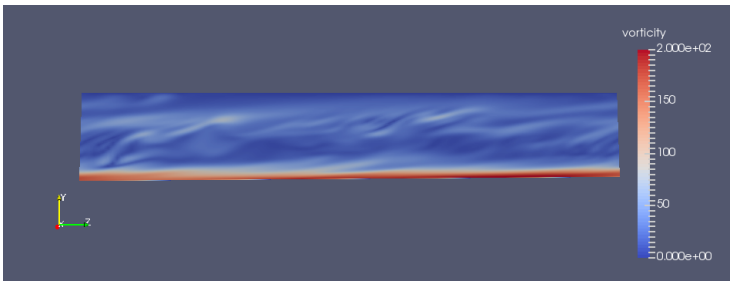


Fig. A.11 Stratified: Vorticity at t = 40

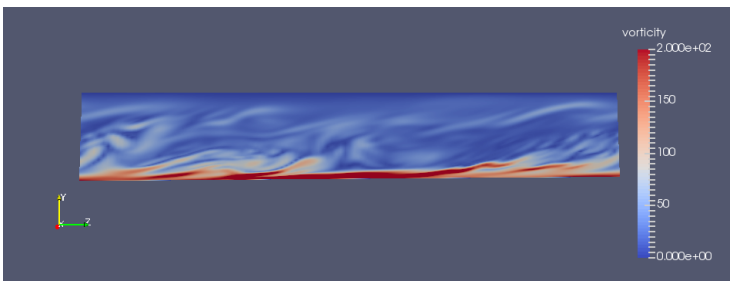


Fig. A.12 Stratified: Vorticity at t = 50

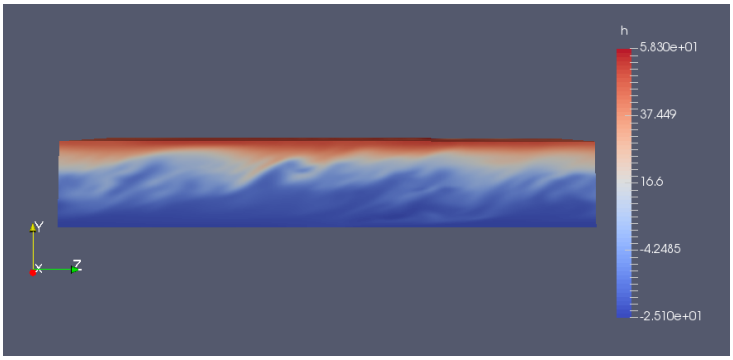


Fig. A.13 Stratified: ϕ at t = 30

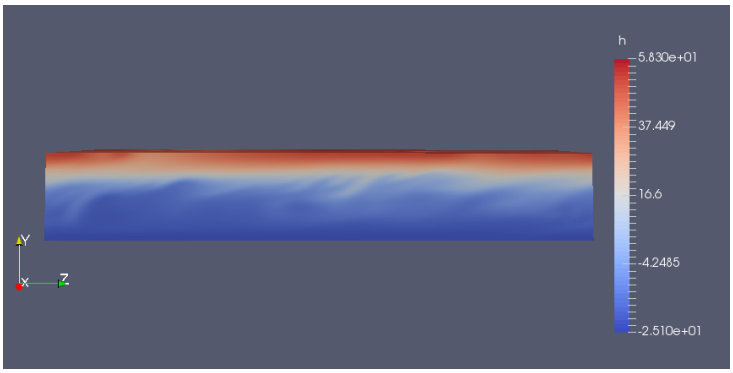


Fig. A.14 Stratified: ϕ at t = 40

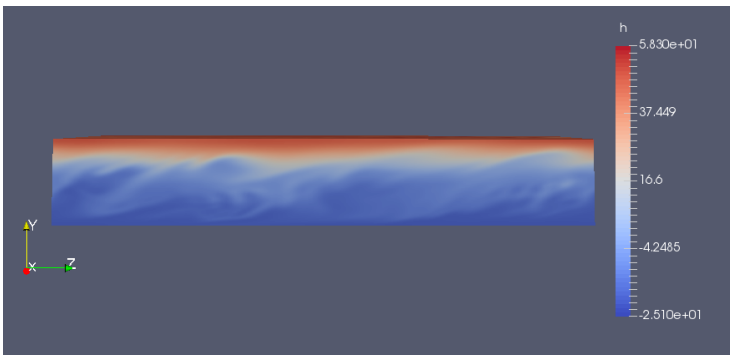


Fig. A.15 Stratified: ϕ at t = 50

A.2 KDE

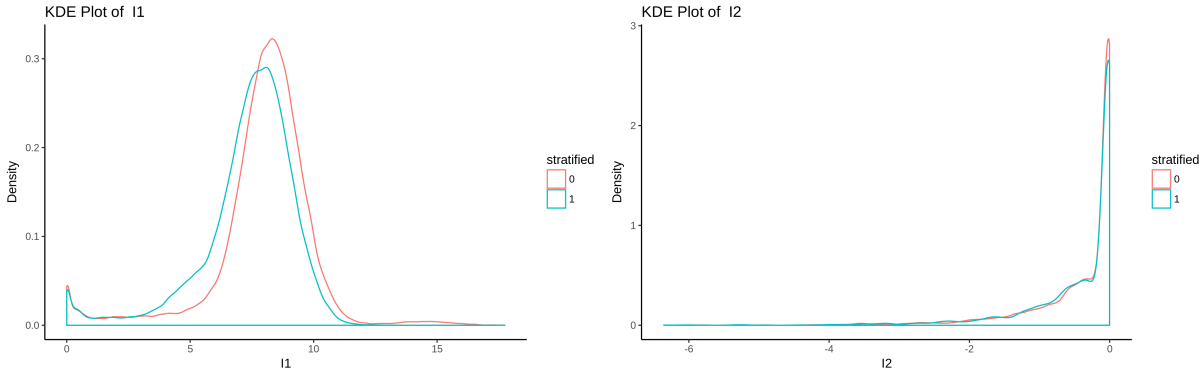


Fig. A.16 KDE of I_1

Fig. A.17 KDE of I_2

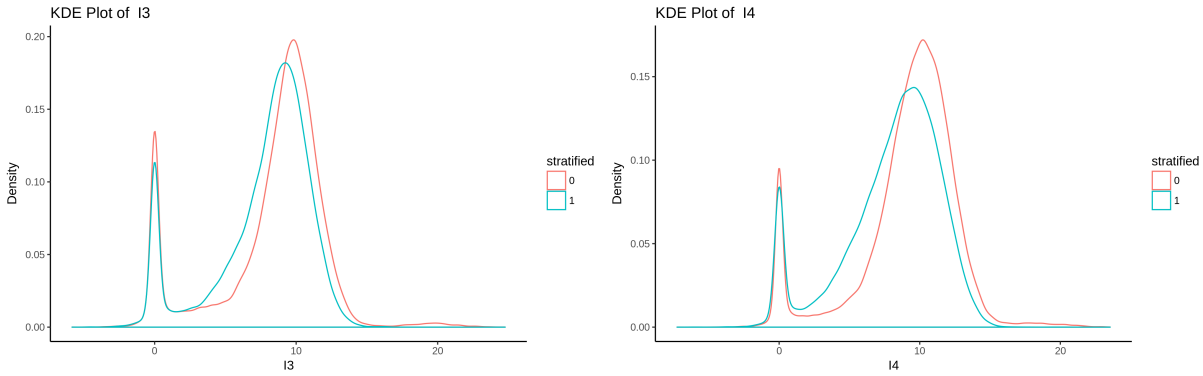


Fig. A.18 KDE of I_3

Fig. A.19 KDE of I_4

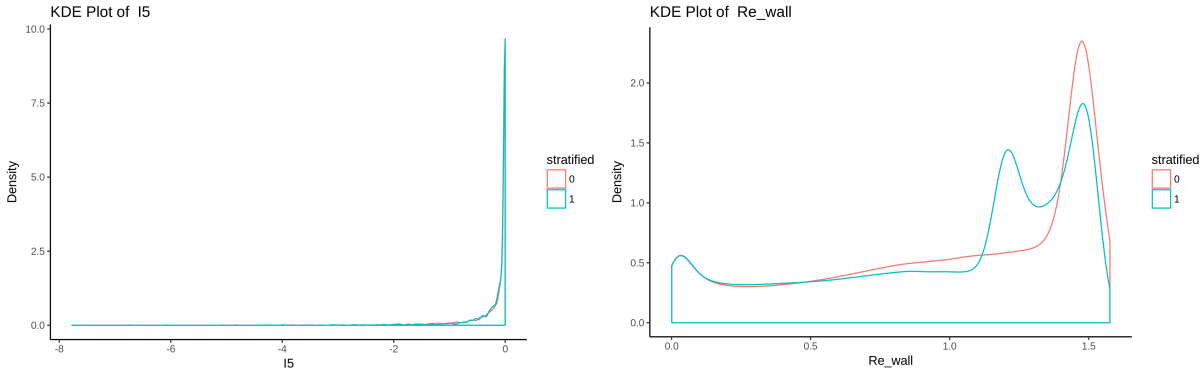


Fig. A.20 KDE of I_5

Fig. A.21 KDE of Re_{wall}

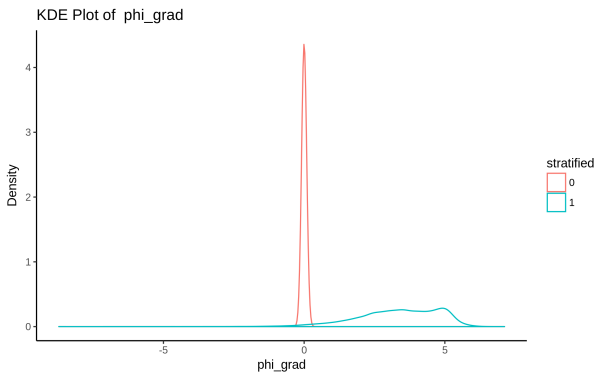


Fig. A.22 KDE of phi_grad

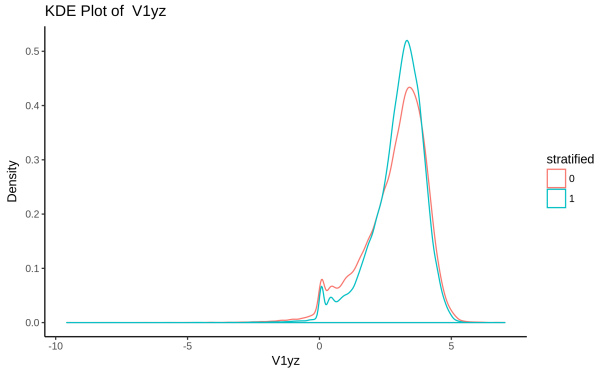


Fig. A.23 KDE of $T_{yz}^{(1)}$

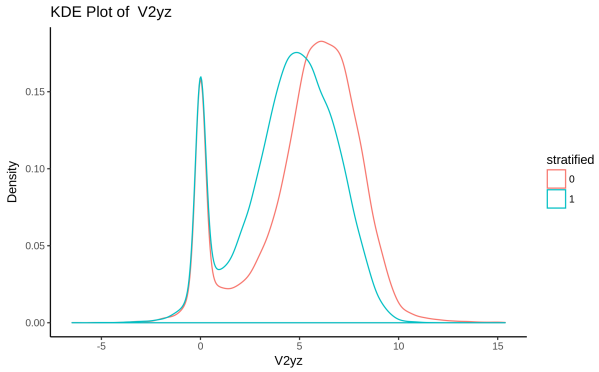


Fig. A.24 KDE of $T_{yz}^{(2)}$

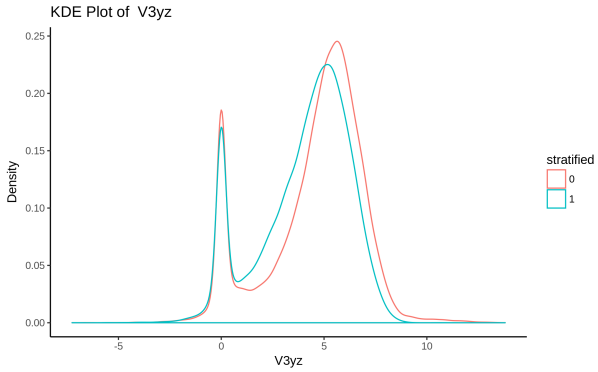


Fig. A.25 KDE of $T_{yz}^{(3)}$

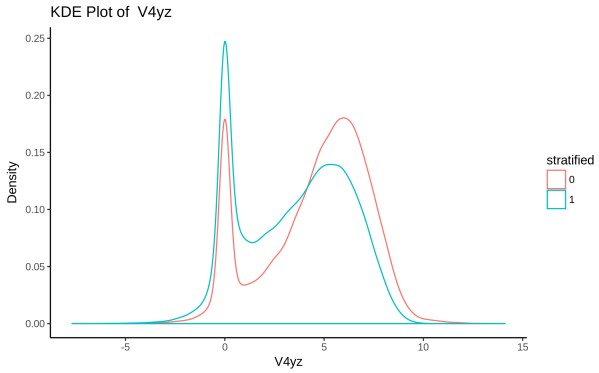


Fig. A.26 KDE of $T_{yz}^{(4)}$

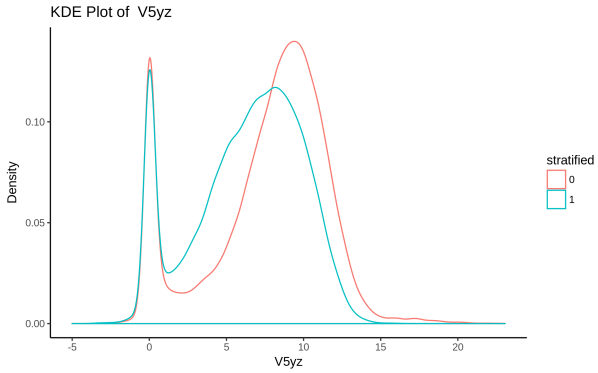


Fig. A.27 KDE of $T_{yz}^{(5)}$

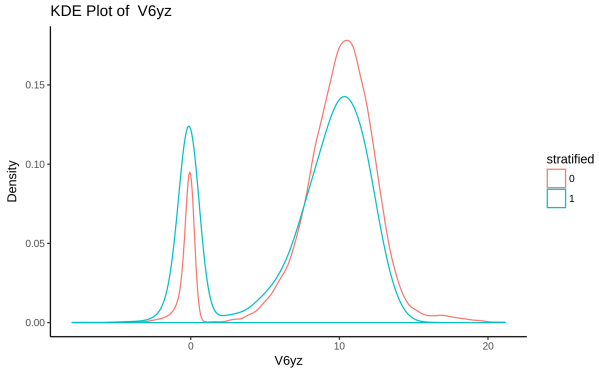


Fig. A.28 KDE of $T_{yz}^{(6)}$

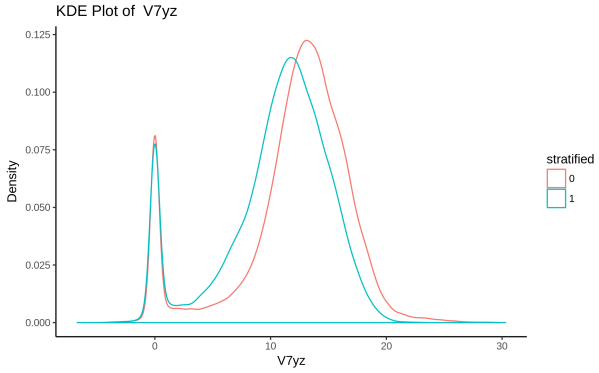


Fig. A.29 KDE of $T_{yz}^{(7)}$

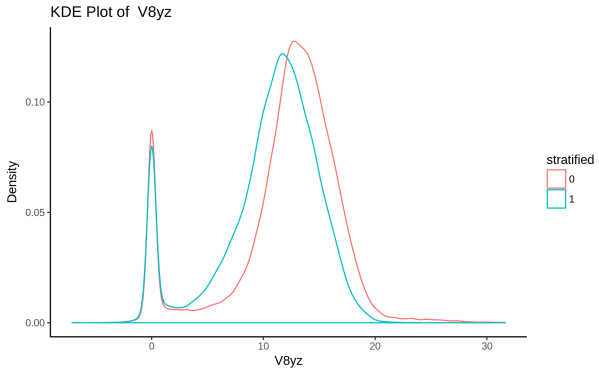


Fig. A.30 KDE of $T_{yz}^{(8)}$

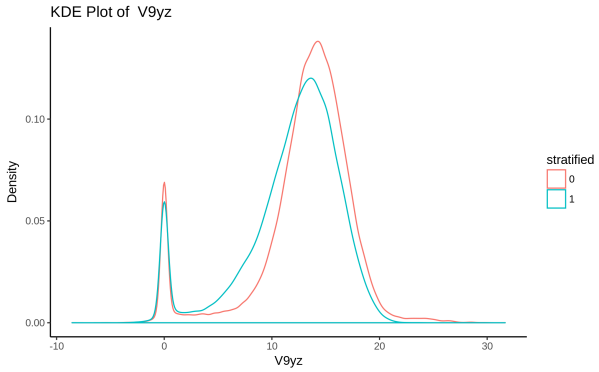


Fig. A.31 KDE of $T_{yz}^{(9)}$

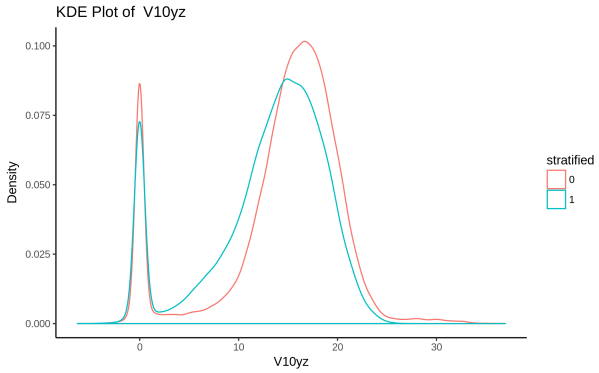


Fig. A.32 KDE of $T_{yz}^{(10)}$

A.3 Anisotropy Reynolds Stress vs Input Variable

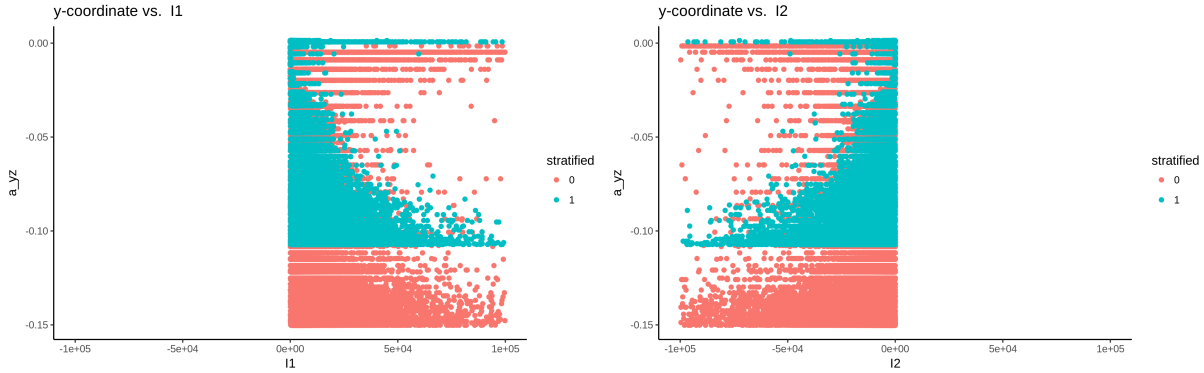


Fig. A.33 Scatter Plot of a_{yz} vs I_1

Fig. A.34 Scatter Plot of a_{yz} vs I_2

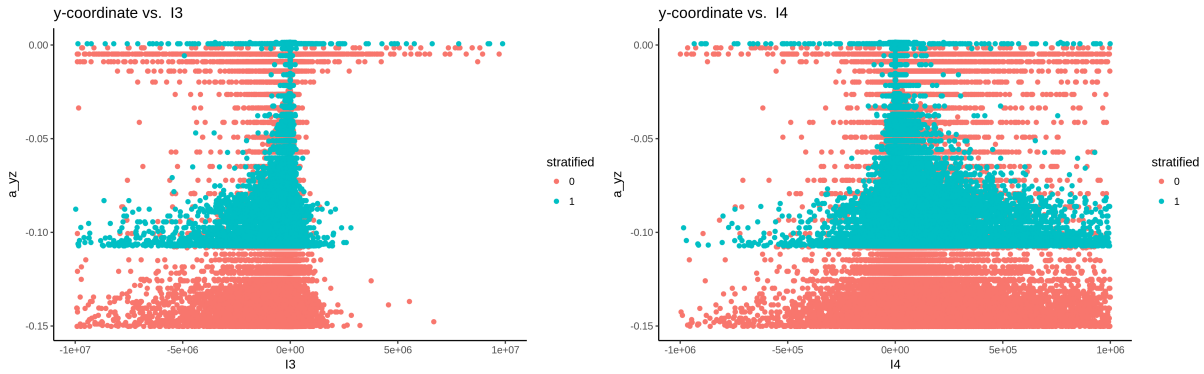


Fig. A.35 Scatter Plot of a_{yz} vs I_3

Fig. A.36 Scatter Plot of a_{yz} vs I_4

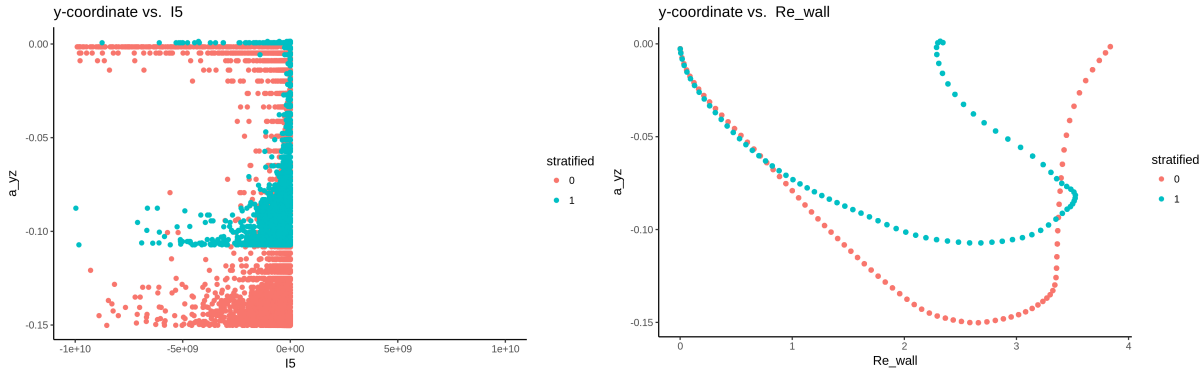


Fig. A.37 Scatter Plot of a_{yz} vs I_5

Fig. A.38 Scatter Plot of a_{yz} vs Re_{wall}

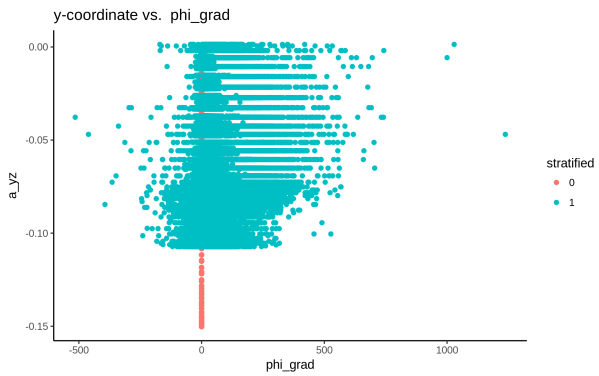


Fig. A.39 Scatter Plot of a_{yz} vs phi_grad

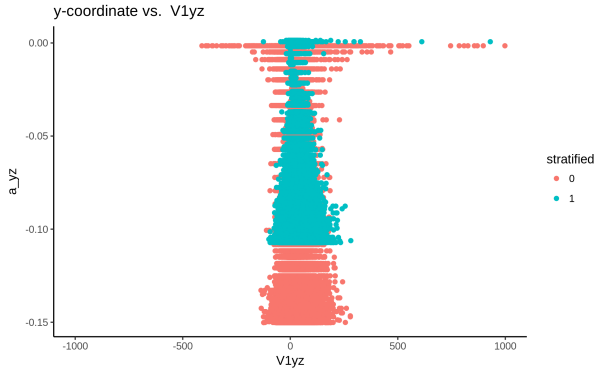


Fig. A.40 Scatter Plot of a_{yz} vs $T_{yz}^{(1)}$

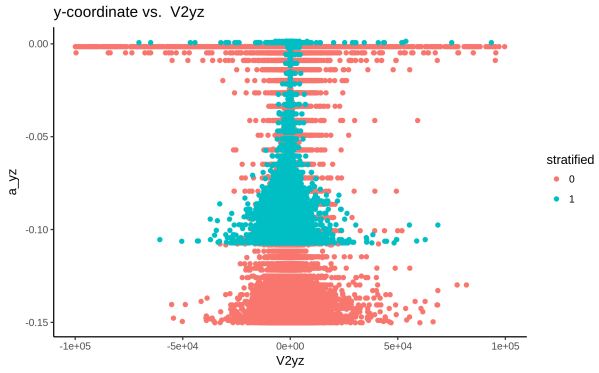


Fig. A.41 Scatter Plot of a_{yz} vs $T_{yz}^{(2)}$

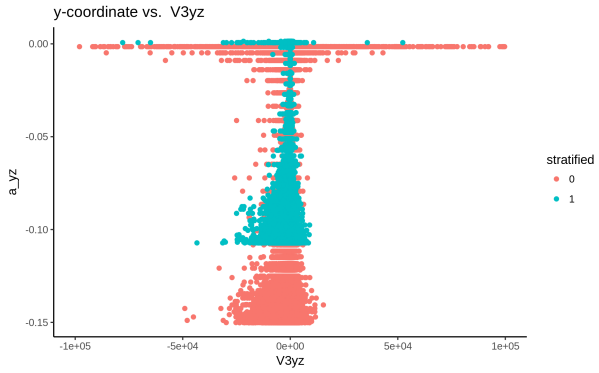


Fig. A.42 Scatter Plot of a_{yz} vs $T_{yz}^{(3)}$

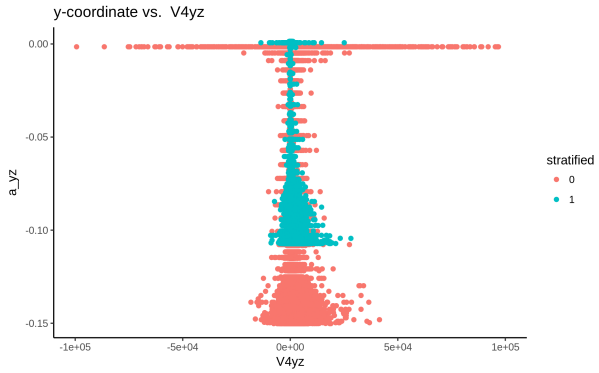


Fig. A.43 Scatter Plot of a_{yz} vs $T_{yz}^{(4)}$

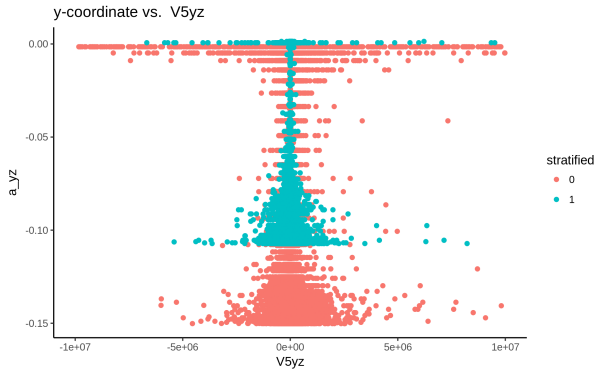


Fig. A.44 Scatter Plot of a_{yz} vs $T_{yz}^{(5)}$

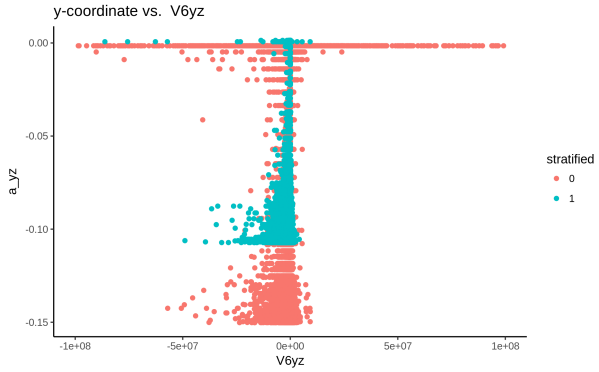


Fig. A.45 Scatter Plot of a_{yz} vs $T_{yz}^{(6)}$

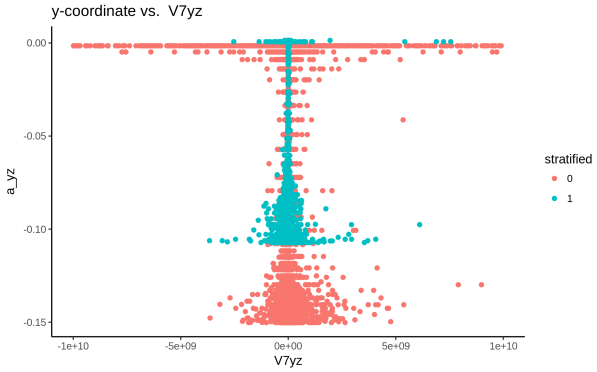


Fig. A.46 Scatter Plot of a_{yz} vs $T_{yz}^{(7)}$

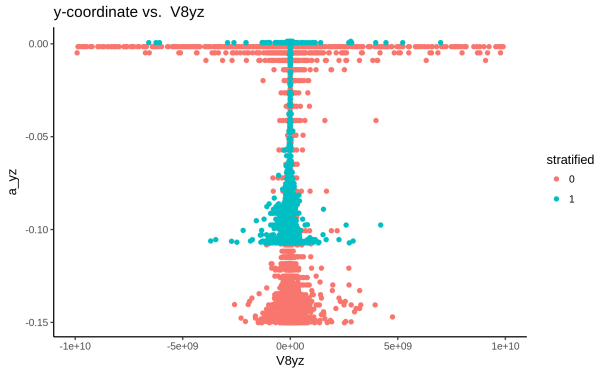


Fig. A.47 Scatter Plot of a_{yz} vs $T_{yz}^{(8)}$

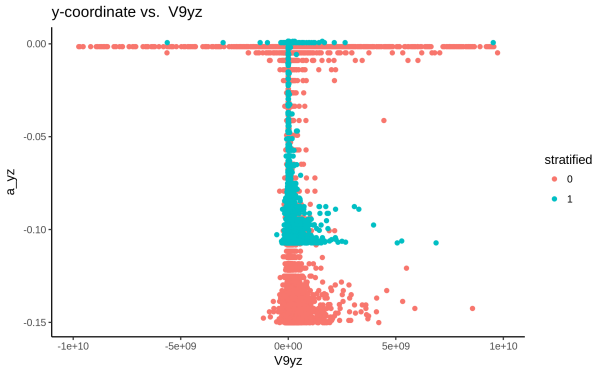


Fig. A.48 Scatter Plot of a_{yz} vs $T_{yz}^{(9)}$

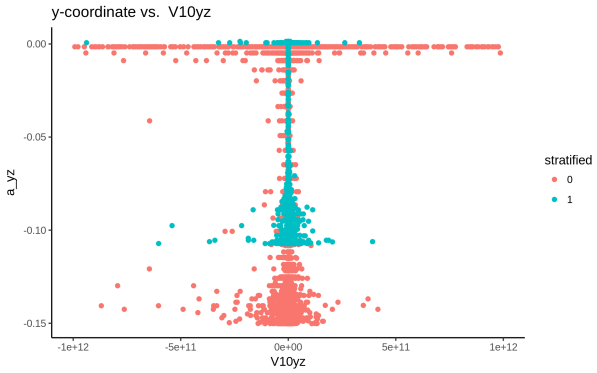


Fig. A.49 Scatter Plot of a_{yz} vs $T_{yz}^{(10)}$

A.4 Y-Coordinate vs Input Variable

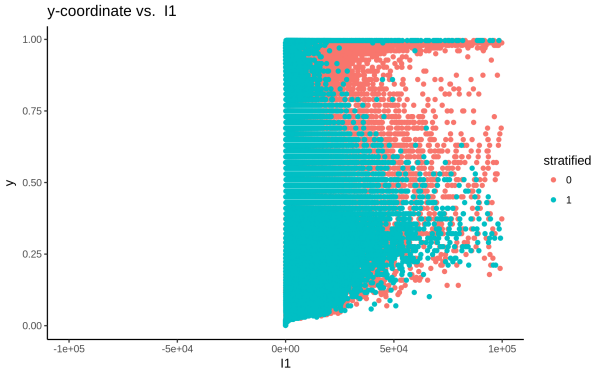


Fig. A.50 Scatter Plot of y-coordinate vs I_1

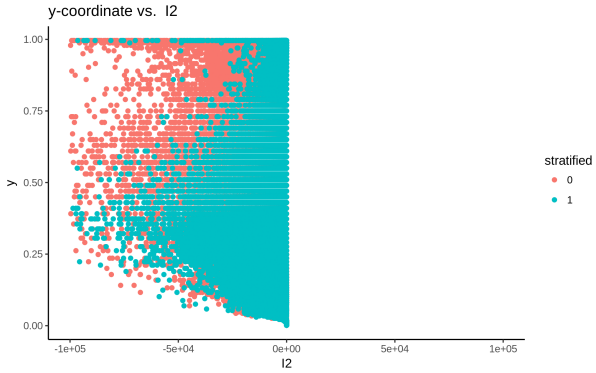


Fig. A.51 Scatter Plot of y-coordinate vs I_2

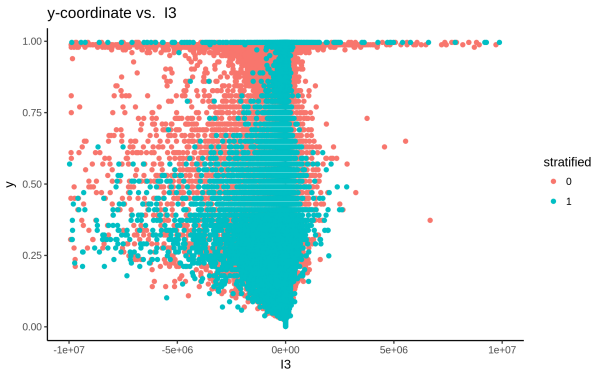


Fig. A.52 Scatter Plot of y-coordinate vs I_3

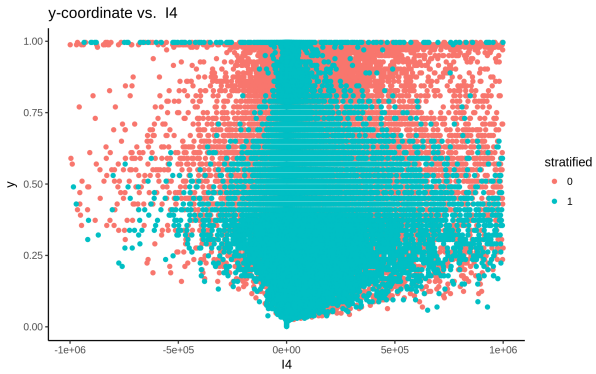


Fig. A.53 Scatter Plot of y-coordinate vs I_4

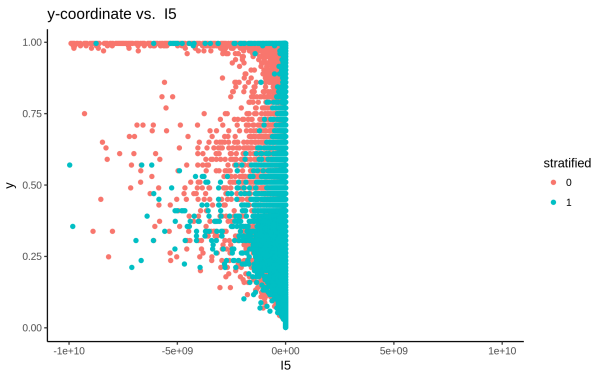


Fig. A.54 Scatter Plot of y-coordinate vs I_5

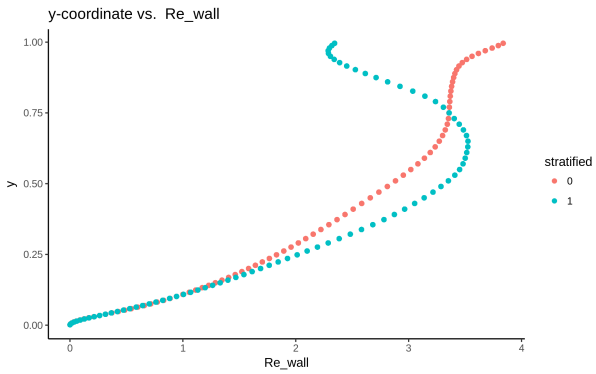


Fig. A.55 Scatter Plot of y-coordinate vs Re_{wall}

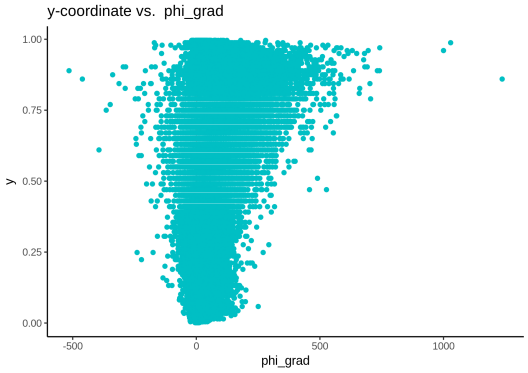


Fig. A.56 Scatter Plot of y-coordinate vs phi_grad

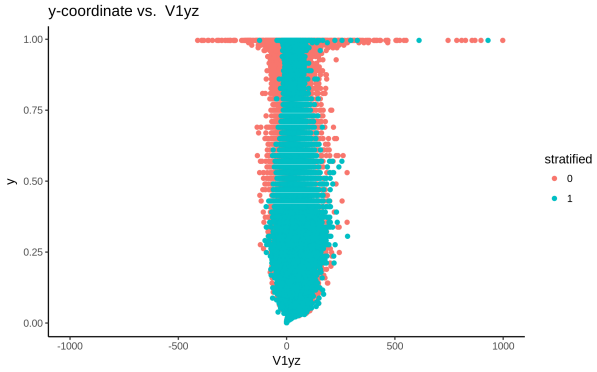


Fig. A.57 Scatter Plot of y-coordinate vs $T_{yz}^{(1)}$

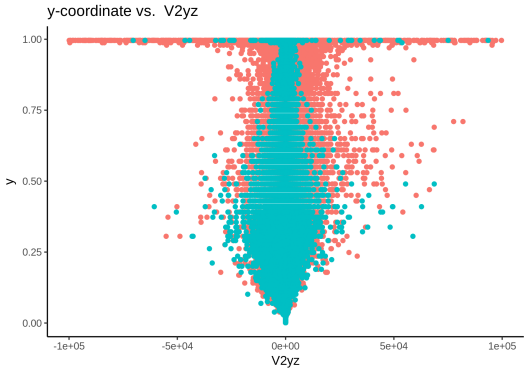


Fig. A.58 Scatter Plot of y-coordinate vs $T_{yz}^{(2)}$

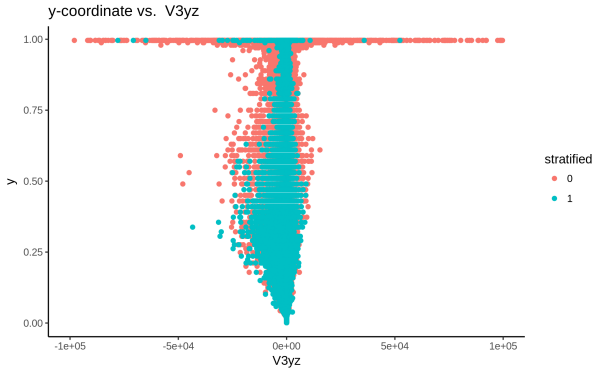


Fig. A.59 Scatter Plot of y-coordinate vs $T_{yz}^{(3)}$

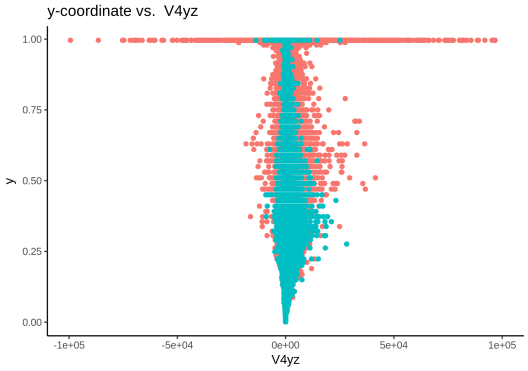


Fig. A.60 Scatter Plot of y-coordinate vs $T_{yz}^{(4)}$

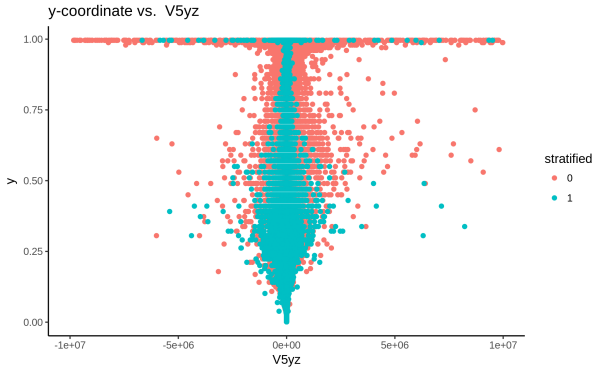


Fig. A.61 Scatter Plot of y-coordinate vs $T_{yz}^{(5)}$

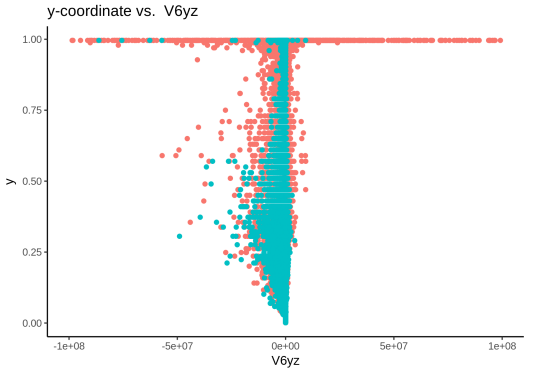


Fig. A.62 Scatter Plot of y-coordinate vs $T_{yz}^{(6)}$

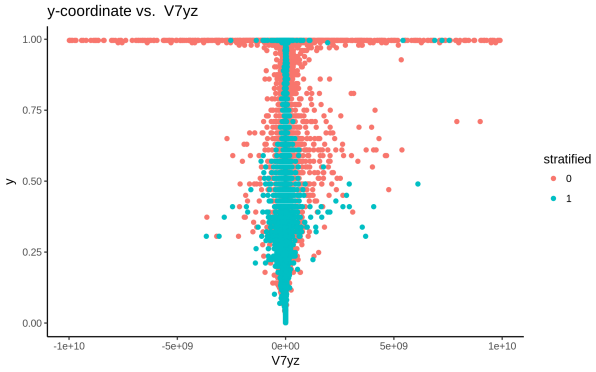


Fig. A.63 Scatter Plot of y-coordinate vs $T_{yz}^{(7)}$

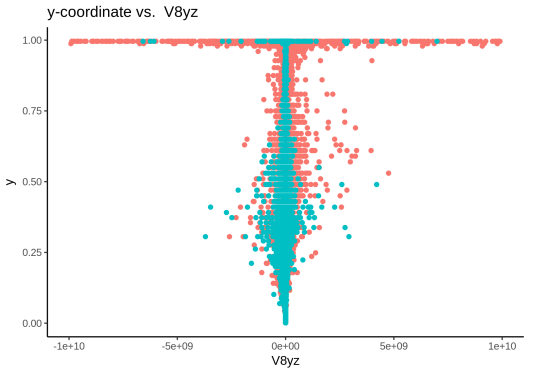


Fig. A.64 Scatter Plot of y-coordinate vs $T_{yz}^{(8)}$

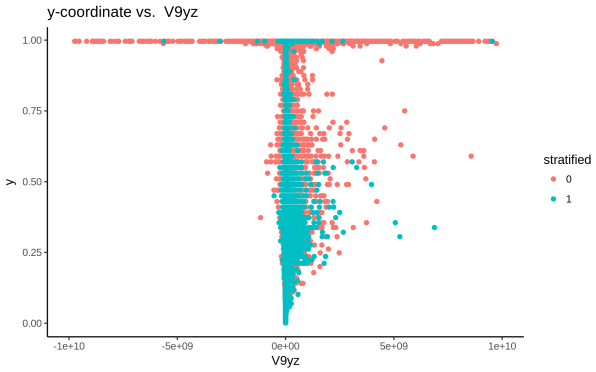


Fig. A.65 Scatter Plot of y-coordinate vs $T_{yz}^{(9)}$

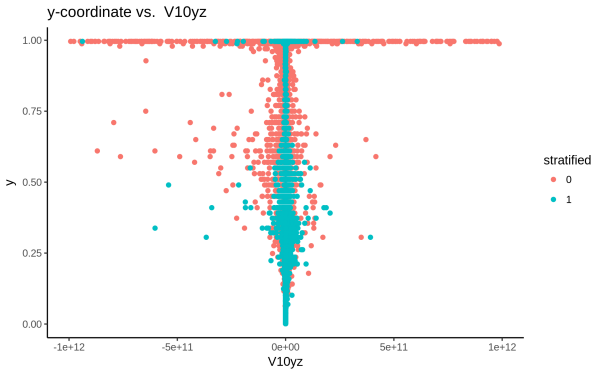


Fig. A.66 Scatter Plot of y-coordinate vs $T_{yz}^{(10)}$

A.5 Sensitivity Analysis

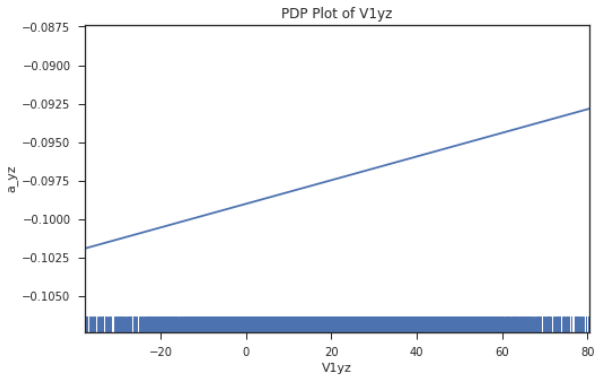


Fig. A.67 PDP of $T_{yz}^{(1)}$

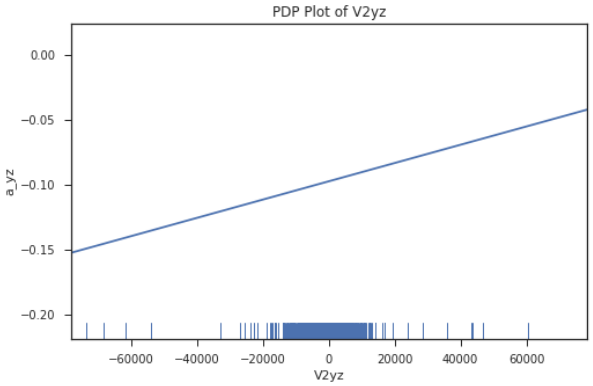


Fig. A.68 PDP of $T_{yz}^{(2)}$

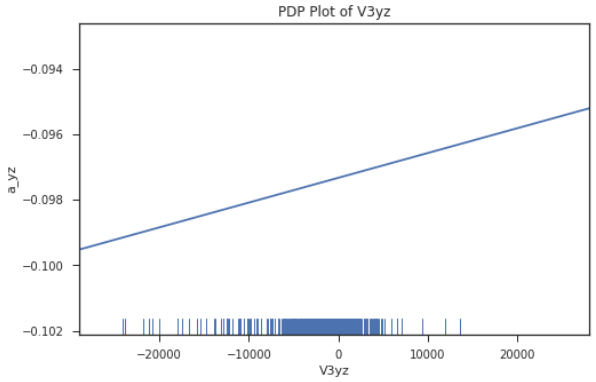


Fig. A.69 PDP of $T_{yz}^{(3)}$

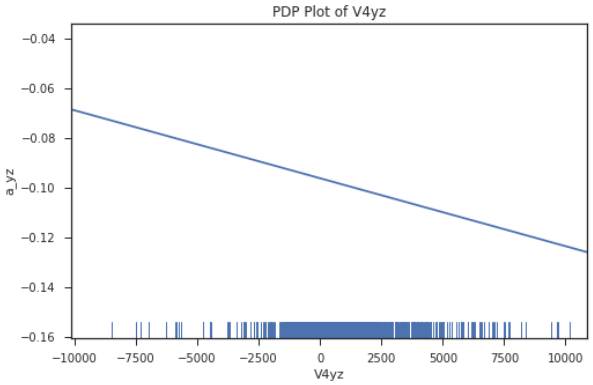


Fig. A.70 PDP of $T_{yz}^{(4)}$

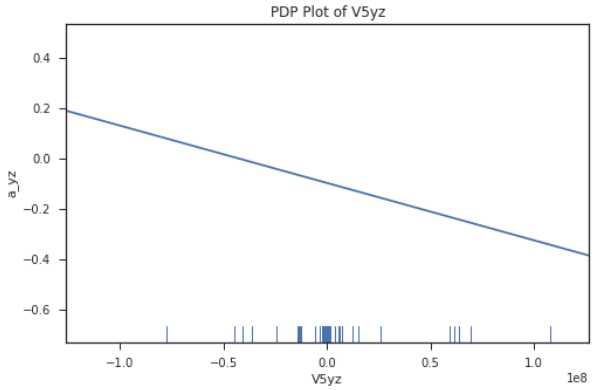


Fig. A.71 PDP of $T_{yz}^{(5)}$

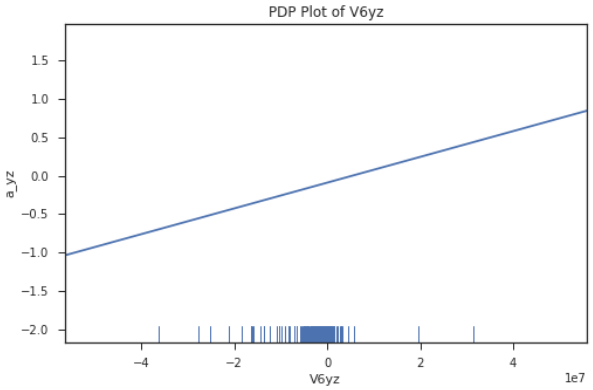


Fig. A.72 PDP of $T_{yz}^{(6)}$

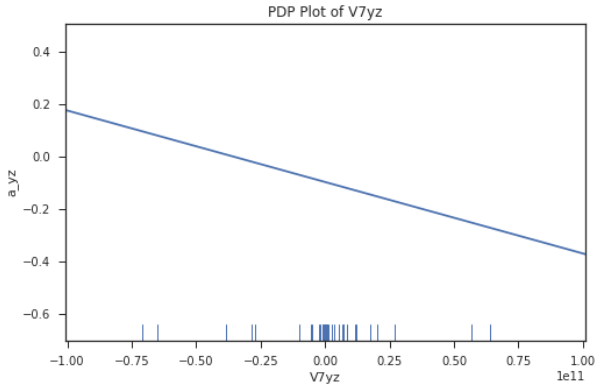


Fig. A.73 PDP of $T_{yz}^{(7)}$

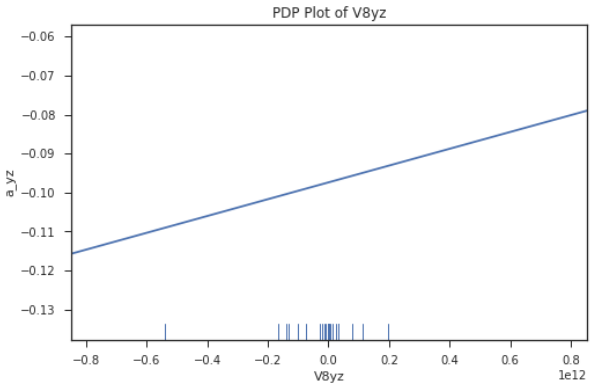


Fig. A.74 PDP of $T_{yz}^{(8)}$

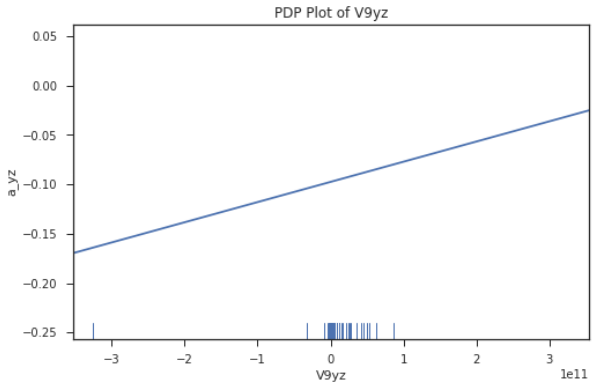


Fig. A.75 PDP of $T_{yz}^{(9)}$

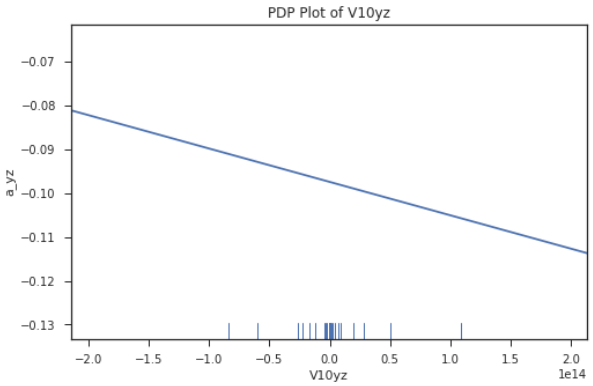


Fig. A.76 PDP of $T_{yz}^{(10)}$

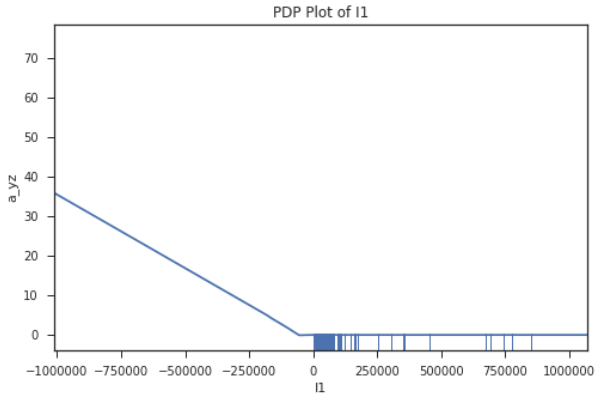


Fig. A.77 PDP of I_1

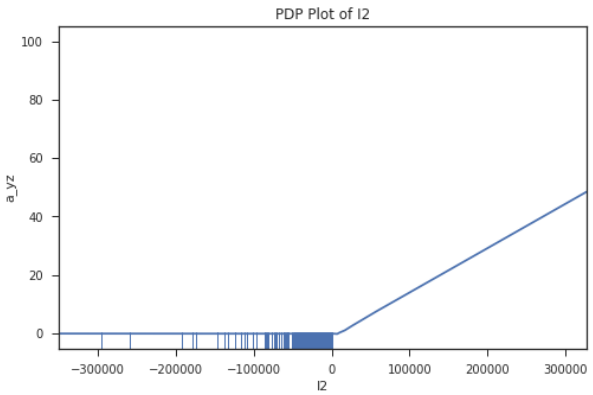


Fig. A.78 PDP of I_2

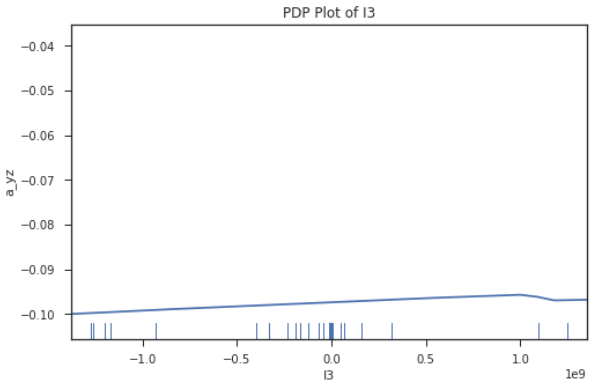


Fig. A.79 PDP of I_3

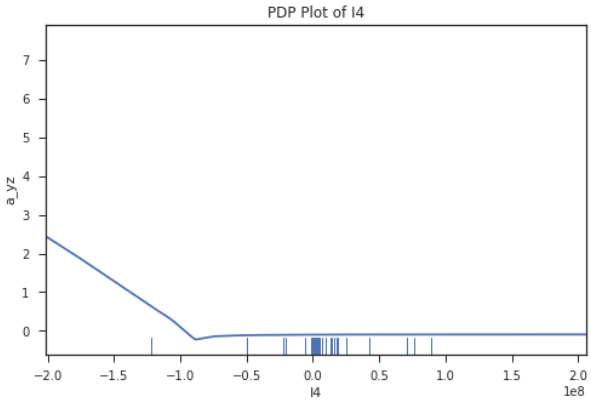


Fig. A.80 PDP of I_4

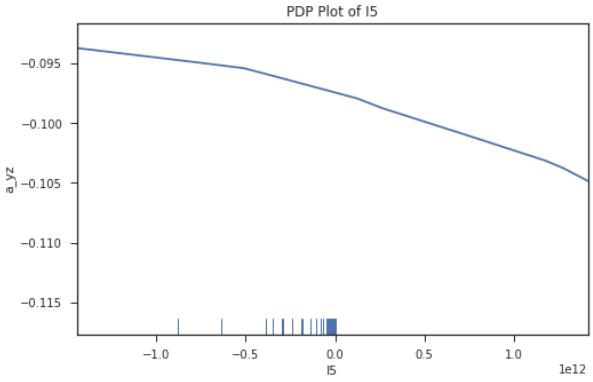


Fig. A.81 PDP of I_5

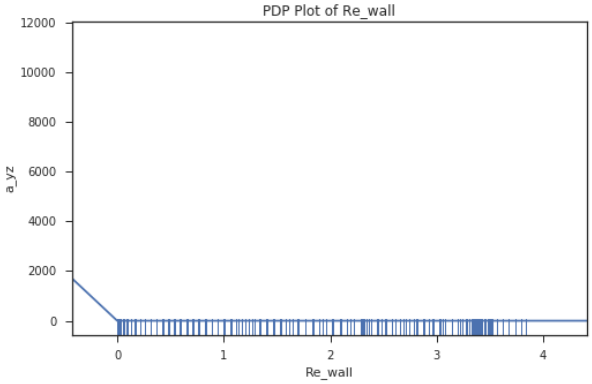


Fig. A.82 PDP of Re_{wall}

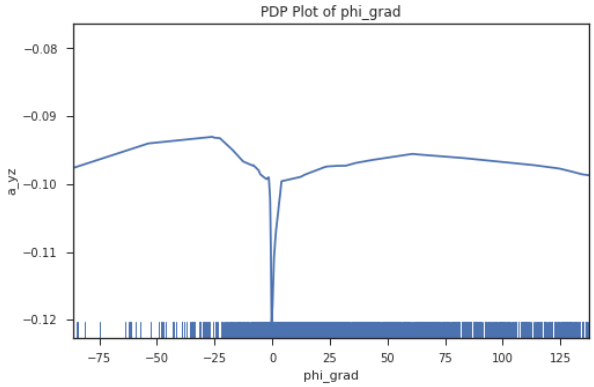


Fig. A.83 PDP of phi_grad

A.6 Bootstrapping

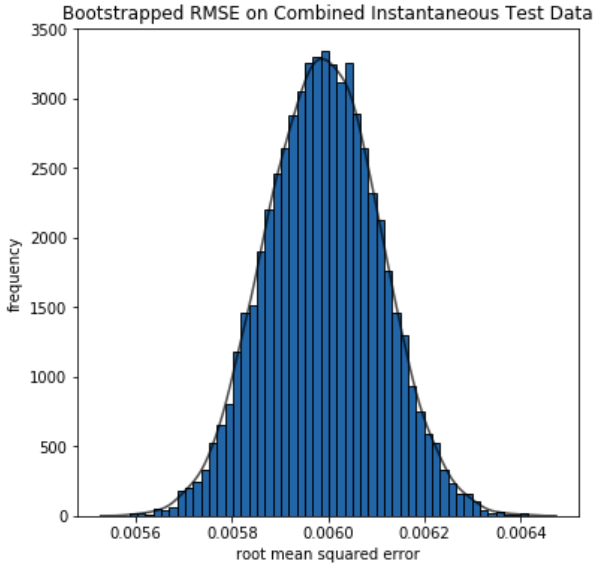


Fig. A.84 Combined: Bootstrapped RMSE Instantaneous Test Predictions

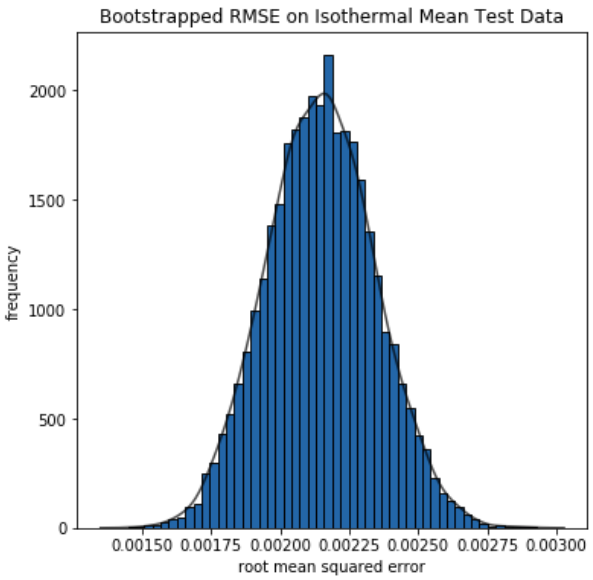


Fig. A.85 Isothermal: Bootstrapped RMSE Mean Test Predictions

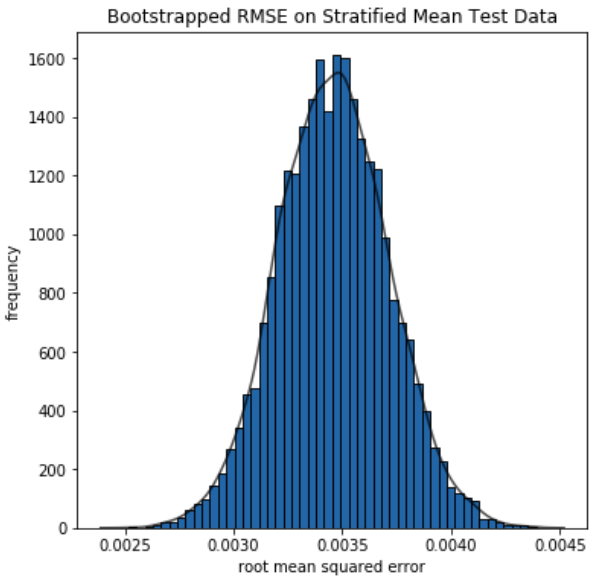


Fig. A.86 Stratified: Bootstrapped RMSE Mean Test Predictions

Appendix B

Final Model

This Appendix contains the details of the final Tensor Basis Neural Network (TBNN) that was developed in this investigation. For instance, B.1 shows the final set of hyperparameters that were determined using the manual grid search. This was the model that minimised the combined validation Root Mean Squared Error (RMSE).

Table B.1 Final Model Hyperparameters

	Model 15
Layer structure	50, 50, 50, 10
Data transformation	StandardScaler
Epochs	300
Batch size	32
Optimiser	adam
Weight initilisation	glorot_uniform
Dropout regularisation	none

Figure B.1 shows the parameters and an overview of the structure of this final model. This output is obtained from the *keras* neural network package in Python. Here, it can be seen, at the bottom of the figure, that there were 6,012 model terms in the TBNN structure.

Layer (type)	Output Shape	Param #	Connected to
input_5 (InputLayer)	(None, 7)	0	
dense_11 (Dense)	(None, 50)	400	input_5[0][0]
dense_12 (Dense)	(None, 50)	2550	dense_11[0][0]
dense_13 (Dense)	(None, 50)	2550	dense_12[0][0]
dense_14 (Dense)	(None, 10)	510	dense_13[0][0]
input_6 (InputLayer)	(None, 10)	0	
dot_3 (Dot)	(None, 1)	0	dense_14[0][0] input_6[0][0]
dense_15 (Dense)	(None, 1)	2	dot_3[0][0]

Total params: 6,012
Trainable params: 6,012
Non-trainable params: 0

Fig. B.1 Parameters Model 15

Finally, Figure B.2 is another output from the *keras* package. This figure shows a high-level overview of the TBNN and the various types of layers (input, hidden and output) that were used in the neural network.

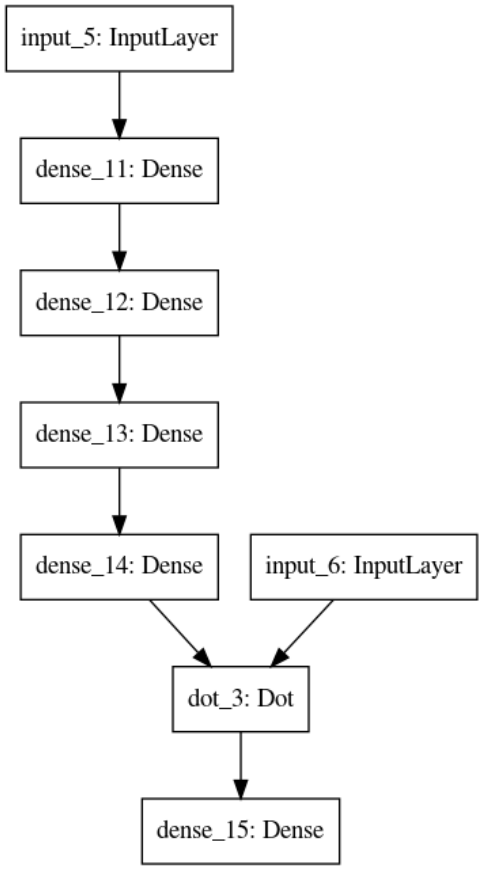


Fig. B.2 TBNN Structure Model 15

Appendix C

Turbulent Boundary Layers

This Appendix outlines how the turbulent boundary layers were calculated for both the isothermal and stratified open channel cases. These were referred throughout the discussion, linking how the turbulent regions affect the accuracy of the machine learning predictions. The equations provided throughout this appendix, can be found in Pope [49].

C.1 Isothermal

For a simple case of the isothermal open-channel flow, the boundary layer height can be obtained using Equation (C.1), where z is the distance from the leading edge of a plate.

$$\delta = \frac{0.382z}{Re_z^{1/5}} \tag{C.1}$$

Using Equation (C.1), where $z = 2\pi$ and $Re_z = 225$, a boundary layer height of $\delta = 0.812$m. For the outer region, the valid range is given by Equation (C.2), where y is the height above the plate.

$$0.3 < \frac{y}{\delta} < 1.0 \tag{C.2}$$

Hence, the outer layer extends up to $y = 0.812$m. Furthermore, the height of the log-law region is given by Equation (C.3).

$$y = 0.3\delta \tag{C.3}$$

Therefore, the log-law region extends up to $y = 0.244$m. Finally, the end of the buffer zone and viscous sub-layer are given by Equation (C.4),

$$y = \frac{y^+ \nu}{u_\tau}. \tag{C.4}$$

In this equation, $y^+ = 40$ for the end of the buffer zone, $y^+ = 5$ for the viscous sub-layer, with $\nu = 1/225$ and $u_\tau = 0.992$ which are both based on the large eddy simulation results from the PUFFIN output. Using these values, the end of the buffer zone is $y = 0.178$m and $y = 0.022$m for the viscous sub-layer.

This result confirms that the mesh outlined in the methodology, in Section 3.2.1, does have a cell within the viscous sub-layer (smallest grid size $\Delta y = 0.003$m). Hence, the results obtained in this sub-layer will be accurate.

C.2 Stratified

For the more complex stratified open-channel flow case, there does not exist an Equation such as (C.1) by which the various turbulent regions are defined as in the isothermal case. Instead, PUFFIN [25], the computational fluid dynamics software which was used to generate the data, instead defines the boundary layer height, δ, numerically with the condition in Equation (C.5).

$$\delta = max(d\phi dy) \tag{C.5}$$

Here, ϕ is the heat effect variable used throughout this thesis. Based on this numerical condition, the boundary layer thickness for the stratified open-channel flow case is $\delta = 0.792$m.

Hence, using Equations (C.3)-(C.4) as per the previous section, the log-law region extends up to $y = 0.234$m, buffer zone up to $y = 0.178$m, and the viscous sub-layer up to $y = 0.022$m. Again, this result confirms the accuracy of the predictions within the viscous sub-layer as the smallest cell within this region is less than the height of the viscous sub-layer.