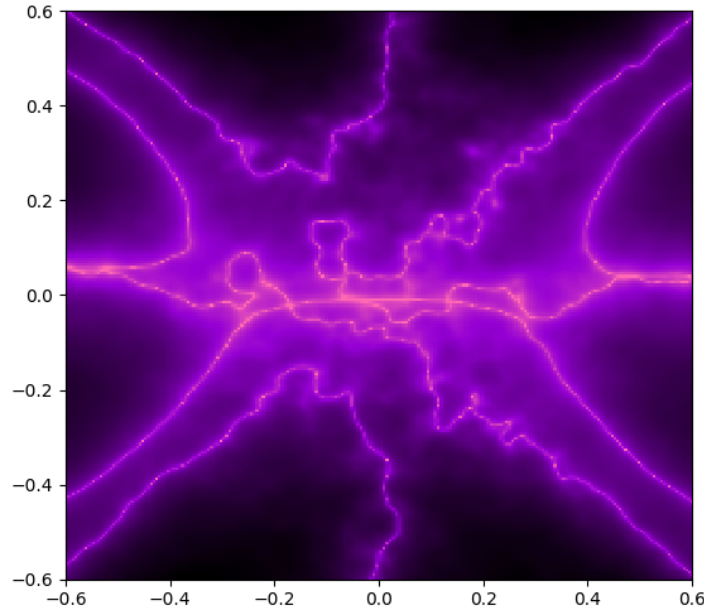




CHALMERS



Tillämpning av maskininlärning inom icke-linjär turbulensmodellering

Kandidatarbete vid institutionen för Mekanik och Maritima vetenskaper (M2)

Berg Thomsen, Per
Claesson, Hugo
Guné, Lucas
Hörnfeldt, Felix
Lehmann, Ludvig
Svanbro, Cassandra

INSTITUTIONEN FÖR MEKANIK OCH MARITIMA VETENSKAPER

CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2024
www.chalmers.se

KANDIDATARBETE I MEKANIK OCH MARITIMA VETENSKAPER

Tillämpning av maskininlärning inom icke-linjär turbulensmodellering

Berg Thomsen, Per
Claesson, Hugo
Guné, Lucas
Hörnfeldt, Felix
Lehmann, Ludvig
Svanbro, Cassandra



CHALMERS

Institutionen för Mekanik och Maritima Vetenskaper
Avdelningen för Strömningslära
CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2024

Tillämpning av maskininlärning inom icke-linjär turbulensmodellering

Berg Thomsen, Per
Claesson, Hugo
Guné, Lucas
Hörnfeldt, Felix
Lehmann, Ludvig
Svanbro, Cassandra

© Berg Thomsen, Per | Claesson, Hugo | Guné, Lucas | Hörnfeldt, Felix
Lehmann, Ludvig | Svanbro, Cassandra | 2024

Handledare: Davidson, Lars | Institutionen för Mekanik och Maritima vetenskaper
(M2)

Examinator: Andersson, Niklas | Institutionen för Mekanik och Maritima vetenska-
per (M2)

Kandidatarbete 2024
Institutionen för Mekanik och Maritima Vetenskaper
Chalmers Tekniska Högskola
SE-412 96 Göteborg
Telefon +46 31 772 1000

Omslagsbild: Heatmap-plot av logaritmen av absolutbeloppet av element A_{23} från
ekvation (2.17) för Square Duct med friktions-Reynoldstal 1000.

Typsatt i L^AT_EX
Göteborg 2024

Tillämpning av maskininlärning inom icke-linjär turbulensmodellering

Berg Thomsen, Per
Claesson, Hugo
Guné, Lucas
Hörnfeldt, Felix
Lehmann, Ludvig
Svanbro, Cassandra

Institutionen för Mekanik och Maritima Vetenskaper
Avdelningen för Strömningslära
Chalmers Tekniska Högskola

Sammandrag

Turbulensmodellering är en vital del inom strömningsmekanik, där Navier-Stokes ekvationer ofta måste lösas numeriskt. Detta är väldigt datorkrävande när turbulens uppstår, vilket skapar ett behov av förenklande turbulensmodeller. Genom att använda tidsmedelvärdade Navier-Stokes ekvationer tillsammans med $k - \varepsilon$ modellen och en icke-linjära Eddy Viscosity modell kan man minska komplexiteten hos beräkningar. Huvudsyftet med arbetet är att använda neurala nätverk och Random Forest Regression för att undersöka om konstanterna som används i dessa modeller kan ersättas med funktioner, för att erhålla bättre resultat. Arbetet fann att användningen av ML-konstruerade funktioner presterade bättre än konstanta standardvärden i två av tre fall, samt att neurala nätverk generellt fungerade bättre än random forest regression. Resultaten är lovande och talar för att detta är ett område som bör undersökas vidare.

Nyckelord: Turbulensmodellering, Reynoldsspänningar, maskininlärning, neurala nätverk, Random Forest Regression, Square Duct, icke-linjär Eddy Viscosity model, $k - \varepsilon$ modell, tidsmedelvärda Navier-Stokes ekvationer.

Applications of machine learning in non-linear turbulence modelling

Berg Thomsen, Per
Claesson, Hugo
Guné, Lucas
Hörnfeldt, Felix
Lehmann, Ludvig
Svanbro, Cassandra

Department of. Mechanics and Maritime Sciences
Division Fluid Dynamics
Chalmers University of Technology

Abstract

Turbulence modeling is a vital part of fluid mechanics, where the Navier-Stokes equations often need to be solved numerically. This is computationally demanding due to turbulence, creating a need for simplified turbulence models. By using Reynolds-Averaged Navier Stokes together with the $k - \varepsilon$ model and a non-linear Eddy Viscosity model, one can reduce the complexity of calculations. The main purpose of this study is to use neural networks and Random Forest Regression to investigate whether the constants used in these models can be replaced with functions, which yield better results. The study found that the use of ML-constructed functions performed better than constant values in two out of three cases, and that neural networks generally worked better than Random Forest Regression. The results are promising and suggest that this is an area that should be further explored.

Keywords: Turbulence modelling, Reynold stresses, machine learning, neural network, Random Forest Regression, deep learning, Square Duct, non-linear Eddy Viscosity model, $k - \varepsilon$ model, Reynolds-Averaged Navier Stokes.

Författarnas tack

Vi i kandidatgruppen vill först och främst tacka vår handledare Lars Davidson som väglett oss genom detta arbete. Tack för all hjälp när vi varit osäkra på vad som gäller och för den kontinuerliga responsen som gjort att vi känt oss på rätt väg genom hela arbetet. Vi vill också tacka vår examinator Niklas Andersson för all hjälp under arbetes gång.

Kandidatgruppen, Göteborg, Maj, 2024

Akronymer

Nedan är listan över akronymer som har använts genom hela denna avhandling listade i alfabetisk ordning:

DNS	Direkt Numerisk Simulering
GridSearchCV	GridSearch Cross Validation
MaAE	Maximum Absolut Error
MAPE	Mean Average Percentage Error
ML	Maskininlärning
NEVM	Icke-linjär Eddy Viscosity Modell (Non-linear Eddy Viscosity Model)
NN	Neuralt nätverk
RANS	Tidsmedelvärda Navier-Stokes ekvationer (Reynolds Averaged Navier-Stokes)
RFR	Random Forest Regression
RMSPE	Root Mean Square Percentage Error
XGB	Extreme Gradient Boost

Nomenklatur

Nedan är nomenklaturen för index, uppsättningar, parametrar och variabler som har använts genom hela denna avhandling.

Index

i, j Index

Variabler

$\overline{u'u'}, \overline{u'v'}, \overline{u'w'},$
 $\overline{v'v'}, \overline{v'w'}, \overline{w'w'},$ Eftersökta turbulenta Reynoldsspänningar
 Re Reynoldstal
 ρ Densitet av medium
 μ Dynamisk viskositet
 L Karakteristisk längd
 ν Kinematisk viskositet
 τ_w Väggskevspänning
 u_τ Friktionshastighet
 Re_τ Friktions-Reynoldstal
 ℓ Längdskala
 ε Dissipation av turbulent kinetisk energi
 ν_t Turbulent kinematisk viskositet
 k Turbulent kinetisk energi
 a_{ij} Anisotropisk tensor
 s_{ij} Skjuvningstensor
 Ω_{ij} Rotationstensor
 δ_{ij} Kronecker delta

$c_i, i = 1, \dots, 7$ Eftersökta konstanter

b_i Nod (neuron)

x_j Indata

w_{ij} Indata vikt

θ_i Indata bias

g Aktiveringsfunktion

Innehåll

Akronymer	vii
Nomenclature	viii
Figurer	xii
Tabeller	xiv
1 Inledning	1
1.1 Bakgrund	1
1.2 Syfte	1
1.3 Problemformulering	1
1.4 Avgränsningar	2
2 Teori	3
2.1 Strömningsmekanik	3
2.1.1 Friktions-Reynoldstal	3
2.1.2 $k - \varepsilon$ modellen	4
2.1.3 Den Icke-linjära Eddy Viscosity modellen (NEVM)	5
2.1.4 Flöde genom olika geometrier	7
2.1.5 Flöde genom Square Duct	9
2.2 Maskininlärning	10
2.2.1 Neurala nätverk	10
2.2.2 Random Forest Regression	12
2.2.2.1 Extreme Gradient Boosting	13
3 Metod	14
3.1 Databehandling och val av in- och utdata för modeller	14
3.2 Neurala nätverk	15
3.3 Random Forest Regression	16
3.4 Felberäkning	16
4 Resultat	18
4.1 Kanalströmning	18
4.2 Gränsskikt	19
4.3 Generalisering mellan kanalströmning & gränsskikt	21
4.4 Gropgeometri	21
4.5 Square Duct	23

4.5.1	Neurala nätverk	23
4.5.2	Random Forest Regression	25
4.6	Generalisering mellan friktions-Reynoldstal för Square Duct	27
4.6.1	Neuralt nätverk	27
4.6.2	Random Forest Regression	28
5	Diskussion	29
5.1	Kanalströmning och gränsskikt	29
5.2	Gropgeometri	30
5.3	Square Duct	30
5.3.1	Neuralt nätverk	31
5.3.2	Random Forest Regression	32
6	Slutsatser	33
	Litteraturförteckning	35
A	Uttryck för okända konstanter	I
B	Komplett kod	II

Figurer

2.1	Geometrin för två fall av flöde.	7
2.2	Geometrin för flöde med en grop.	8
2.3	Geometrin för ett fyrkantigt rör, även kallat Square Duct.	10
2.4	Figuren visar processen för summation mellan indata och utdata med hjälp av bias och aktiveringsfunktion för en given nod.	11
2.5	Exempel på hur lagerstrukturen hos ett neuralt nätverk kan se ut.	11
2.6	Struktur hos ett godtyckligt beslutsträd.	12
3.1	Heatmap plots av $\arctan(c_0)$, $\sqrt[3]{A_{43}}$ och $\sqrt[3]{A_{32}}$ i Square Duct med friktions-Reynoldstal 150. Arctangens och kubrötter har tagits för att göra de delade geometriska mönstren synliga.	15
4.1	Prediktioner för c_i för det neurala nätverket och RFR, jämfört med det sanna värdet från DNS-data.	19
4.2	Reynoldsspänningar för kanalströmning predikterade med neurala nätverket, RFR och konstanta c_i , jämfört med de sanna värdena från DNS-datan.	19
4.3	Prediktioner för c_i för det neurala nätverket och RFR, jämfört med det sanna värdet från DNS-data.	20
4.4	Reynoldsspänningar för gränsskikt predikterade med neurala nätverket, RFR och konstanta c_i , jämfört med de sanna värdena från DNS-datan.	20
4.5	Det absoluta felet för stresstensorerna $\overline{u'u'^+}$, $\overline{u'v'^+}$, $\overline{v'v'^+}$ och $\overline{w'w'^+}$ för det neurala nätverket, jämfört med det sanna värdet från DNS-data.	22
4.6	Det absoluta felet för stresstensorerna $\overline{u'u'^+}$, $\overline{u'v'^+}$, $\overline{v'v'^+}$ och $\overline{w'w'^+}$ för RFR, jämfört med det sanna värdet från DNS-data.	22
4.7	Reynoldsspänning $\overline{u'w'}$ i flöde Square Duct med friktions-Reynoldstal 1000 plottad mot y^+ . Figuren visar DNS-data, resultat av modell med konstant c_i och modell med neuralt nätverk tränat på samma friktions-Reynoldstal.	24
4.8	Reynoldsspänning $\overline{u'w'}$ i flöde Square Duct med friktions-Reynoldstal 1000 plottad mot y^+ . Figuren visar DNS-data, resultat av modell med konstant c_i och modell med neuralt nätverk tränat på samma friktions-Reynoldstal.	24

4.9	Predikterade Reynoldspänningar $\widehat{u'v'}$ och $\widehat{u'w'}$ i flöde Square Duct med friktions-Reynoldstal 1000 plottad mot de sanna värdena $\overline{u'v'}$ och $\overline{u'w'}$. Graferna innehåller resultat för konstanta c_i och för modell med neuralt nätverk tränat på samma friktions-Reynoldstal.	25
4.10	Reynoldsspänning $\overline{u'u'}$ i flöde Square Duct med friktions-Reynoldstal 1000 plottad mot y^+ . Figuren visar DNS-data, resultat av modell med konstant c_i och modell med RFR tränat på samma friktions-Reynoldstal.	26
4.11	Reynoldsspänning $\overline{v'w'}$ i flöde Square Duct med friktions-Reynoldstal 1000 plottad mot y^+ . Figuren visar DNS-data, resultat av modell med konstant c_i och modell med RFR tränat på samma friktions-Reynoldstal.	26
4.12	Predikterade Reynoldspänningar $\widehat{u'v'}$ och $\widehat{u'w'}$ i flöde Square Duct med friktions-Reynoldstal 1000 plottad mot de sanna värdena $\overline{u'v'}$ och $\overline{u'w'}$. Graferna innehåller resultat för konstanta c_i och för modell med RFR tränat på samma friktions-Reynoldstal.	27

Tabeller

4.1	Fel för neuralt nätverk och RFR, tränade och testade på kanalströmning med friktions-Reynoldstal 5200.	18
4.2	Fel för neuralt nätverk och RFR, tränade och testade på gränsskikt med friktions-Reynoldstal 2500.	20
4.3	Fel för neuralt nätverk och RFR, tränade på turbulens av typ kanalströmning med friktions-Reynoldstal 5200 och testade på gränsskikt med friktions-Reynoldstal 2500.	21
4.4	Fel för neuralt nätverk och RFR, tränade på turbulens av typen gränsskikt med friktions-Reynoldstal 2500 och testade på kanalströmning med friktions-Reynoldstal 5200.	21
4.5	Fel för neuralt nätverk och RFR, tränade och testade på turbulens med friktions-Reynoldstal 550 simulerat över gropgeometri.	23
4.6	Fel för neuralt nätverk, tränat på Square Duct med friktions-Reynoldstal 1000, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 1000.	23
4.7	Fel för neuralt nätverk, tränat på Square Duct med friktions-Reynoldstal 500, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 500.	23
4.8	Fel för RFR, tränat på Square Duct med friktions-Reynoldstal 1000, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 1000.	25
4.9	Fel för RFR, tränat på Square Duct med friktions-Reynoldstal 500, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 500.	25
4.10	Fel för neuralt nätverk, tränat på Square Duct med friktions-Reynoldstal 1000, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 500.	27
4.11	Fel för neuralt nätverk, tränat på Square Duct med friktions-Reynoldstal 500, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 1000.	28
4.12	Fel för RFR, tränat på Square Duct med friktions-Reynoldstal 1000, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 500.	28
4.13	Fel för RFR, tränat på Square Duct med friktions-Reynoldstal 500, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 1000.	28

1

Inledning

1.1 Bakgrund

Inom strömningsmekaniken beskriver Navier-Stokes ekvationer fluiders rörelse. Dessa ekvationer saknar en allmän analytisk lösning och löses därför numeriskt, ofta med hjälp av Direct Numerical Simulation (DNS). Detta blir dock extremt datorkrävande när turbulens förekommer, så för att göra beräkningarna kostnadseffektiva används turbulensmodeller. Dessa uppskattar turbulensen i mindre komplexa termer som underlättar beräkningar. Exempel på relevanta turbulensmodeller är $k - \epsilon$ modellen kombinerad med en icke-linjär Eddy Viscosity modell (NEVM), som bygger på konstanter framtagna via empirisk data. Arbetet fokuserar på att behandla Navier-Stokes ekvationer, speciellt de tidsmedelvärderade Navier-Stokes ekvationerna (RANS), och Reynoldsspänningar.

Verktyg inom maskininlärning (ML) har utvecklats snabbt och uppmärksammats inom beräkningstunga forskningsdiscipliner. Därmed finns ett stort intresse av att undersöka hur dessa kan användas inom turbulensmodellering. I nuläget används empiriskt bestämda konstanter i modeller för turbulens, men bättre resultat kan frambringas genom att ersätta konstanterna med funktioner beroende av olika egenskaper hos flödet. Dessa funktioner kan tas fram med hjälp av ML.

1.2 Syfte

Syftet med arbetet är att undersöka om vedertagna numeriska simuleringsmodeller för icke-linjära turbulensmodeller kan vidareutvecklas med hjälp av neurala nätverk (NN) och Random Forest Regression (RFR). Ambitionen är att identifiera modeller som kan användas till att prediktera Reynoldsspänningar för flera olika typer av flöden och förbättra resultaten med hjälp av ML.

1.3 Problemformulering

Det centrala problemet som avses lösas är att med hjälp av neurala nätverk och Random Forest Regression lösa stängningsproblemet till RANS-ekvationer. Modellerna skall tränas på redan tillgänglig DNS-data. Utgångspunkten är att ersätta konstanter i Craft *et al.s* icke-linjära $k - \epsilon$ modell med funktioner anpassade ge-

nom ML. Detta ska göras för DNS-data som representerar flöden i olika geometrier. Målet är att ta fram modeller som generaliserar till andra geometrier och friktions-Reynoldstal än vad de tränades på.

Modellernas prestanda ska även optimeras och resultat ska jämföras mot standardvärden från litteratur. Generaliseringen till olika scenarion hos modellerna ska även utforskas.

1.4 Avgränsningar

För att genomföra arbetet inom den givna tidsramen krävs det vissa avgränsningar. Dessa begränsningar omfattar olika aspekter såsom metod samt hur resultat värderas.

DNS-datan nyttjad i arbete är inte genererad av kandidatgruppen utan hämtas av externa källor. Denna data är en central del arbetet och resultaten kan variera beroende på kvalitén hos DNS-datan.

Utöver detta innefattar metoden endast två olika ML-modeller: neurala nätverk och RFR, då dessa bedömdes mest relevanta för arbetet. För resultaten tas inte heller beräkningstiden för varje modell i beaktning, då detta inte anses vara relevant för detta arbete.

2

Teori

I denna del av rapporten presenteras relevant teori om strömningsmekanik, neurala nätverk och Random Forest Regression, som ligger till grund för arbetet. Teoriavsnittet utgår från att läsaren har kunskap om grundläggande principer inom strömningsmekanik.

2.1 Strömningsmekanik

Alla typer av strömning beror på ett flödes egenskaper. Dessa egenskaper inkluderar fältets hastighetsförändring, dess tryckgradient, flödets viskositet. Strömningen kan beräknas utifrån Navier-Stokes ekvationer. För att förenkla modelleringen nyttjas tidsmedelvärdering för att få ekvation (2.1)

$$\underbrace{\bar{u} \frac{\partial \bar{u}}{\partial x} + \bar{v} \frac{\partial \bar{u}}{\partial y}}_{\text{Tidsderivata av hastighetsfält}} = \underbrace{-\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x}}_{\text{Tryckgradient}} + \underbrace{\nu \left(\frac{\partial^2 \bar{u}}{\partial x^2} + \frac{\partial^2 \bar{u}}{\partial y^2} \right)}_{\text{Viskösa termer}} - \underbrace{\frac{\partial \overline{u'u'}}{\partial x} - \frac{\partial \overline{u'v'}}{\partial y}}_{\text{Reynolds-spänningar}} \quad (2.1)$$

De tidsmedelvärda Navier-Stokes ekvationerna, även kallad RANS-ekvationerna, inkluderar okända Reynoldsspänningar $\overline{u'_i u'_j}$ som behöver modelleras. Arbetet gruppen ska göra grundrar sig i föregående års kandidatarbete [1] där modellering av Reynoldsspänningar för fullt utvecklad kanalströmning undersöktes. Likt förra årets kandidatarbete kommer detta årets utgå från den icke-linjära Eddy Viscosity modellen för att ta fram de okända Reynoldsspänningarna. Däremot kommer detta året istället för $k-\omega$ modellen nyttja $k-\varepsilon$ modellen samt undersöka fall för gropgeometri och Square Duct utöver kanalströmning och gränsskikt.

2.1.1 Friktions-Reynoldstal

För att ta reda på vad för typ av flöde som behandlas används Reynoldstalet, definierat som

$$Re = \frac{\rho u L}{\mu} = \frac{u L}{\nu}, \quad \nu = \frac{\mu}{\rho}. \quad (2.2)$$

I fallet av väggavgränsade turbulenta flöden kan man använda ett modifierat Reynoldstal, ett så kallat friktions-Reynoldstal. Friktions-Reynoldstalet använder sig av friktionshastigheten, u_τ , istället för den karakteristiska hastigheten u . Definitionen

av friktionshastigheten utgår ifrån definitionen av väggskjuvspänning τ_w i (2.3)

$$\tau_w = \mu \left. \frac{\partial u}{\partial y} \right|_{y=0}. \quad (2.3)$$

Väggskjuvspänningen τ_w kan även uttryckas som i ekvation (2.4) [2]

$$\tau_w = \rho u_\tau^2. \quad (2.4)$$

Användning av (2.3) i (2.4) leder till

$$\tau_w = \rho u_\tau^2 \implies \rho u_\tau^2 = \mu \left. \frac{\partial u}{\partial y} \right|_{y=0} \implies u_\tau = \left(\nu \left. \frac{\partial u}{\partial y} \right|_{y=0} \right)^{1/2}. \quad (2.5)$$

Friktions-Reynoldstalet definieras sedan som

$$Re_\tau = \frac{\rho u_\tau L}{\mu} = \frac{u_\tau L}{\nu}. \quad (2.6)$$

2.1.2 $k - \varepsilon$ modellen

$k - \varepsilon$ modellen är en så kallad två-ekvations modell som nyttjar två skalärer, den kinetiska turbulensen k samt dissipationen ε hos turbulensen för att lösa stängningsproblemet medan konvektion och diffusion termerna antas vara försumbara. De två ekvationer som kvarstår beskrivs enligt (2.7) samt (2.8)

$$\rho \bar{u} k = \left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \right] k + P_k - \rho \frac{k^{3/2}}{\ell}, \quad (2.7)$$

$$\rho \bar{u} \varepsilon = \left[\left(\rho + \frac{\rho_t}{\sigma_\varepsilon} \right) \varepsilon \right] + \frac{\varepsilon}{k} (c_{\varepsilon 1} P_k - c_{\varepsilon 2} \rho \varepsilon). \quad (2.8)$$

Detta i samband med antagandet att konvektion och diffusion antas vara försumbara leder till ekvation (2.9)

$$0 = \frac{\partial}{\partial y} \left[\frac{\mu_t}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial y} \right] + (c_{\varepsilon 1} - c_{\varepsilon 2}) \rho \frac{\varepsilon^2}{k}. \quad (2.9)$$

För att sedan lösa ut Reynoldstensorerna definieras längdskalan ℓ samt den turbulenta kinematiska viskositeten ν_t enligt

$$\ell = \frac{k^{3/2}}{\varepsilon}, \quad (2.10)$$

$$\nu_t = c_\mu k^{1/2} \ell = c_\mu \frac{k^2}{\varepsilon}. \quad (2.11)$$

Detta skapar ett ekvationssystem där k och ε kan lösas ut för att sedan ansätta $\tau = k/\varepsilon$. Detta används sedan i den icke-linjära Eddy Viscosity modellen [3].

2.1.3 Den Icke-linjära Eddy Viscosity modellen (NEVM)

För att modellera de okända Reynoldsspänningar, med hjälp av NEVM, utgår vi från Boussinesq-antagandet i (2.12),

$$\begin{aligned} a_{ij} &= -2\nu_t \frac{s_{ij}}{k}, \\ s_{ij} &= \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right). \end{aligned} \quad (2.12)$$

Den anisotropiska tensorn a_{ij} i (2.12) kan skrivas som [4]

$$\begin{aligned} a_{ij} &= \frac{\overline{u'_i u'_j}}{k} - \frac{2}{3} \delta_{ij}, \\ \delta_{ij} &= \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \end{aligned} \quad (2.13)$$

där δ_{ij} är Kronecker delta. För $k - \varepsilon$ modellen kan den anisotropiska tensorn a_{ij} modelleras med [4], [5]

$$\begin{aligned} a_{ij} &= -2c_\mu \tau s_{ij} \\ &+ c_1 \tau^2 \left(s_{ik} s_{kj} - \frac{1}{3} s_{kl} s_{kl} \delta_{ij} \right) \\ &+ c_2 \tau^2 (\Omega_{ik} s_{kj} - s_{ik} \Omega_{kj}) \\ &+ c_3 \tau^2 \left(\Omega_{ik} \Omega_{jk} - \frac{1}{3} \Omega_{kl} \Omega_{kl} \delta_{ij} \right) \\ &+ c_4 \tau^3 (s_{ik} s_{kl} \Omega_{lj} - \Omega_{il} s_{lk} s_{kj}) \\ &+ c_5 \tau^3 \left(\Omega_{il} \Omega_{lm} s_{mj} + s_{il} \Omega_{lm} \Omega_{mj} - \frac{2}{3} \Omega_{mn} \Omega_{nl} s_{lm} \delta_{ij} \right) \\ &+ c_6 \tau^3 s_{ij} s_{kl} s_{kl} \\ &+ c_7 \tau^3 s_{ij} \Omega_{kl} \Omega_{kl}, \end{aligned} \quad (2.14)$$

där $\tau = k/\varepsilon$ för $k - \varepsilon$ modellen. Skjuvningstensorn s_{ij} definieras enligt (2.12) och rotationstensorn Ω_{ij} definieras som

$$\Omega_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} - \frac{\partial \bar{u}_j}{\partial x_i} \right).$$

Exakta formler existerar för c_i men ofta sätts de till vedertagna konstanta värden. Bland annat kan de ansättas vara [2]

$$c_1 = -0.05, \quad c_2 = 0.11, \quad c_3 = 0.21, \quad c_4 = -0.8, \quad c_5 = 0, \quad c_6 = -0.5, \quad c_7 = 0.5. \quad (2.15)$$

I matrisform kan den anisotropiska tensorn a_{ij} representeras med

$$\mathbf{a} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix}, \quad (2.16)$$

då den är symmetrisk. Utöver detta är spåret i matrisen noll, vilket medför villkoret att $a_{11} + a_{22} + a_{33} = 0$ [4]. Eftersom a_{ij} är symmetrisk kan en ny kolonnmatris konstrueras $\mathbf{a}^* = (a_{11} \ a_{12} \ a_{13} \ a_{22} \ a_{23})^T$ som innehåller de unika värdena i \mathbf{a} . Ofta används konstanta värden för c_i , men för att få bättre precision kan funktionsvärden användas istället. Det som söks är värden på konstanterna c_i , $i = 1, \dots, 7$, men då det finns maximalt 5 element i \mathbf{a}^* går det inte att hitta värden för alla konstanter. Den fullständiga ekvationen som ska lösas är

$$\mathbf{a}^* = \mathbf{A}\mathbf{c} + \mathbf{b}, \quad (2.17)$$

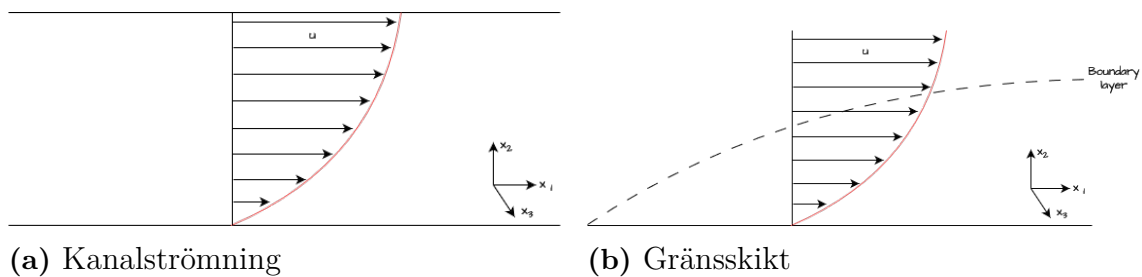
där \mathbf{A} är en kvadratisk matris ($n \times n$), vars element är uttrycken som multipliceras med konstanterna i (2.14), där \mathbf{c} är en kolonnmatris ($n \times 1$) med de ovan nämnda intressanta konstanterna och \mathbf{b} är en kolonnmatris ($n \times 1$) med värde $\mathbf{b} = -2c_\mu \tau s_{ij}$. Notera även att storleken på matriserna bestäms utifrån det relevanta flödet, då n minskar om flöde sker i färre riktningar.

2.1.4 Flöde genom olika geometrier

Förra årets kandidatarbete [1] undersökte fullt utvecklade kanalströmningar och gränsskikt liksom det i figur 2.1. För kanalströmning gäller det att $\bar{u}_2 = \bar{u}_3 = \partial/\partial x_1 = \partial/\partial x_3 = 0$, alltså att det endast är ett flöde i x_1 riktningen och att förändring endast sker i x_2 riktningen. För att beräkna exakta värden i varje punkt för de intressanta konstanterna $c_0 \equiv c_1 + c_3$ och c_2 används följande formler:

$$\begin{aligned} c_0 &= -\frac{6a_{33}}{\tau^2 \left(\frac{\partial \bar{u}_1}{\partial x_2}\right)^2}, \\ c_2 &= \frac{2a_{11} + a_{33}}{\tau^2 \left(\frac{\partial \bar{u}_1}{\partial x_2}\right)^2} \end{aligned} \quad (2.18)$$

Formler ovan gäller även för gränsskikt, liksom det i figur 2.1b [4].

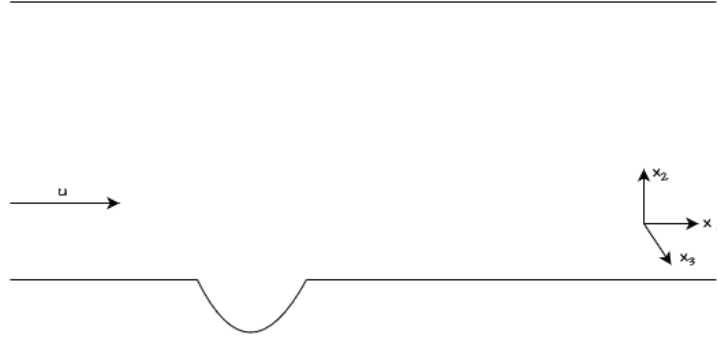


Figur 2.1: Geometrin för två fall av flöde.

Utöver kanalströmning och gränsskikt undersöks även formlerna för gropgeometri, se figur 2.2. För gropgeometri gäller det att $\bar{u}_3 = \partial/\partial x_3 = 0$ [2], varav skjuvnings-tensorn s_{ij} och rotationstensorn Ω_{ij} får följande matrisform

$$\begin{aligned} \mathbf{s} &= \begin{pmatrix} \frac{\partial \bar{u}_1}{\partial x_1} & \frac{1}{2} \left(\frac{\partial \bar{u}_1}{\partial x_2} + \frac{\partial \bar{u}_2}{\partial x_1} \right) & 0 \\ \frac{1}{2} \left(\frac{\partial \bar{u}_1}{\partial x_2} + \frac{\partial \bar{u}_2}{\partial x_1} \right) & \frac{\partial \bar{u}_2}{\partial x_2} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \\ \mathbf{\Omega} &= \begin{pmatrix} 0 & \frac{1}{2} \left(\frac{\partial \bar{u}_1}{\partial x_2} - \frac{\partial \bar{u}_2}{\partial x_1} \right) & 0 \\ -\frac{1}{2} \left(\frac{\partial \bar{u}_1}{\partial x_2} - \frac{\partial \bar{u}_2}{\partial x_1} \right) & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \end{aligned} \quad (2.19)$$

där följande samband $s_{12} = s_{21}$ och $\Omega_{12} = -\Omega_{21}$ kan uppmärksammas.



Figur 2.2: Geometrin för flöde med en grop.

Från dessa härleds följande uttryck för den anisotropiska tensorn a_{ij}

$$\begin{aligned}
 a_{11} &= -2c_\mu s_{11}\tau + \frac{1}{3}c_1\tau^2(2s_{11}s_{11} + s_{12}s_{12} - s_{22}s_{22}) + 2c_2\tau^2 s_{12}\Omega_{12} \\
 &\quad + \frac{1}{3}c_3\tau^2\Omega_{12}\Omega_{12} - 2c_4\tau^2 s_{12}\Omega_{12}(s_{11} + s_{22}) \\
 &\quad + c_5\tau^3\Omega_{12}\Omega_{12}\left(-\frac{4}{3}s_{11} + \frac{2}{3}s_{22}\right) \\
 &\quad + c_6\tau^3 s_{11}(s_{11}s_{11} + 2s_{12}s_{12} + s_{22}s_{22}) + 2c_7\tau^3 s_{11}\Omega_{12}\Omega_{12}, \\
 a_{12} &= -2c_\mu s_{12}\tau + c_1\tau^2 s_{12}(s_{11} + s_{22}) + c_2\tau^2\Omega_{12}(-s_{11} + s_{22}), \\
 &\quad + c_4\tau^2\Omega_{12}(s_{11}s_{11} - s_{22}s_{22}) \\
 &\quad - 2c_5\tau^3 s_{12}\Omega_{12}\Omega_{12} \\
 &\quad + c_6\tau^3 s_{12}(s_{11}s_{11} + 2s_{12}s_{12} + s_{22}s_{22}) + 2c_7\tau^3 s_{12}\Omega_{12}\Omega_{12}, \\
 a_{22} &= -2c_\mu s_{22}\tau + \frac{1}{3}c_1\tau^2(-s_{11}s_{11} + s_{12}s_{12} + 2s_{22}s_{22}) - 2c_2\tau^2 s_{12}\Omega_{12} \\
 &\quad + \frac{1}{3}c_3\tau^2\Omega_{12}\Omega_{12} + 2c_4\tau^2 s_{12}\Omega_{12}(s_{11} + s_{22}) \\
 &\quad + \frac{2}{3}c_5\tau^3\Omega_{12}\Omega_{12}(s_{11} - 2s_{22}) \\
 &\quad + c_6\tau^3 s_{22}(s_{11}s_{11} + 2s_{12}s_{12} + s_{22}s_{22}) + 2c_7s_{22}\tau^3\Omega_{12}\Omega_{12}, \\
 a_{33} &= -\frac{1}{3}c_1\tau^2(s_{11}s_{11} + 2s_{12}s_{12} + s_{22}s_{22}) - \frac{2}{3}c_3\tau^2\Omega_{12}\Omega_{12} \\
 &\quad + \frac{2}{3}c_5\tau^3\Omega_{12}\Omega_{12}(s_{11} + s_{22}).
 \end{aligned} \tag{2.20}$$

Till skillnad från kanalströmning och gränsskikt är det svårt att hitta exakta formler

för att beräkna de intressanta konstanterna. Utöver detta är även systemet underbestämt då endast tre av fyra ekvationer i (2.20) kan användas. Detta på grund av villkoret att spåret i \mathbf{a} är noll.

I och med underbestämtheten i ekvationerna för a_{ij} , för gropgeometri, begränsas beräkningarna till att endast titta på konstanterna c_1 , c_2 och c_3 . Resterande konstanter sätts till 0, $c_4 = c_5 = c_6 = c_7 = 0$. Därmed minimeras (2.20) till

$$a_{11} = -2c_\mu s_{11}\tau + \frac{1}{3}c_1\tau^2(2s_{11}s_{11} + s_{12}s_{12} - s_{22}s_{22}) + 2c_2\tau^2 s_{12}\Omega_{12} + \frac{1}{3}c_3\tau^2\Omega_{12}\Omega_{12},$$

$$a_{12} = -2c_\mu s_{12}\tau + c_1\tau^2 s_{12}(s_{11} + s_{22}) + c_2\tau^2\Omega_{12}(-s_{11} + s_{22}),$$

$$a_{33} = -\frac{1}{3}c_1\tau^2(s_{11}s_{11} + 2s_{12}s_{12} + s_{22}s_{22}) - \frac{2}{3}c_3\tau^2\Omega_{12}\Omega_{12}.$$

Från (2.21) härleds explicita uttryck för okända konstanter c_1 , c_2 , c_3 , se Appendix A.

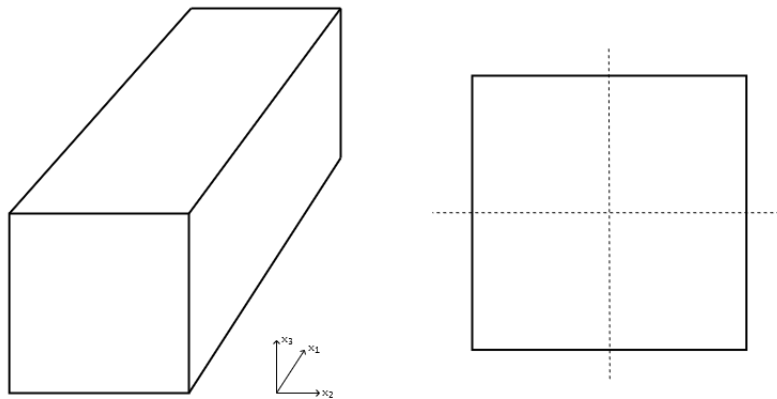
2.1.5 Flöde genom Square Duct

Den sista geometriska formen som underöks är det fyrkantiga röret, ofta benämnt Square Duct. Flödet i Square Duct sker i tre dimensioner där röret i sig kan delas in i 4 symmetriska områden, se figur 2.3. För flöde genom Square Duct gäller det endast att hastighetsförändring i x_1 riktning är noll, $\partial/\partial x_1 = 0$, då flödet är fullt utvecklad. Flödet sker alltså i alla riktningar och förändras i x_2 och x_3 riktning. Tensorerna s_{ij} och Ω_{ij} får matrisformerna

$$\mathbf{s} = \begin{pmatrix} 0 & \frac{1}{2}\frac{\partial\bar{u}_1}{\partial x_2} & \frac{1}{2}\frac{\partial\bar{u}_1}{\partial x_3} \\ \frac{1}{2}\frac{\partial\bar{u}_1}{\partial x_2} & \frac{\partial\bar{u}_2}{\partial x_2} & \frac{1}{2}\left(\frac{\partial\bar{u}_2}{\partial x_3} + \frac{\partial\bar{u}_3}{\partial x_2}\right) \\ \frac{1}{2}\frac{\partial\bar{u}_1}{\partial x_3} & \frac{1}{2}\left(\frac{\partial\bar{u}_2}{\partial x_3} + \frac{\partial\bar{u}_3}{\partial x_2}\right) & \frac{\partial\bar{u}_3}{\partial x_3} \end{pmatrix},$$

$$\mathbf{\Omega} = \begin{pmatrix} 0 & \frac{1}{2}\frac{\partial\bar{u}_1}{\partial x_2} & \frac{1}{2}\frac{\partial\bar{u}_1}{\partial x_3} \\ -\frac{1}{2}\frac{\partial\bar{u}_1}{\partial x_2} & 0 & \frac{1}{2}\left(\frac{\partial\bar{u}_2}{\partial x_3} - \frac{\partial\bar{u}_3}{\partial x_2}\right) \\ -\frac{1}{2}\frac{\partial\bar{u}_1}{\partial x_3} & -\frac{1}{2}\left(\frac{\partial\bar{u}_2}{\partial x_3} - \frac{\partial\bar{u}_3}{\partial x_2}\right) & 0 \end{pmatrix}.$$

Som kan ses innehåller matriserna väldigt få nollor vilket gör att få termer försvinner vid beräkningen av den anisotropiska tensor a_{ij} med hjälp av ekvation (2.14). I Square Duct fallet kan alla 5 unika element nyttjas i \mathbf{a}^* . På grund av att det är svårt att ta fram rimliga analytiska uttryck för de intressanta c_i konstanterna, tas värden på konstanterna numeriskt istället för analytiskt.



Figur 2.3: Geometrin för ett fyrkantigt rör, även kallat Square Duct.

2.2 Maskininlärning

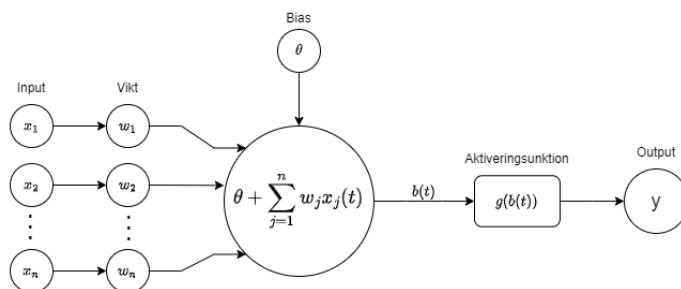
Regressionsproblemet ska lösas genom att implementera ML på två olika sätt: neurala nätverk och Random Forest Regression, den senare i en form som kallas Extreme Gradient Boosting (XGB).

2.2.1 Neurala nätverk

Neurala nätverk är en form av ML som inspirerats av det koncept som ligger till grund för hur hjärnan kan lagra information. Hjärnan är uppbyggd av sammankopplade neuroner som tar emot och skickar vidare signaler. På ett liknande sätt använder neurala nätverk sina noder för att upptäcka mönster eller ta intelligenta beslut. Nätverket tar in data och summerar samt modifierar den på lämpligt sätt för att lösa det givna problemet [6] [7].

Varje nod, b_i , i nätverket tar in och summerar alla indata, x_j , multiplicerat med deras korresponderande vikt, w_{ij} och adderar en bias, θ_i . Sedan appliceras aktiveringsfunktionen g på resultatet och värdet som fås är utdatan för den givna noden b_i . Detta kan ses på ekvationsform i ekvation (2.23) och visuellt i figur 2.4.

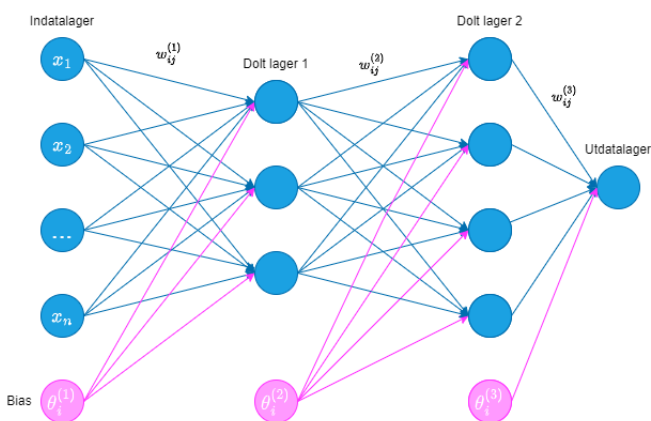
$$\begin{aligned} b_i(t) &= \theta_i + \sum_{j=1}^N w_{ij}x_j(t) \\ x_i(t+1) &= g(b_i(t)) \end{aligned} \tag{2.23}$$



Figur 2.4: Figuren visar processen för summation mellan indata och utdata med hjälp av bias och aktiveringsfunktion för en given nod.

Aktiveringsfunktioner används för att kunna styra över hur och när noderna skall aktiveras. Detta kan liknas med hur neuroner i hjärnan endast stimuleras av en tillräckligt hög signal, på liknade sätt aktiveras endast noden i neurala nätverket ifall indatan är tillräckligt stor [7]. Biasen används för att nätverket ska kunna anpassas oberoende av indata, ty bias är fristående från datan. Till stor del förskjuter biasen aktiveringsfunktionen och detta kan ses som att den flyttar aktiveringsgränser. Både aktiveringsfunktionen och biasen kombineras så att modellen kan finjusteras efter vad som önskas [8]. Två vanliga aktiveringsfunktioner är *ReLU* och identitetsfunktionen, som definieras $ReLU(x) = \max(0, x)$ och $Identitet(x) = x$ [9]. Både aktiveringsfunktionen och bias reducerar det linjära sambandet mellan in- och utdata, alltså ger de nätverket verktyg för att upptäcka mer invecklade samband [8].

Ett neuralt nätverk består av flera lager noder: ett indatalager, ett eller flera dolda lager och ett utdatalager. Alla lager kan ha olika antal noder. Processen initieras då indatan passerar genom ett första lager av noder och transformeras till utdata, denna utdata blir då indata till nästa lager av noder. Denna process upprepas tills datan passerat alla lager noder, då har ett slutgiltigt resultat producerats. Genom att använda flera lager kan nätverket upptäcka mer komplexa mönster [9]. Ett exempel på hur ett nätverk med två dolda lager kan se ut ses i figur 2.5.



Figur 2.5: Exempel på hur lagerstrukturen hos ett neuralt nätverk kan se ut.

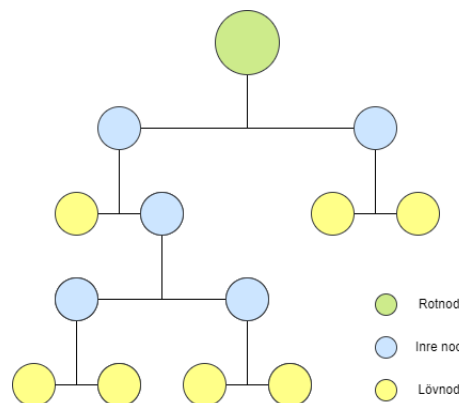
Grundtanken med neurala nätverk är att de ska optimeras med hjälp av tränings-

data för att sedan kunna användas för att göra förutsägelser. När modellen tränas passerar indatan alla lagar av noder och transformerar enligt ekvation 2.23 till utdata. Utdata jämförs sedan med det korrekta svaret givet från träningsdatan. Efter jämförelsen justeras vikterna $w_{i,j}$ hos noderna för att optimera resultatet så att nätverket ger mer precisa förutsägelser [8].

Hyperparametrarna epoch, batchstorlek och inlärningshastighet används för att justera träningshastigheten. Epoch beskriver hur många gånger nätverket ska träna på det givna datasetet och batchstorleken bestämmer hur många datapunkter som nätverket ska tränas på innan det uppdaterar och justera de interna parametrarna, exempelvis vikterna [10]. Inlärningshastigheten beskriver hur stor justeringen av de interna parametrarna skall vara för varje uppdatering [8].

2.2.2 Random Forest Regression

Beslutsträd är en form av övervakad ML för klassificering och regression. Ett beslutsträd består av en rot och noder samt kanter där trädet grenar sig. Sista noden på varje gren kallas för en lövnod. Varje nod representerar ett särskilt attribut och kanterna representerar ett värdeintervall [11]. Trädstrukturen kan ses i figur 2.6.



Figur 2.6: Struktur hos ett godtyckligt beslutsträd.

Uppdelningen av datan sker sedan baserat på attributen. Träden kan då genom att följa en serie beslut eller regler nå fram till en slutsats eller förutsägelse [12]. Varje nod i ett beslutsträd representerar en testning av ett attribut, och varje gren representerar resultatet av detta test. Genom att gå igenom trädet från roten till en lövnod kan beslutssekvensen följas och då kan slutsatser dras baserade på testresultaten [11].

Varje träd får ett unikt set data genom att använda bootstrapping på originaldatan, vilket innebär att ny data genereras baserat på statistiska egenskaper hos datasetet [13]. Eftersom varje träd baseras på olika data blir de oberoende av varandra, så beslutsvariationen ökar.

Ett slumpmässigt beslutsträd definieras av att det väljs från en uppsättning av träd, där träden har K slumpmässiga attribut vid varje nod. Detta sker på så sätt att varje träd har samma sannolikhet att väljas, alltså är fördelningen likformlig [11].

RFR är en form av övervakad ML som kombinerar förutsägelser från flera slumpmässiga beslutsträd för att ge en mer kvalificerad gissning än vad som kan åstadkommas med enskilda beslutsträd, då dessa är benägna till överanpassning. Detta är ett exempel av ensemble learning, vilket innebär att det slutgiltiga resultatet fås av medelvärdet av förutsägelser för varje enskilt slumpmässiga beslutsträd [12].

2.2.2.1 Extreme Gradient Boosting

En variant av RFR, kallad Extreme Gradient Boost, minimerar fel genom att använda en algoritm som lär sig av föregående träd. XGB använder sig av ett antal hyperparametrar som den optimerar med hjälp av GridSearch Cross Validation (GridSearchCV). Användaren väljer vilka parametrar som är relevanta och ger sedan flera inputvärden på dessa. GridSearchCV testar därefter utfallet av träningsdatan utifrån alla kombinationer av dom valda värdena. Avvikelsen från måldatan beräknas slutligen för varje utfall, baserat på en i förväg vald måttstock, varefter det bästa utfallet selekteras till modellen. Sedan undersöks vilka inputvärden som givit detta utfall, och om dessa antagit högsta respektive lägsta värde i utgångsintervallet, höjs respektive sänks gränsen för intervallet och proceduren upprepas. Detta itereras tills ett läge där prestandan konvergerar och på så sätt minimeras risken för överanpassning [14].

3

Metod

I detta avsnitt förklaras tillvägagångssätt och procedurer som använts för att fullfölja arbetet, genom hela processen. Detta inkluderar databehandling och konstruering av ML-modellerna som används för turbulensmodelleringen. All programmering och utveckling av maskininlärningsmodeller kommer att utföras i Python och komplett kod kan hittas på den GitLab som finns i Appendix B.

3.1 Databehandling och val av in- och utdata för modeller

Datan som använts är DNS-simuleringar för kanalströmning, gränsskikt och gropgeometri, hämtade från [15], och Square Duct hämtad från [16]. Värdena på c_i som predikterades ingår däremot inte i någon av datan och behövde därför lösas ut. I fallen kanalströmning och gränsskikt kunde formlerna (2.18) användas och i fallet med gropgeometri togs formlerna (A.2) från Appendix A fram symboliskt. I Square Duct löstes \mathbf{c} ut numeriskt ur ekvation (2.17) för varje punkt. All data delades även upp i 80% träningsdata och 20% testdata innan den användes.

Indatan till nätverket varierar beroende på vilken geometri som modelleras, men geometrin användes aldrig som indata. Detta beror på att modellerna ska kunna generalisera mellan olika geometrier. Istället används olika hastighetsgradienter som indata, då dessa är gemensamma egenskaper för all form av strömning och därför inte påverkar generaliseringen.

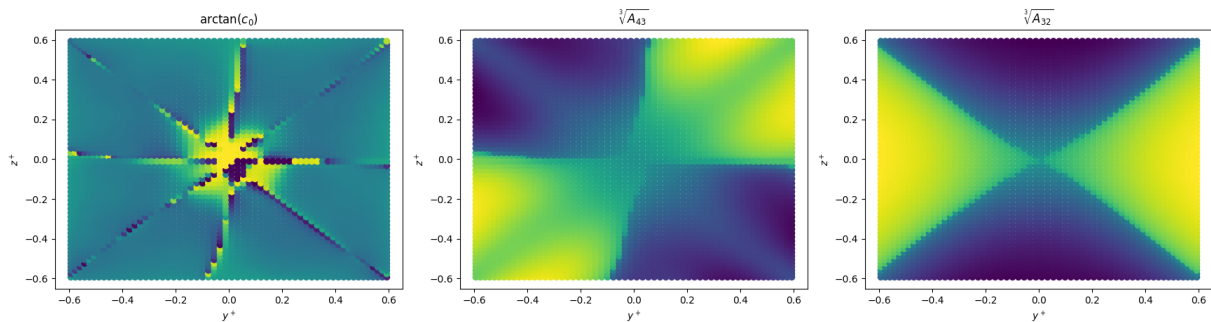
För kanalströmning och gränsskikt har

$$T^2 \left(\frac{\partial U}{\partial y} \right)^2, \quad \left(T \frac{\partial U}{\partial y} \right)^{-1}, \quad T = \frac{k}{\varepsilon},$$

tidigare använts som indata till modeller för att prediktera c_0 samt c_2 [4]. Samma variabler användes för modellerna i detta arbete. Gradienterna i kvadrat $(\partial U / \partial y)^2$ och inverserna $(\partial U / \partial y)^{-1}$ användes eftersom man kan se i ekvation (2.18) att c_i beror på just dessa. Dessa görs dimensionslösa genom en lokal skalning med T vilket underlättar generalisering.

För flöde genom gropgeometri och Square Duct uppvisar c_i mer komplext geometriskt beteende, och även diskontinuiteter och exponentiell ökning. Dessa beteenden

gör anpassning av modeller betydligt svårare, och det kan vara utmanande att ens få träningsprocessen att konvergera. Uppgiften underlättas om det finns indata som delar c_i 's komplexa beteende, exempelvis geometriska egenskaper. För flöde genom gropgeometri och Square Duct visade sig elementen A_{ij} ur ekvation (2.17) vara lämpliga. Exempel på delade geometriska drag mellan c_i och A_{ij} illustreras i figur 3.1.



Figur 3.1: Heatmap plots av $\arctan(c_0)$, $\sqrt[3]{A_{43}}$ och $\sqrt[3]{A_{32}}$ i Square Duct med friktions-Reynoldstal 150. Arctangens och kubrötter har tagits för att göra de delade geometriska mönstren synliga.

Exponentiellt beteende utgör dock fortfarande ett problem vid anpassning av modeller och därför användes $\arctan(A_{ij})$ som indata och vi predikterar $\arctan(c_i)$ istället för att prediktera c_i direkt.

I alla flöden orsakar även väggarna problem eftersom hastighetsgradienterna blir extremt stora nära dem, vilket svårgör konvergens vid träning av modeller. I och med detta begränsades datan båda vid träning och testning i varje fall till ett visst avstånd ifrån väggarna. Detta gjordes dock inte för flöde över grop då detta skar bort den mest intressanta delen av datan.

För att möjliggöra jämförelse implementerades också modellerna med konstanta c_i . För kanalströmning och gränsskikt användes konstanterna ur (2.15). För Square Duct användes istället

$$c_1 = -0.1, c_2 = 0.1, c_3 = 0.26, c_4 = -10c_\mu^2, c_5 = 0,$$

tagna från [17] då dessa visade sig prestera betydligt bättre.

3.2 Neurala nätverk

Det neurala nätverket implementeras med hjälp av Python-paketet PyTorch. Nätverket som användes för att modellera kanalströmning och gränsskikt bestod av två dolda lager med 50 noder vardera, samt indata- och utdatalager. Nätverket tränades med 10 000 epoker, inlärningshastighet 0.5 och hade batchstorlek 5. För flöde med gropgeometri användes istället två dolda lager med 10 noder, 1 000 epoker och batchstorlek 10 och inlärningshastighet 0.0001. För Square Duct användes nätverk

med 2 dolda lager av bredd 11 och med batch normalization. Nätverken tränades med 1000 epoker, batchsize 5 och learning rate 0.00007. Aktiveringsfunktionen som användes för samtliga modellers dolda lager är $ReLU = \max(0, b_i)$ och identitetsfunktionen till utdatalagret.

3.3 Random Forest Regression

Modellen för RFR utvecklades med XGB genom Python-paketerna `sklearn` och `xgboost`. Modellen använde `XGBRegressor` för att skapa en prediktiv regressor baserad på datan från databehandlingen. I denna applikation valdes att specificera följande hyperparametrar: antalet beslutsträd, beslutsträdens maximala djup, inlärningshastighet samt hur träningsdatan valts ut. Utdatan som returnerades var de koefficienter c_i som existerar för den givna geometrin.

XGB anpassades till varje flöde utifrån resultatet av `GridSearchCV`. Därefter rangordnades modellerna med avseende på deras genomsnittliga procentuella avvikelse från testdatan (MAPE). XGB utfördes på samma vis för alla dataset oavsett flöde. Detta beror på att antalet koefficienter som modelleras är irrelevant för genomförandet, ty XGB stödjer multipel linjär regression. Utdatan kommer alltid ha samma dimension som indatan, vilket enkelt tillåter att modellera olika geometrier av strömning.

3.4 Felberäkning

Prestandan av modellerna med ML jämfördes gentemot modeller med konstant c_i och de exakta simulerade värdena, se avsnitt 2.1. Detta användes sedan för att utvärdera och optimera modellerna.

Undersökning av hur väl modellerna generaliserade mellan de olika geometrierna samt mellan olika friktions-Reynoldstal genomfördes. Detta genom att träna modellerna på träningsdata från en geometri/friktions-Reynoldstal och sedan testa modellen på testdata från en annan geometri/friktions-Reynoldstal. På så sätt kunde modellerna utvärderas i hur väl de representerar en allmän lösning för de olika flödena.

Alla jämförelser av modellerna gjordes med en felanalys där felet uppskattades genom olika metoder. En metod som användes var Mean Average Percentage Error (MAPE) som definieras enligt

$$MAPE = \frac{\sum_{i=1}^N |x_i^{DNS} - x_i^{Pred}|}{\sum_{i=1}^N |x_i^{DNS}|} \quad (3.1)$$

där x_i^{DNS} är en given Reynoldsspänning från DNS-datan och x_i^{Pred} är predikationen för samma Reynoldsspänning. En annan metod som användes var Root Mean Square Percentage Error (RMSPE) som definieras enligt

$$RMSP E = \sqrt{\frac{\sum_{i=1}^N |x_i^{DNS} - x_i^{Pred}|^2}{\sum_{i=1}^N |x_i^{DNS}|^2}}. \quad (3.2)$$

För Square Duct användes även en tredje metod kallad Maximum Absolute Error (MaAE) som definieras enligt

$$MaAE = \max_{i \in N} |x_i^{DNS} - x_i^{Pred}|. \quad (3.3)$$

4

Resultat

Resultatavsnittet ämnar att presentera de fynd och resultat som erhållits från det genomförda arbetet. Resultaten utgör grunden för uppsatsens efterföljande avsnitt bestående av analys, diskussion och tolkning. För samtliga flöden kommer precision mätas för prediktering på egen testdata mellan c_i framtagna med ML och med vedertagna konstanter från litteratur. Utöver detta kommer generalisering mellan geometrier respektive friktions-Reynoldstal undersökas för kanalströmning och gränsskikt respektive Squire Duct.

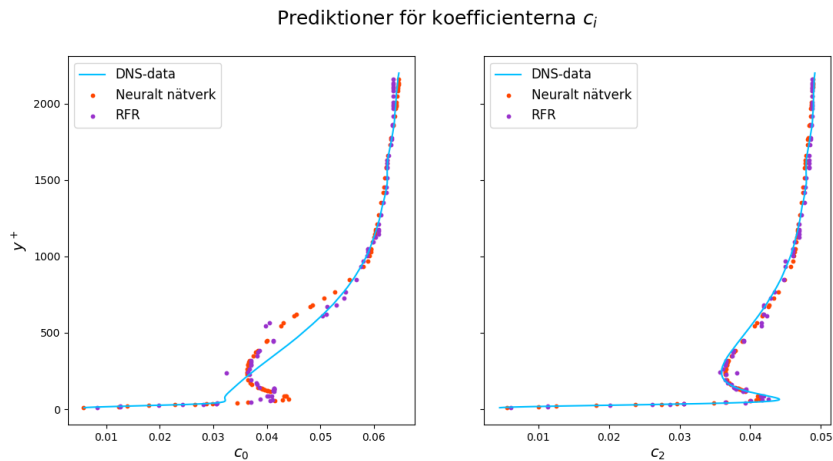
4.1 Kanalströmning

Nedan följer resultat från neurala nätverket och RFR för kanalströmning med friktions-Reynoldstal 5200. Modellerna har tränats och testats på samma data (geometri och friktions-Reynoldstal). I tabell 4.1 ses MAPE och RMSPE för c_0 , c_2 , $\overline{u'u'^+}$, $\overline{v'v'^+}$ och $\overline{w'w'^+}$. Som kan ses är skillnaden i fel mellan metoderna jämförelsevis liten, förutom för c_0 och $\overline{v'v'^+}$ där RFR presterar sämre.

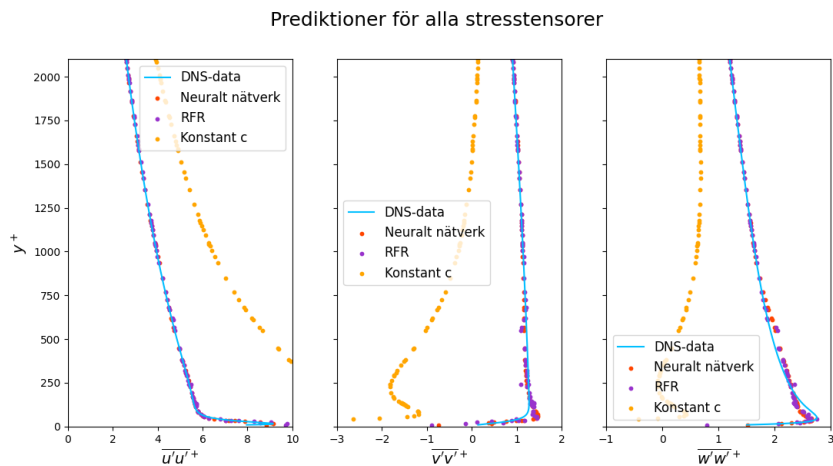
Tabell 4.1: Fel för neuralt nätverk och RFR, tränade och testade på kanalströmning med friktions-Reynoldstal 5200.

	MAPE NN	RMSPE NN	MAPE RFR	RMSPE RFR	MAPE konstant c_i	RMSPE konstant c_i
c_0	5.14%	7.92%	4.99%	7.54%	-	-
c_2	1.42%	2.40%	1.37%	2.32%	-	-
$\overline{u'u'^+}$	0.87%	2.21%	1.43%	4.54%	134.41%	306.28%
$\overline{v'v'^+}$	5.81%	11.95%	6.52%	14.72%	332.93%	817.18%
$\overline{w'w'^+}$	2.33%	3.62%	2.83%	5.31%	137.05%	301.93%

I figur 4.1 ses hur väl neuralt nätverk och RFR predikterar c_0 och c_2 . Det framgår att c_0 är svåröverskattat för båda metoderna. Figur 4.2 illustrerar hur stresstensorerna ser ut för modellerna jämfört med vad som fås från DNS-data samt genom att använda konstanta värden för c_i . Som kan ses presterar ML avsevärt bättre jämfört med de vanliga konstanterna. Notera dock att vissa värden för $\overline{v'v'^+}$ är negativa, vilket är ofysikaliskt.



Figur 4.1: Prediktioner för c_i för det neurala nätverket och RFR, jämfört med det sanna värdet från DNS-data.



Figur 4.2: Reynoldsspänningar för kanalströmning predikterade med neurala nätverket, RFR och konstanta c_i , jämfört med de samma värdena från DNS-datan.

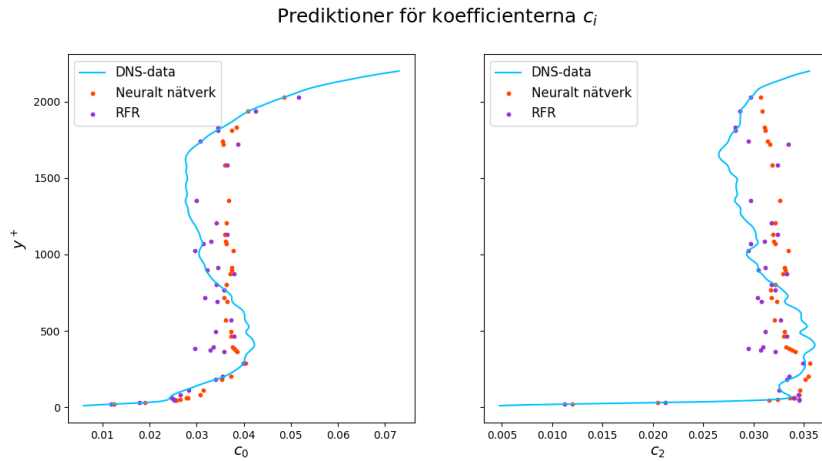
4.2 Gränsskikt

Nedan följer resultat för gränsskikt med friktions-Reynoldstal 2500 för de två metoderna. Modellerna har tränats på samma data som de testas för. I tabell 4.2 ses RMSPE och MAPE för c_0 , c_2 , $\overline{u'u'^+}$, $\overline{v'v'^+}$ och $\overline{w'w'^+}$. För MAPE gäller generellt att RFR presterar bättre än det neurala nätverket, men tvärtom för RMSPE. Att använda konstanter ger som innan stora feltermar.

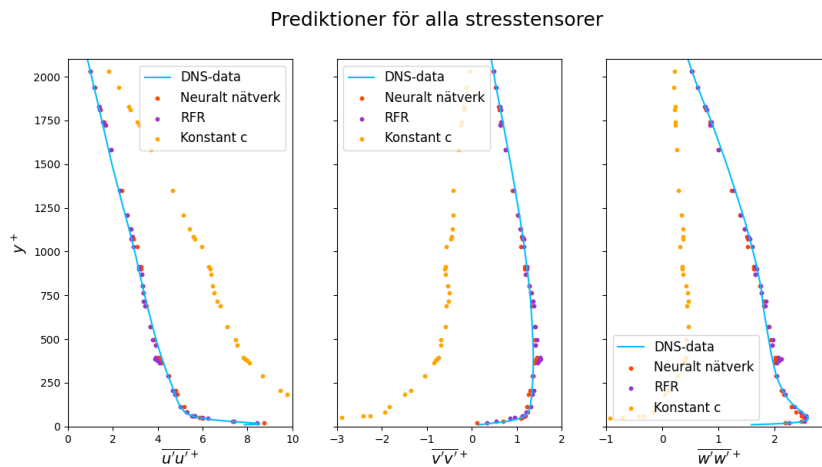
Tabell 4.2: Fel för neuralt nätverk och RFR, tränade och testade på gränsskikt med friktions-Reynoldstal 2500.

	MAPE NN	RMSPE NN	MAPE RFR	RMSPE RFR	MAPE konstant c_i	RMSPE konstant c_i
c_0	10.75%	12.46%	8.96%	12.78%	-	-
c_2	6.16%	7.20%	5.86%	8.11%	-	-
$\overline{u'u'^+}$	2.41%	2.81%	2.39%	3.07%	135.49%	215.31%
$\overline{v'v'^+}$	4.55%	6.84%	5.33%	7.59%	261.25%	451.18%
$\overline{w'w'^+}$	2.41%	2.68%	1.87%	2.64%	120.38%	192.81%

Figur 4.3 och figur 4.4 visar hur väl de olika metoderna presterar när de predikterar c_0 och c_2 , respektive $\overline{u'u'^+}$, $\overline{v'v'^+}$ och $\overline{w'w'^+}$. Även här presterar de konstanta c_i sämre än ML. Värt att notera att ingen av metoderna får särskilt exakta värden för konstanterna, men att detta inte verkar påverka resultaten för stresstensorerna.



Figur 4.3: Prediktioner för c_i för det neurala nätverket och RFR, jämfört med det sanna värdet från DNS-data.



Figur 4.4: Reynoldsspänningar för gränsskikt predikterade med neurala nätverket, RFR och konstanta c_i , jämfört med de sanna värdena från DNS-datan.

4.3 Generalisering mellan kanalströmning & gränsskikt

Nedan följer resultatet för generalisering mellan geometrierna kanalströmning och gränsskikt. Modellerna tränades på strömningsfallet kanalströmning eller gränsskikt och testades sedan på det motsatta. Tabell 4.3 visar att generaliseringen för modellen som tränats på kanalströmning och sedan testats på gränsskikt ger större procentuella fel än när modellerna testades på egen testdata. Speciellt konstanterna c_0 och c_2 samt Reynoldsspänningen $\overline{v'v'^+}$ modellerades dåligt av både neurala nätverket och RFR. Felen är dock fortfarande betydligt bättre än att använda konstant c_i .

Tabell 4.3: Fel för neuralt nätverk och RFR, tränade på turbulens av typ kanalströmning med friktions-Reynoldstal 5200 och testade på gränsskikt med friktions-Reynoldstal 2500.

	MAPE NN	RMSPE NN	MAPE RFR	RMSPE RFR	MAPE konstant c_i	RMSPE konstant c_i
c_0	18.87%	21.66%	18.47%	22.19%	-	-
c_2	20.41%	24.11%	23.54%	27.13%	-	-
$\overline{u'u'^+}$	4.69%	5.03%	6.18%	6.10%	135.49%	215.31%
$\overline{v'v'^+}$	14.26%	14.71%	17.03%	17.42%	261.25%	451.18%
$\overline{w'w'^+}$	3.93%	4.44%	3.73%	4.31%	120.38%	192.81%

Även generalisering åt andra hållet, då modellen tränats på data från gränsskikt och testas på kanalströmning, ger stora procentuella fel. Som kan ses i tabell 4.4 är felen som störst för c_0 , c_2 och $\overline{v'v'^+}$, men precis som innan fungerar ML-metoderna bättre än att använda konstanta c_0 och c_1 .

Tabell 4.4: Fel för neuralt nätverk och RFR, tränade på turbulens av typen gränsskikt med friktions-Reynoldstal 2500 och testade på kanalströmning med friktions-Reynoldstal 5200.

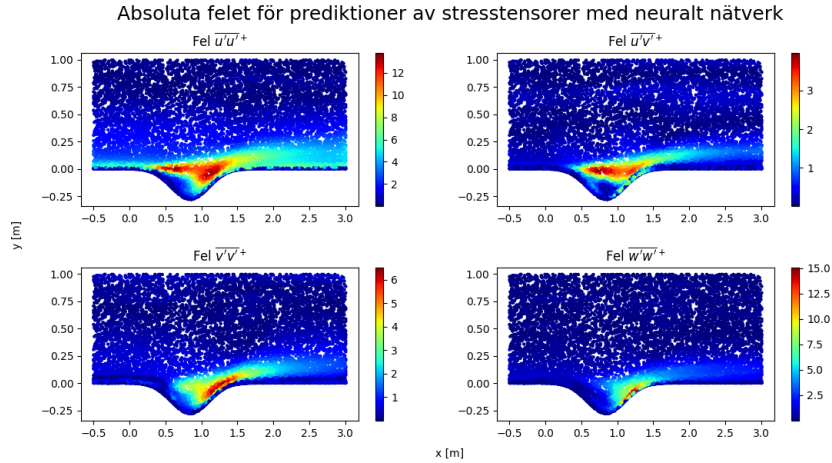
	MAPE NN	RMSPE NN	MAPE RFR	RMSPE RFR	MAPE konstant c_i	RMSPE konstant c_i
c_0	11.23%	12.84%	8.45%	12.74%	-	-
c_2	21.87%	24.87%	24.05%	26.91%	-	-
$\overline{u'u'^+}$	6.91%	8.13%	7.98%	9.26%	134.41%	306.28%
$\overline{v'v'^+}$	29.68%	33.12%	32.25%	36.01%	332.93%	817.18%
$\overline{w'w'^+}$	3.96%	6.03%	3.49%	6.38%	137.05%	301.93%

Notera att felen för stresstensorerna är avsevärt lägre för första generaliseringsfallet (tränat på kanalströmning och testat på gränsskikt). Detta gäller för både metoderna och trots att felen för koefficienterna c_i är relativt lika. Det framgår från båda tabellerna att RFR får högre fel under generaliseringen än vad det neurala nätverket får.

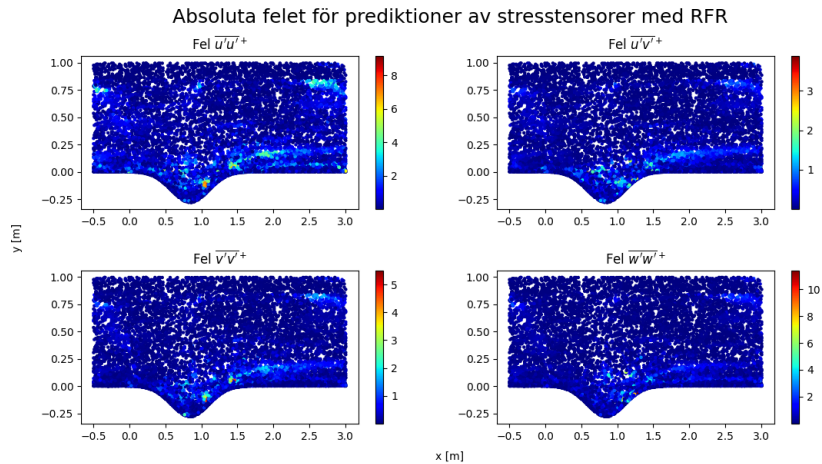
4.4 Gropgeometri

Detta avsnitt redovisar resultaten som fås av turbulensmodellering på flöde med gropgeometri, med friktions-Reynoldstal 550. Som kan ses från figur 4.5 och figur 4.6 har resultaten stora fel jämfört med det sanna DNS-värdet för båda metoderna, men

dock störst för det neurala nätverket. Det absoluta felet i figur 4.5 och figur 4.6 är absolutbeloppet av skillnaden mellan den simulerade spänningen och den framräknade DNS-spänningen i varje punkt av testdatan. I figur 4.5 är felen koncentrerade till där turbulensen förväntas vara, vilket antyder att modellen misslyckas med just turbulensmodelleringen.



Figur 4.5: Det absoluta felet för stresstensorerna $\overline{u'u'^+}$, $\overline{u'v'^+}$, $\overline{v'v'^+}$ och $\overline{w'w'^+}$ för det neurala nätverket, jämfört med det sanna värdet från DNS-data.



Figur 4.6: Det absoluta felet för stresstensorerna $\overline{u'u'^+}$, $\overline{u'v'^+}$, $\overline{v'v'^+}$ och $\overline{w'w'^+}$ för RFR, jämfört med det sanna värdet från DNS-data.

Tabell 4.5 redovisar ytterligare att felen som fås med de båda modellerna är relativt stora, även procentuellt. Detta gäller speciellt för alla c_i . Störst fel fås av RFR för c_1 och c_2 .

Tabell 4.5: Fel för neuralt nätverk och RFR, tränade och testade på turbulens med friktions-Reynoldstal 550 simulerat över gropgeometri.

	RMSPE för NN	MAPE för NN	RMSPE för RFR	MAPE för RFR
c_1	100.0%	100.0%	307.92%	374.84%
c_2	100.0%	100.0%	98.99%	92.33%
c_3	100.0%	100.0%	143.21%	253.36%
$\overline{u'u'^+}$	82.81%	73.5%	22.35%	17.49%
$\overline{u'v'^+}$	76.52%	61.29%	32.7%	26.57%
$\overline{v'v'^+}$	80.66%	70.03%	28.18%	23.64%
$\overline{w'w'^+}$	84.53%	64.91%	26.52%	22.7%

4.5 Square Duct

Detta avsnittet redovisar resultaten som fås av turbulensmodellering för Square Duct med friktions-Reynoldstal 1000 och friktions-Reynoldstal 500, precisionen mäts för prediktion på testdata med samma friktions-Reynoldstal.

4.5.1 Neurala nätverk

Tabell 4.6 och tabell 4.7 presenterar fel i Reynoldsspänningar för modeller tränade och testade på flöde med friktions-Reynoldstal 1000 respektive 500, med hjälp av neurala nätverk.

Tabell 4.6: Fel för neuralt nätverk, tränat på Square Duct med friktions-Reynoldstal 1000, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 1000.

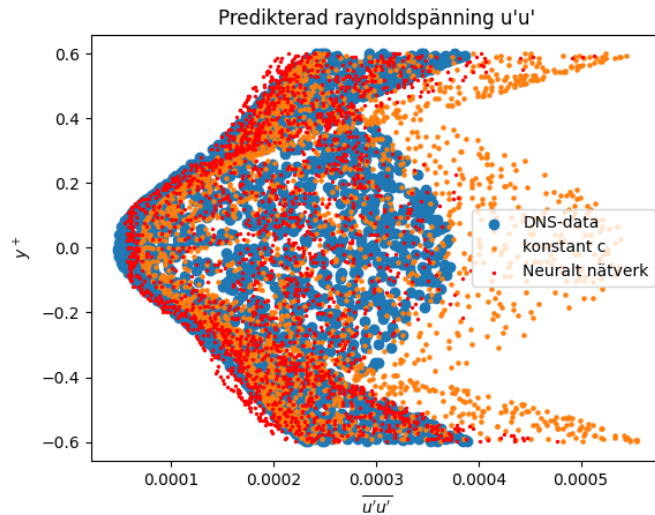
	MAPE NN	RMSPE NN	MaAE NN	MAPE konstant c_i	RMSPE konstant c_i	MaAE konstant c_i
$u'u'$	7.48%	9.98%	$1.44 \cdot 10^{-4}$	22.95%	29.13%	$1.72 \cdot 10^{-4}$
$u'v'$	34.47%	54.72%	$2.40 \cdot 10^{-4}$	25.97%	23.69%	$3.98 \cdot 10^{-5}$
$u'w'$	42.85%	65.52%	$1.85 \cdot 10^{-4}$	28.73%	26.99%	$4.95 \cdot 10^{-5}$
$v'v'$	6.51%	10.04%	$9.89 \cdot 10^{-5}$	18.47%	25.26%	$1.03 \cdot 10^{-4}$
$v'w'$	41.14%	55.21%	$2.14 \cdot 10^{-5}$	82.43%	99.46%	$2.29 \cdot 10^{-5}$

Tabell 4.7: Fel för neuralt nätverk, tränat på Square Duct med friktions-Reynoldstal 500, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 500.

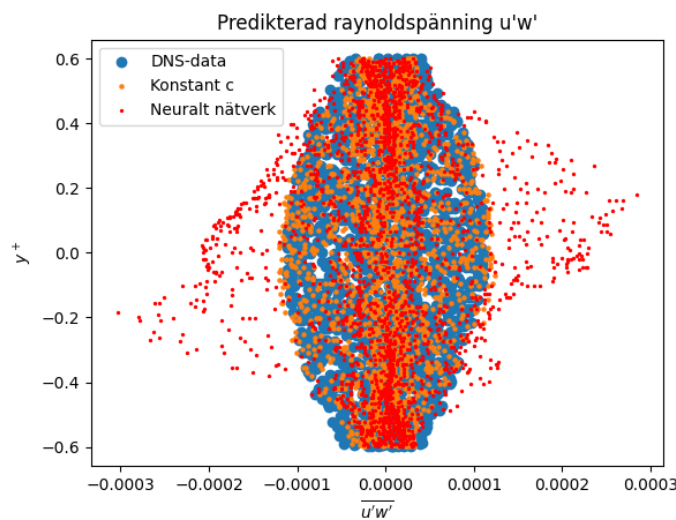
	MAPE NN	RMSPE NN	MaAE NN	MAPE konstant c_i	RMSPE konstant c_i	MaAE konstant c_i
$u'u'$	12.65%	17.89%	$2.0 \cdot 10^{-4}$	24.68%	27.51%	$1.58 \cdot 10^{-4}$
$u'v'$	35.25%	57.7%	$2.67 \cdot 10^{-4}$	34.26%	29.71	$3.73 \cdot 10^{-5}$
$u'w'$	42.15%	77.39%	$3.64 \cdot 10^{-4}$	7.31%	29.99%	$3.33 \cdot 10^{-5}$
$v'v'$	9.61%	15.78%	$1.55 \cdot 10^{-4}$	17.01%	21.43%	$1.03 \cdot 10^{-4}$
$v'w'$	29.19%	36.66%	$2.15 \cdot 10^{-5}$	39.97%	43.53%	$1.23 \cdot 10^{-5}$

Medan de genomsnittliga felen i tabell 4.6 och tabell 4.7 i de flesta fall minskar relativt modellen med konstanta c_i syns samtidigt att de maximala felen i flera fall ökar. Specifikt ökar MaAE för $\overline{u'v'}$ och $\overline{u'w'}$ med nästan en tiopotens i båda tabellerna. Detta tyder på att modellerna medför en viss instabilitet, det vill säga en

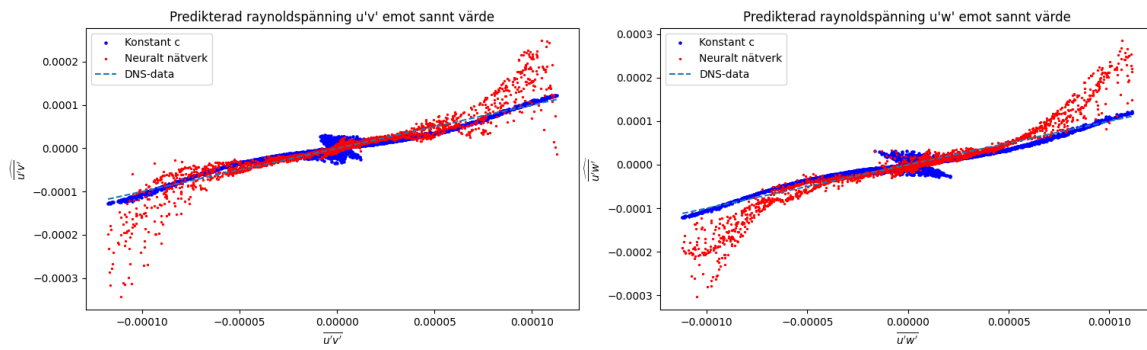
systematisk feltrend för prediktionerna av $\overline{u'v'}$ och $\overline{u'w'}$. I figur 4.8 illustreras ett exempel där instabiliteten framgår tydligt och i figur 4.7 ett där instabilitet inte märks av. Var instabiliteten förekommer kan ses i figur 4.9 som illustrerar de predikterade Reynoldsspänningarna mot de sanna värdena. Instabiliteten förekommer särskilt vid de extrema värdena, men för konstanta c_i är detta inte fallet.



Figur 4.7: Reynoldsspänning $\overline{u'u'}$ i flöde Square Duct med friktions-Reynoldstal 1000 plottad mot y^+ . Figuren visar DNS-data, resultat av modell med konstant c_i och modell med neuralt nätverk tränat på samma friktions-Reynoldstal.



Figur 4.8: Reynoldsspänning $\overline{u'w'}$ i flöde Square Duct med friktions-Reynoldstal 1000 plottad mot y^+ . Figuren visar DNS-data, resultat av modell med konstant c_i och modell med neuralt nätverk tränat på samma friktions-Reynoldstal.



Figur 4.9: Predikterade Reynoldsspänningar $\widehat{u'v'}$ och $\widehat{u'w'}$ i flöde Square Duct med friktions-Reynoldstal 1000 plottad mot de sanna värdena $\overline{u'v'}$ och $\overline{u'w'}$. Graferna innehåller resultat för konstanta c_i och för modell med neuralt nätverk tränat på samma friktions-Reynoldstal.

4.5.2 Random Forest Regression

Tabell 4.8 och tabell 4.9 visar avvikelser i Reynoldsspänningar för modeller tränade och testade med RFR, enligt samma principer som för neurala nätverk. Även här ses en tydlig förbättring jämfört med konstanta värden för c_i . Man ser i tabell 4.8 och tabell 4.9 att $\overline{v'w'}$ får markanta högre fel jämfört med resterande Reynoldsspänningar, detta både för modellerna och med konstanta c_i . Gällande MaAE är tendensen den samma som för neurala nätverk, nämligen att vissa av de predikterade Reynoldsspänningarna har stora avvikelser vilket antyder på vissa fel hos modellerna.

Tabell 4.8: Fel för RFR, tränat på Square Duct med friktions-Reynoldstal 1000, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 1000.

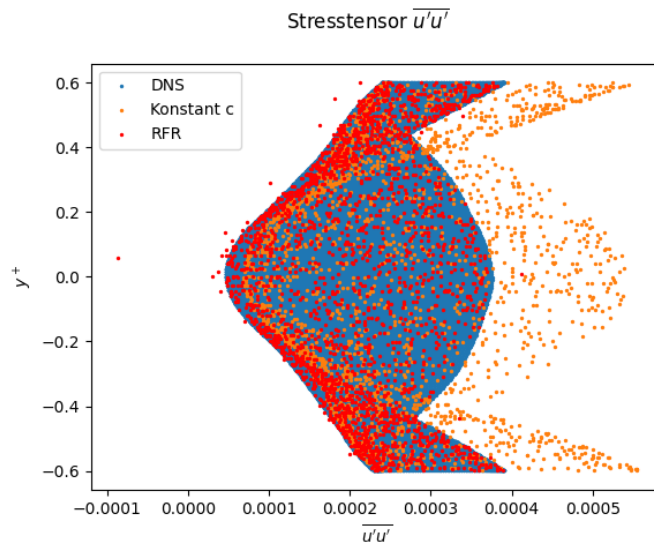
	MAPE RFR	RMSPE RFR	MaAE RFR	MAPE konstant c_i	RMSPE konstant c_i	MaAE konstant c_i
$u'u'$	0.98%	5.72%	$5.9 \cdot 10^{-4}$	22.95%	29.13%	$1.7 \cdot 10^{-4}$
$u'v'$	1.55%	6.37%	$1.4 \cdot 10^{-4}$	25.97%	23.69%	$3.98 \cdot 10^{-5}$
$u'w'$	1.42%	5.42%	$1.6 \cdot 10^{-4}$	28.73%	26.99%	$4.5 \cdot 10^{-5}$
$v'v'$	1.65%	12.03%	$6.2 \cdot 10^{-4}$	18.47%	25.26%	$1.0 \cdot 10^{-4}$
$v'w'$	13.35%	79.0%	$3.5 \cdot 10^{-4}$	82.43%	99.46%	$2.29 \cdot 10^{-5}$

Tabell 4.9: Fel för RFR, tränat på Square Duct med friktions-Reynoldstal 500, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 500.

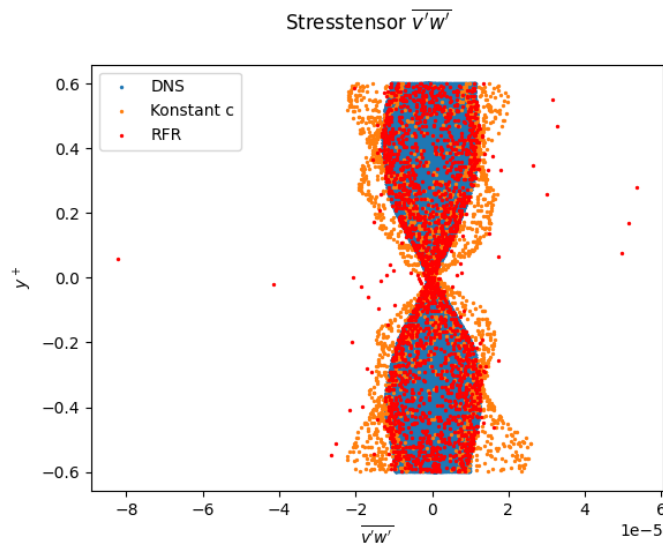
	MAPE RFR	RMSPE RFR	MaAE RFR	MAPE konstant c_i	RMSPE konstant c_i	MaAE konstant c_i
$u'u'$	1.34%	4.29%	$2.3 \cdot 10^{-4}$	24.68%	27.51%	$1.58 \cdot 10^{-4}$
$u'v'$	1.86%	11.31%	$3.1 \cdot 10^{-4}$	34.26%	29.71%	$3.73 \cdot 10^{-5}$
$u'w'$	1.87%	10.4%	$2.8 \cdot 10^{-4}$	34.51%	29.99%	$3.33 \cdot 10^{-5}$
$v'v'$	2.01%	6.61%	$1.6 \cdot 10^{-4}$	17.01%	21.43%	$1.03 \cdot 10^{-4}$
$v'w'$	17.17%	73.04%	$1.6 \cdot 10^{-4}$	39.97%	43.53%	$1.23 \cdot 10^{-5}$

Figur 4.10 och figur 4.11 illustrerar två exempel på hur Reynoldsspänningarna presterar jämfört med DNS-data och konstanta c_i . Även här framgår att RFR predikterar turbulenserna bättre än konstanta c_i samt att avvikelserna sker i enstaka punkter. Man kan även se i figur 4.12 att avvikelserna ser ut som enstaka störningar

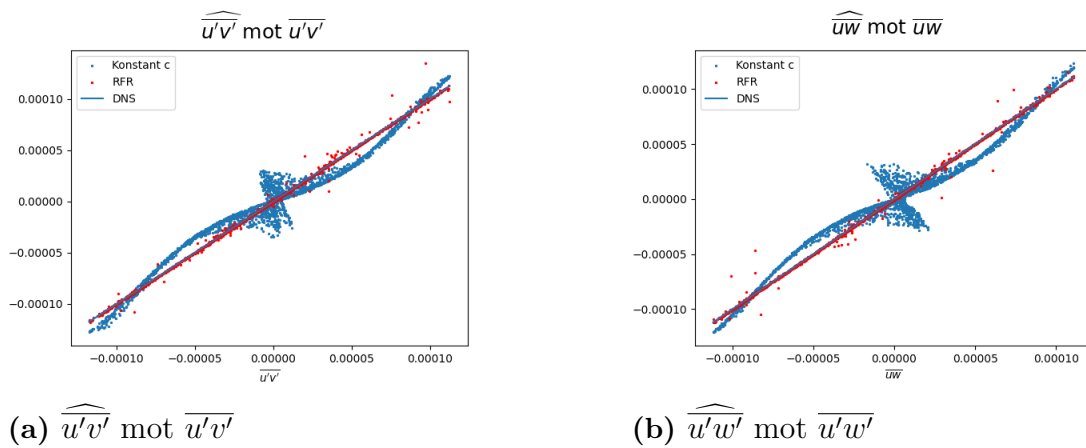
snarare än en tydlig trend.



Figur 4.10: Reynoldsspänning $\overline{u'u'}$ i flöde Square Duct med friktions-Reynoldstal 1000 plottad mot y^+ . Figuren visar DNS-data, resultat av modell med konstant c_i och modell med RFR tränat på samma friktions-Reynoldstal.



Figur 4.11: Reynoldsspänning $\overline{v'w'}$ i flöde Square Duct med friktions-Reynoldstal 1000 plottad mot y^+ . Figuren visar DNS-data, resultat av modell med konstant c_i och modell med RFR tränat på samma friktions-Reynoldstal.



Figur 4.12: Predikterade Reynoldspänningar $\widehat{u'v'}$ och $\widehat{u'w'}$ i flöde Square Duct med friktions-Reynoldstal 1000 plottad mot de samma värdena $\overline{u'v'}$ och $\overline{u'w'}$. Graferna innehåller resultat för konstanta c_i och för modell med RFR tränat på samma friktions-Reynoldstal.

4.6 Generalisering mellan friktions-Reynoldstal för Square Duct

Detta avsnittet redovisar resultaten som fås av generalisering mellan friktions-Reynoldstal för Square Duct. Modellerna tränades med ett friktions-Reynoldstal, antingen 1000 eller 500 och testades sedan på motsatta.

4.6.1 Neurtalt nätverk

Resultat för modellerna med neurala nätverk presenteras i tabell 4.10 och tabell 4.11. Det syns att de genomsnittliga felen blir mindre med neurtalt nätverk än med konstanta c_i för modellen tränat på flöde med friktions-Reynoldstal 1000. Notera att MAPE är lägre i tabell 4.10 än tabell 4.6 för alla Reynoldsturbulenser förutom $\overline{u'w'}$, men där är skillnaden endast 0.6%. Detsamma gäller dock inte för modellen tränad på flöde med friktions-Reynoldstal 500, där ger konstanta c_i genomgående sämre resultat.

Tabell 4.10: Fel för neurtalt nätverk, tränat på Square Duct med friktions-Reynoldstal 1000, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 500.

	MAPE NN	RMSPE NN	MaAE NN	MAPE konstant c_i	RMSPE konstant c_i	MaAE konstant c_i
$u'u'$	8.09%	10.08%	$1.13 \cdot 10^{-2}$	24.68%	27.51%	$1.58 \cdot 10^{-4}$
$u'v'$	25.75%	30.83%	$1.19 \cdot 10^{-4}$	34.26%	29.71%	$3.73 \cdot 10^{-5}$
$u'w'$	29.63%	38.48%	$1.53 \cdot 10^{-4}$	34.51%	29.99%	$3.33 \cdot 10^{-5}$
$v'v'$	5.6%	7.79%	$7.41 \cdot 10^{-5}$	17.01%	21.43%	$1.03 \cdot 10^{-4}$
$v'w'$	33.16%	36.02%	$1.39 \cdot 10^{-5}$	39.97%	43.53%	$1.23 \cdot 10^{-5}$

Tabell 4.11: Fel för neuralt nätverk, tränat på Square Duct med friktions-Reynoldstal 500, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 1000.

	MAPE NN	RMSPE NN	MaAE NN	MAPE konstant c_i	RMSPE konstant c_i	MaAE konstant c_i
$u'u'$	17.35%	24.71%	$3.42 \cdot 10^{-4}$	22.95%	29.13%	$1.72 \cdot 10^{-4}$
$u'v'$	72.64%	115.93%	$4.00 \cdot 10^{-4}$	25.97%	23.69%	$3.98 \cdot 10^{-5}$
$u'w'$	84.02%	134.41%	$4.25 \cdot 10^{-4}$	28.73%	26.99%	$4.95 \cdot 10^{-5}$
$v'v'$	16.23%	27.9%	$2.81 \cdot 10^{-4}$	18.47%	25.26%	$1.03 \cdot 10^{-4}$
$v'w'$	61.51%	76.76%	$3.77 \cdot 10^{-5}$	82.43%	99.46%	$2.27 \cdot 10^{-5}$

4.6.2 Random Forest Regression

Modellerna med RFR presterar allmänt sämre än med konstanta c_i vid generalisering mellan friktions-Reynoldstal, vilket ses i tabell 4.12 och tabell 4.13. Störst fel får generaliseringen för flöde från högt till lågt friktions-Reynoldstal och av Reynolds-spänningarna är det speciellt $\overline{v'w'}$ som modelleras dåligt.

Tabell 4.12: Fel för RFR, tränat på Square Duct med friktions-Reynoldstal 1000, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 500.

	MAPE RFR	RMSPE RFR	MaAE RFR	MAPE konstant c_i	RMSPE konstant c_i	MaAE konstant c_i
$u'u'$	62.65%	400.33%	$1.18 \cdot 10^{-2}$	24.68%	27.51%	$1.58 \cdot 10^{-4}$
$u'v'$	42.47%	103.5%	$1.28 \cdot 10^{-3}$	34.26%	29.71%	$3.73 \cdot 10^{-5}$
$u'w'$	36.96%	94.66%	$2.1 \cdot 10^{-3}$	34.51%	29.99%	$3.33 \cdot 10^{-5}$
$v'v'$	145.66%	1099.04%	$2.2 \cdot 10^{-2}$	17.01%	21.43%	$1.03 \cdot 10^{-4}$
$v'w'$	198.77%	776.5%	$1.6 \cdot 10^{-3}$	39.97%	43.53%	$1.23 \cdot 10^{-5}$

Tabell 4.13: Fel för RFR, tränat på Square Duct med friktions-Reynoldstal 500, och med konstanta c_i , testade på Square Duct med friktions-Reynoldstal 1000.

	MAPE RFR	RMSPE RFR	MaAE RFR	MAPE konstant c_i	RMSPE konstant c_i	MaAE konstant c_i
$u'u'$	11.18%	22.52%	$7.6 \cdot 10^{-4}$	22.95%	29.13%	$1.72 \cdot 10^{-4}$
$u'v'$	27.6%	27.98%	$7.77 \cdot 10^{-5}$	25.97%	23.69%	$3.98 \cdot 10^{-5}$
$u'w'$	33.47%	36.8%	$1.6 \cdot 10^{-4}$	28.73%	26.99%	$4.95 \cdot 10^{-5}$
$v'v'$	15.24%	35.09%	$7.3 \cdot 10^{-4}$	18.47%	25.26%	$1.03 \cdot 10^{-4}$
$v'w'$	174.67%	491.24%	$3.7 \cdot 10^{-4}$	82.43%	99.46%	$2.29 \cdot 10^{-5}$

5

Diskussion

Diskussionsavsnittet syftar till att redogöra tolkningar och betydelse av resultaten som presenterats i rapporten. Metoderna skall ställas mot varandra för att se hur väl de presterar i de olika geometrierna och för olika friktions-Reynoldstal.

En genomgående felkälla för alla geometrier är att alla modeller endast tränats på 80% av ett strömningsfall, dvs. en simulering per geometri. Att inte använda flera simuleringar introducerar risker att modellerna överanpassas till den specifika simuleringen och därmed blir sämre på att generalisera. RFR drabbas speciellt av detta och bättre resultat skulle kunna fås med tillgång till mer DNS-data.

Ytterligare gäller att extrapolering är en svårare uppgift än interpolering för ML-metoder [18]. Exempel på hur detta används i arbetet är generalisering mellan geometrier och mellan friktions-Reynoldstal. Då generaliseringen sker från en simpel till en komplex geometri måste maskininlärningen extrapolera, likaväl när det generaliserar från ett lägre friktions-Reynoldstal. Däremot motsvarar generalisering från en komplex geometri till en simpel och från ett högt friktions-Reynoldstal till ett lågt istället interpolering. Detta leder till att felen förväntas vara större i fallen då generaliseringen motsvarar extrapolering än för fallen då de motsvarar interpolering.

5.1 Kanalströmning och gränsskikt

För kanalströmning och gränsskikt presterade de olika metoderna ungefär lika bra. Båda fick även betydligt bättre resultat än konstant c_i . Neurala nätverket fick större fel än RFR på predikteringen av c_0 och c_2 , men för turbulenserna var istället det neurala nätverket mer precist. Det är även intressant att för både neurala nätverket och RFR påverkar inte de stora felen hos c_i resultaten för turbulenserna. Detta syns speciellt för kanalströmningen, där både neurala nätverket och RFR får stora fel för c_0 , och för gränsskikt där förutsägelsen för c_0 och c_2 varierar mycket.

För båda metoderna samt för båda geometrierna presterar förutsägelseerna av $\overline{v'v'}^+$ sämst. En orsak till detta skulle kunna vara att eftersom systemen är överbestämda används bara a_{11} och a_{33} för att konstruera c_0 och c_2 . Teoretiskt bör detta inte vara ett problem eftersom spåret av a är noll, men ifall DNS-datan inte är perfekt kan det vara problematiskt. Ty varken a_{11} eller a_{33} beror på \bar{v} , alltså tränas möjligtvis inte ML-modellerna på turbulens i den riktningen. Detta skulle kunna vara orsaken

till den här anomaliteten i resultaten.

För generaliseringarna fås bra resultat för båda metoderna och prediktionerna med modellerna är bättre än de konstanta c_i i båda fallen. Speciellt då modellerna tränats på kanalflöde och testats på gränsskikt. En anledning till detta skulle kunna vara att kanalströmning har en mer komplicerad geometri jämfört med gränsskikt. Dessutom är modellen för kanalflöde i överlag bättre än den för gränsskikt, vilket understöds från felen som fås i resultaten. Dock har kanalströmning lägre friktions-Reynoldstal än gränsskikt, vilket bör ge sämre resultat under generaliseringen. Effekten av detta tycks inte övervinna de positiva faktorerna, eftersom generaliseringen i resultatet fungerade bäst åt detta håll. Det är dock svårt att avgöra exakt hur dessa faktorer påverkar resultatet då de är mycket komplicerade.

Anledningen till att båda generaliseringarna får höga procentuella fel jämfört med prediktering på sin egen testdata, men ändå presterar bättre än vad som fås för konstanta c_i kan antas vara för att processerna är komplexa, och därmed även svårmodellerade med konstanter. Därför är det förväntat att modellerna inte kan prediktera på varandra med samma säkerhet som de gör på sin egna testdata. Att de lyckas bättre än konstanta c_i trots de svårigheter som generaliseringen medför, talar för att generalisering är möjlig och går att optimera ytterligare.

5.2 Gropgeometri

Predikteringen på gropgeometri visade sig vara en svår uppgift för båda metoder, speciellt för neurala nätverk. Felen blir relativt stora, och vid närmare undersökning av datan syns det att c_1 och c_3 pendlar mellan positiva och negativa värden, vilket försvårar formuleringen av en funktion. Orsakerna till detta är inte helt uppenbara, men en teori är att gropgeometrin ger så oregelbundna flöden att det inte går att finna något exakt samband med våra parametrar och nuvarande modeller.

Eftersom detta resultat blir avsevärt sämre än för de andra geometrierna, tyder detta på ett grundläggande fel hos modellerna som måste undersökas djupare innan några slutsatser kan dras. Därför kommer modeller med gropgeometri inte bejakas i slutsatserna.

5.3 Square Duct

För Square Duct gäller att ML-metoderna för det mesta presterade bättre än de konstanta c_i . Denna förbättringen är dock inte lika markant för Square Duct som för kanalströmning och gränsskikt, speciellt under generaliseringen. En orsak till detta skulle kunna vara att de konstanta c_i som användes i Square Duct optimerades vid implementeringen, som beskrivet i metoden. En annan anledning skulle kunna vara att fler konstanter användes för modellen med konstanta c_i i Square Duct, jämfört med kanalströmning och gränsskikt. Detta gör att modellen får bättre precision eftersom fler icke-linjära termer tas med i approximationen av turbulenserna. Denna

effekt inte är lika stor för de modellerade c_i , ty de kan anpassas för att kompensera noggrannheten som förloras när få konstanter används.

Som visades i resultaten för Square Duct sker instabilitet vid de extrema värdena av turbulens för neurala nätverket, men mer utspritt och inte till samma grad för RFR. Orsaken till varför denna skillnad uppstår är oklar, men eftersom varken DNS-datan eller ett konstanta c_i uppvisar något sådant beteende måste detta vara något som uppstår under träningen av modellerna. Alltså något problematiskt som inte hanteras. En potentiell lösning på detta skulle kunna vara att sätta ett intervall som predikteringen av turbulenserna måste tillhöra, d.v.s ett maximalt och minimalt värde som turbulenserna kan anta. Det är dock inte säkert att detta skulle eliminera problemet med instabilitet helt, eftersom alla avvikande värden är inom rimliga gränser för resultaten.

5.3.1 Neurtalt nätverk

Modellerna från neurala nätverket predikterar turbulensen relativt bra, med undantag för $\overline{u'v'}$ och $\overline{u'w'}$. För dessa Reynoldsspänningar uppstår problem med instabilitet som orsakar stora fel, detta kan ses i figur 4.8 och figur 4.9. Alltså presterar modellen bra utom i vissa områden där felet blir mycket stora. I testfall då resultatet redan är känt är det lätt att känna igen fel orsakade av instabilitet då dessa avvikelser sticker ut. Däremot för en verklig simulering, där resultatet är okänt, skulle sådana fel kunna vara mycket svåra att upptäcka, särskilt om resultaten till största delen är rimliga. Om problemet med instabilitet går att lösa med full säkerhet är metoden mycket lovande, men tills dess är det ifrågasättbart om metoden går att lita på för verkliga simuleringar.

Till skillnad från fallen med kanalströmning och gränsskikt, leder generalisering med neurala nätverket, här mellan friktions-Reynoldstal, till blandade resultat. För generaliseringen som tränats på friktions-Reynoldstal 1000 och testats på friktions-Reynoldstal 500 presterar neurala nätverket fortfarande bättre än konstanta c_i . Här kan man även se att det är speciellt de Reynoldsspänningar som upplever instabilitet som förbättras. Alltså verkar det som att instabiliteten delvis motverkas av generaliseringen, vilket är oväntat. Ty, generalisering bör vara en svårare uppgift än testning mot egen testdata. En orsak till detta skulle kunna vara att prediktering på flöde med lägre friktions-Reynoldstal är allmänt lättare och att det är detta som frambringat resultatet.

För generalisering åt motsatt håll ger neurala nätverket bättre prediktioner än konstanta c_i endast för de Reynoldsspänningar som inte upplever instabilitet. Detta betyder att en modell som inte upplever instabilitet möjligtvis även hade kunnat generalisera från flöde med lågt friktions-Reynoldstal till ett med högre. Den modell som presenteras i tabell 4.11 är dock ingen lyckad generalisering. Sammanfattningsvis, gäller för neurala nätverket att generaliseringen fungerar bättre när modellen tränas på ett högre friktions-Reynoldstal än det testas på. Detta är väntat på grund

av interpolering, som förklarats tidigare.

5.3.2 Random Forest Regression

Modellerna från RFR får genomgående bra resultat vid predikation på egen testdata. RFR-modellerna presterar bättre än konstanta c_i för flöde med friktions-Reynoldstal 500 samt 1000. Predikteringarna är även bättre än de som fås av det neurala nätverket, vilket är intressant eftersom båda modellerna tränats på samma indata, alltså måste denna noggrannhet härstamma från träningen eller strukturen hos RFR-metoden.

På generaliseringen får RFR avsevärt större fel än konstanta c_i . Här skiljer sig MAPE med en faktor över 10 för alla Reynoldsspänningar, jämfört med vad som fås när träningsdatan och testdatan har samma friktions-Reynoldstal. RFR verkar alltså vara betydligt sämre på generalisering än neurala nätverket. Detta skulle kunna bero på att RFR lyckades producera noggranna modeller, men att dessa är något överanpassade till den DNS-simulering de tränades på och därför ger större fel under generaliseringen. Detta skulle dels kunna undvikas genom att träna modellen på flera DNS-simuleringar, samtidigt som hyperparametrarna skulle kunna justeras mer ingående för att erhålla bättre resultat.

En ytterligare anomalitet är att RFR generaliserar bättre när den tränats på ett flöde med ett lågt friktions-Reynoldstal och testats på flödet med ett högre. Detta är oväntat eftersom det motsvarar att modellen extrapolerar, vilket är generellt svårt för ML. Dessutom presterar modellen för flöde med friktions-Reynoldstal 1000 bättre än den för 500 när de predikterar på sin egna testdata, se tabell 4.9 och tabell 4.8. Detta gör att de resultat som fås för generaliseringen är ännu mer orimliga, ty den bättre modellen generaliserar sämre trots att den interpolerar. Orsaken till detta resultat är okänd, men de fel som fås från RFR under generaliseringen är allmänt underliga och tyder på att metoden inte bör användas till sådana syften eller att den behöver optimeras ytterligare för att kunna ge givande resultat.

6

Slutsatser

Slutsatsavsnittet syftar till att ge en avslutande reflektion av arbetet samt att belysa de viktigaste fynden som uppkommit under arbetets gång. Avsnittet tar även upp förbättringspotential för de olika metoderna samt möjligheter till vidare forskning.

För samtliga geometrier gav både neurala nätverket och RFR betydligt bättre resultat än att använda konstanta c_i , undantaget $\overline{u'v'}$ och $\overline{u'w'}$ för det neurala nätverket i Square Duct samt generalisering. Detta tyder därmed på att det lönar sig att prediktera konstanterna med ML i syfte att optimera turbulensmodellering.

Neurala nätverket tycks vara en bra metod att använda för att modellera turbulens, speciellt i kombination med NEVM. För kanalströmning och gränsskikt fick neurala nätverket bättre resultat än RFR; även för Square Duct lyckades neurala generalisera väl mellan olika friktions-Reynoldstal. Om man lyckas träna en modell utan instabiliteten som förekom för Square Duct, skulle det neurala nätverket vara en bra metod för att möjliggöra generalisering. Dessutom, är det definitivt möjligt att förbättra nätverket genom databehandling eller modifiering av hyperparametrarna som används så att den även producerar eftertraktade resultat för $\overline{u'v'}$ och $\overline{u'w'}$.

Även RFR, specifikt med XGB, är en lämplig metod för att modellera turbulens. Prestandan är generellt hög, men metoden kräver mycket jobb med justering av hyperparametrar för att erhålla kvalitativa resultat, vilket i Square Duct visade sig kräva oväntat stora resurser. Denna justering av hyperparametrar är avgörande för pålitligheten, då RFR har en tendens att överanpassa sig till datan, vilket är speciellt problematiskt då det inte finns tillgång till flera träningsdataset. RFR verkar inte vara en metod som är lämpad för generalisering, utan har istället styrkor där den anpassas till enstaka flöden. En slutsats som kan dras är därför att ett välimplementerat neuralt nätverk är mer allmänt användbart, medan RFR möjligen besitter en högre potential under rätt förutsättningar.

En ytterligare slutsats som kan dras är att generalisering är möjlig, både mellan geometrier och mellan friktions-Reynoldstal. Detta kan ses i kandidatarbetet, då generaliseringen mellan kanalströmning och gränsskikt gav bättre resultat än konstanta c_i . Dock testades endast generalisering av geometrier med enkla flöden och hur väl generalisering av mer komplexa flöden skulle resultera går ej att avgöra ifrån arbetet. Generaliseringen mellan friktions-Reynoldstal för Square Duct presterade väl för neuralt nätverk, avsevärt sämre för RFR. Det är alltså fullt möjligt att generalisera mellan friktions-Reynoldstal. Generaliseringen påverkas bevisligen av såväl

olika flödesgeometrier som friktions-Reynoldstal, och detta är ett område som lämpar sig för vidare undersökning.

Sammanfattningsvis visar arbetet att det finns goda möjligheter för fortsatt användning av ML inom turbulensmodellering. Det är ett område där det finns ett stort behov av nya simuleringmetoder för att kunna analysera olika flöden effektivare än genom att använda konstanter. Både neurala nätverk och RFR har visat stor potential för att kunna fylla ett sådant behov. Trots att detta arbete inte lyckades hitta fungerande modeller för alla geometrier som ämnades, har ändå en potential påvisats och det finns stora förbättringsmöjligheter hos båda metoderna som skulle möjliggöra större användningsområden. Dels gällande optimering av modeller för enskilda flöden, men även för att kunna generalisera mellan geometrier och friktions-Reynoldstal. Därför uppmuntras vidare forskning som fortsätter bygga på de metoder och tillvägagångssätt som använts i arbetet.

Litteratur

- [1] F. Batlouni, B. Elm Jonsson, Fjeldså, N. Ole Persson och L. Ånestrand, “Utveckling av turbulensmodeller med hjälp av maskininlärning i Python,” diss., Chalmers Tekniska Högskola, Göteborg, 2023.
- [2] L. Davidson, *Fluid mechanics, turbulent flow and turbulence modelling*, Göteborg, 2024. URL: https://www.tfd.chalmers.se/~lada/postscript_files/solids-and-fluids_turbulent-flow_turbulence-modelling.pdf.
- [3] L. Davidson, “An Introduction to Turbulence Models,” Institutionen Mekanik och Maritima vetenskaper, Chalmers Tekniska Högskola, Göteborg, tekn. rapport, 1997. URL: https://www.tfd.chalmers.se/~lada/postscript_files/kompendium_turb.pdf.
- [4] L. Davidson, “Using Machine Learning for Improving a Non-Linear $k-\epsilon$ Model: A First Attempt,” Institutionen Mekanik och Maritima vetenskaper, Chalmers Tekniska Högskola, Göteborg, tekn. rapport, 2023. URL: https://research.chalmers.se/publication/538321/file/538321_Fulltext.pdf.
- [5] T. J. Craft, B. E. Launder och K. Suga, “Prediction of turbulent transitional phenomena with a nonlinear eddy-viscosity model,” *International Journal of Heat and Fluid Flow*, årg. Vol 18, nr No 1, febr. 1997.
- [6] N. Keijsers, “Neural Networks,” i *Encyclopedia of Movement Disorders*, Elsevier, jan. 2010, s. 257–259. DOI: 10.1016/B978-0-12-374105-9.00493-7. URL: <https://linkinghub.elsevier.com/retrieve/pii/B9780123741059004937>.
- [7] B. Paprocki, A. Pregowska och J. Szczepanski, “Optimizing information processing in brain-inspired neural networks,” *Bulletin of the Polish Academy of Sciences: Technical Sciences*, årg. 68, nr 2, s. 225–233, 2020, ISSN: 23001917. DOI: 10.24425/bpasts.2020.131844.
- [8] G. Dreyfus, “Neural Networks: An Overview,” i *Neural Networks*, Berlin/Heidelberg: Springer-Verlag, 2002, s. 1–83. DOI: 10.1007/3-540-28847-3_{\ }1. URL: http://link.springer.com/10.1007/3-540-28847-3_1.
- [9] J. Ling, A. Kurzawski och J. Templeton, “Reynolds averaged turbulence modelling using deep neural networks with embedded invariance,” *J. Fluid Mech*, årg. 807, s. 155–166, okt. 2016. DOI: 10.1017/jfm.2016.615. URL: <https://doi.org/10.1017/jfm.2016.615>.

- [10] S. Choi, I. Jung, H. Kim, J. Na och J. M. Lee, “Physics-informed deep learning for data-driven solutions of computational fluid dynamics,” *Korean Journal of Chemical Engineering*, årg. 39, nr 3, s. 515–528, mars 2022, ISSN: 19757220. DOI: 10.1007/s11814-021-0979-x.
- [11] J. Ali, R. Khan, N. Ahmad och I. Maqsood, “Random Forests and Decision Trees,” 2012. URL: www.IJCSI.org.
- [12] S. J. Rigatti, “Random Forest,” *JOURNAL OF INSURANCE MEDICINE Copyright C 2017 Journal of Insurance Medicine J Insur Med*, årg. 47, s. 31–39, 2017. URL: http://meridian.allenpress.com/jim/article-pdf/47/1/31/1736157/inism-47-01-31-39_1.pdf.
- [13] B. Efron och T. Hastie, *Computer age statistical inference : algorithms, evidence, and data science*. 2016, s. 475, ISBN: 9781107149892.
- [14] T. Chen och C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” mars 2016. DOI: 10.1145/2939672.2939785. URL: <http://arxiv.org/abs/1603.02754>
<http://dx.doi.org/10.1145/2939672.2939785>.
- [15] L. Davidson, *Assignment in Machine Learning for turbulence modeling*. URL: <https://www.tfd.chalmers.se/~lada/ML-IMS135/>.
- [16] D. Modesti, *Direct numerical simulation dataset of turbulent square duct flow*, maj 2020. DOI: <https://doi.org/10.4121/uuid:3226d808-f354-47b1-b6f8-3a8048215193>.
- [17] H. Gnanga, H. Naji och G. Mompean, “Computation of a three-dimensional turbulent flow in a square duct using a cubic eddy-viscosity model,” *Comptes Rendus - Mécanique*, årg. 337, nr 1, s. 15–23, jan. 2009, ISSN: 16310721. DOI: 10.1016/j.crme.2009.02.001.
- [18] G. Martius och C. H. Lampert, “Extrapolation and learning equations,” tekn. rapport, okt. 2016. DOI: <https://doi.org/10.48550/arXiv.1610.02995>. URL: <http://arxiv.org/abs/1610.02995>.

A

Uttryck för okända konstanter

För att ta fram formler för gropgeometri använde vi oss av en symbolisk hanterare i Python kallad `SymPy`. Skjuvningstensorn s_{ij} och rotationstensorn Ω_{ij} definieras enligt (2.19). Då vi valt att endast titta på konstanterna c_1 , c_2 och c_3 , och sätter resterande till noll, kommer vi ha $n = 3$. \mathbf{a}^* , \mathbf{b} och \mathbf{c} får storlek (3×1) och \mathbf{A} får storlek (3×3) . För att förenkla processen väljer vi att skapa en ny matris \mathbf{a}' som vi definierar som differensen mellan \mathbf{a}^* och \mathbf{b}

$$\mathbf{a}' = \mathbf{a}^* - \mathbf{b}. \quad (\text{A.1})$$

Systemet vi vill lösa är därmed $\mathbf{a}' = \mathbf{A}\mathbf{c}$. Detta går att göra med hjälp av `SymPy` funktionen `solve`, där vi även har definierat \mathbf{A} som uttrycken framför varje c_i i (2.21). Användning av `solve` finner följande formler på konstanterna:

$$c_1 = \frac{2a_{11}s_{11} - 2a_{11}s_{22} + 4a_{12}s_{12} + a_{33}s_{11} - a_{33}s_{22} + 4c_\mu t(s_{11}s_{11} - s_{11}s_{22} + 4s_{12}s_{12})}{t^2(s_{11}s_{11}s_{11} - s_{11}s_{11}s_{22} + 4s_{11}s_{12}s_{12} - s_{11}s_{22}s_{22} + 4s_{12}s_{12}s_{22} + s_{22}s_{22}s_{22})}$$

$$c_2 = \frac{2a_{11}s_{12} - a_{12}s_{11} + a_{12}s_{22} + a_{33}s_{12} + 2c_\mu t(s_{11}s_{12}t + s_{12}s_{22})}{\Omega_{12}t^2(s_{11}s_{11} - 2s_{11}s_{22} + 4s_{12}s_{12} + s_{22}s_{22})}$$

$$\begin{aligned} c_3 = & (-a_{11}s_{11}s_{11}s_{11} + a_{11}s_{11}s_{11}s_{22} - 2a_{11}s_{11}s_{12}s_{12} - a_{11}s_{11}s_{22}s_{22} + 2a_{11}s_{12}s_{12}s_{22} \\ & + a_{11}s_{22}s_{22}s_{22} - 2a_{12}s_{11}s_{11}s_{12} - 4a_{12}s_{12}s_{12}s_{12} - 2a_{12}s_{12}s_{22}s_{22} - 2a_{33}s_{11}s_{11}s_{11} \\ & + 2a_{33}s_{11}s_{11}s_{22} - 7a_{33}s_{11}s_{12}s_{12} + a_{33}s_{11}s_{22}s_{22} - 5a_{33}s_{12}s_{12}s_{22} - a_{33}s_{22}s_{22}s_{22} \\ & - 2c_\mu s_{11}s_{11}s_{11}s_{11}t + 2c_\mu s_{11}s_{11}s_{11}s_{22}t - 8c_\mu s_{11}s_{11}s_{12}s_{12}t - 2c_\mu s_{11}s_{11}s_{22}s_{22}t \\ & + 4c_\mu s_{11}s_{12}s_{12}s_{22}t + 2c_\mu s_{11}s_{22}s_{22}s_{22}t - 8c_\mu s_{12}s_{12}s_{12}s_{12}t - 4c_\mu s_{12}s_{12}s_{22}s_{22}t) \\ & \div \\ & t^2\Omega_{12}\Omega_{12}(s_{11}s_{11}s_{11} - s_{11}s_{11}s_{22} + 4s_{11}s_{12}s_{12} - s_{11}s_{22}s_{22} + 4s_{12}s_{12}s_{22} + s_{22}s_{22}s_{22}) \end{aligned} \quad (\text{A.2})$$

Koden som användes för att få fram dessa symboliska uttryck hittas i Appendix B.

B

Komplett kod

[Länk till komplett kod och data på GitLab](#)