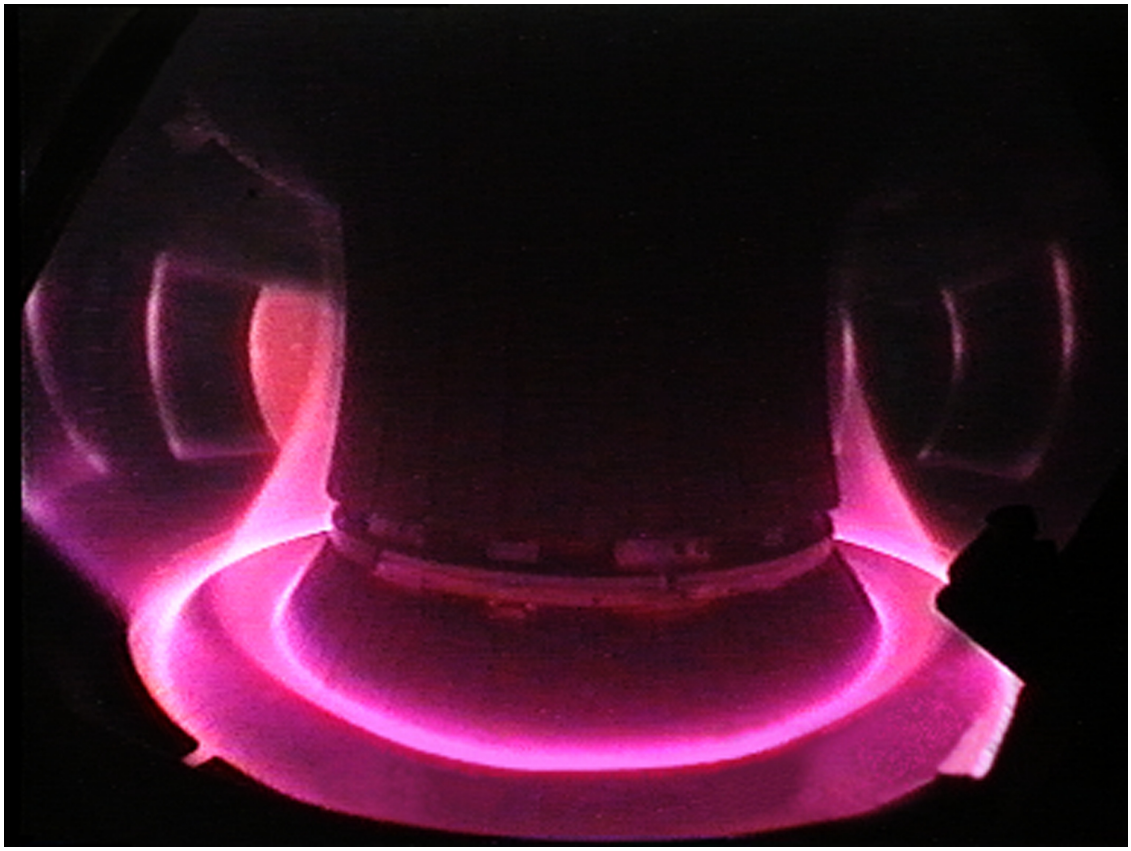
CHALMERS



Konstruktion av tolkbara neurala nätverk för analys av QuaLiKiz-modellen

Undersökning av hur neurala nätverk predikterar instabiliteter i ett fusionsplasma

Kandidatarbete inom programmen Teknisk fysik och Datateknik

William Enström, Mattias Kvartsén,
Ylva Liljegren och Hedvig Wennberg

INSTITUTIONEN FÖR RYMD-,GEO- OCH MILJÖVETENSKAP
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2024
www.chalmers.se

Kandidatarbete 2024

Konstruktion av tolkbara neurala nätverk för analys av QuaLiKiz-modellen

Undersökning av hur neurala nätverk predikterar instabiliteter i ett fusionsplasma

WILLIAM ENSTRÖM, MATTIAS KVARTSÉN,
YLVA LILJEGREN och HEDVIG WENNBERG

CHALMERS

Institutionen för Rymd-, geo- och miljövetenskap
Chalmers Tekniska Högskola
Göteborg, Sverige 2024

Konstruktion av tolkbara neurala nätverk för analys av QuaLiKiz-modellen
Undersökning av hur neurala nätverk predikterar instabiliteter i ett fusionsplasma
WILLIAM ENSTRÖM, MATTIAS KVARTSÉN,
YLVA LILJEGREN och HEDVIG WENNBERG

Handledare: Andreas Gillgren och Andrei Osipov, Rymd-, geo- och miljövetenskap
Examinator: Pär Strand, Rymd-, geo- och miljövetenskap

Kandidatarbete 2024
Institutionen för Rymd-, geo- och miljövetenskap
Chalmers Tekniska Högskola
SE-412 96 Göteborg
Telefon +46 31 772 1000

Omslagsbild: Plasma i ASDEX Upgrade; en tokamak vid Max Planck Institut för plasmafysik (Fotografi: MPI for Plasma Physics). Bilden används med tillstånd från MPI for Plasma Physics.

Skriven i LaTeX
Göteborg, Sweden 2024

Konstruktion av tolkbara neurala nätverk för analys av QuaLiKiz-modellen
Undersökning av hur neurala nätverk predikterar instabiliteter i ett fusionsplasma
WILLIAM ENSTRÖM, MATTIAS KVARTSÉN, YLVA LILJEGREN och HEDVIG WENNBERG
Institutionen för Rymd-, geo- och miljövetenskap
Chalmers Tekniska Högskola

Sammandrag

Fusion är en förnybar energikälla som bygger på att man slår samman lätta atomer för att utvinna energi. Ett av dagens problem med fusionsreaktorer är uppkomsten av turbulenta instabiliteter i fusionsplasmat, dessa begränsar möjligheterna att få ut mer energi än vad som förbrukas. Denna rapport ämnar till att öka förståelsen för hur maskininlärningsmodeller förutsäger instabiliteterna i plasmat i en fusionsreaktor. Detta gjordes genom att konstruera neurala nätverk i olika konfigurationer som tränades på en datamängd producerad av den numeriska modellen QuaLiKiz, som beräknar tillväxten av turbulenta instabiliteter i plasmat. Inledningsvis gjordes detta genom att konstruera en autoencoder för att se hur mycket QuaLiKiz-datan kan komprimeras. Därefter undersöktes vilka parametrar som var mest relevanta för att prediktera hur snabbt instabiliteter växer på elektronskalan. Efter detta undersöktes hur parametrarna kan kombineras för att underlätta analys av det neurala nätverket. Slutligen analyserades relationerna mellan de relevanta parametrarna via scatterplots. Resultaten indikerar att tillväxthastigheterna på elektronskalan som beräknats av QuaLiKiz har en lågdimensionell representation i två datapunkter. De viktigaste storheterna för att representera instabiliteterna på elektronskalan verkar vara: elektrondensitetsgradienten, elektrontemperaturgradienten, kollisionalitet, säkerhetsfaktor, magnetisk skjuvning, jon-elektrontemperatur-ratio och effektivt atomnummer i plasmat. Resultaten visar även att analysen av det neurala nätverket kan underlättas genom att dela upp den neurala nätverkstrukturen i olika grenar. Med denna metod kan därmed relationerna mellan storheterna som påverkar instabiliteterna i fusionsplasmat lättare fastställas.

Nyckelord: Maskininlärning, neurala nätverk, fusionsplasma, QuaLiKiz, turbulenta instabiliteter.

Construction of interpetable neural networks for analysis of the QuaLiKiz-model
Investigation of how neural networks predics intabilities in a fusion plasma
WILLIAM ENSTRÖM, MATTIAS KVARTSÉN, YLVA LILJEGREN and
HEDVIG WENNBERG
Department of Space, Earth and Environment
Chalmers University of Technology

Abstract

Fusion is a renewable energy source that is based on colliding light atoms in order to extract energy. One of the challenges present in existing fusion reactors is the occurrence of turbulent instabilities in the fusion plasma, which inhibit the ability to extract more energy than what is consumed. This report aims to increase understanding of how machine learning models predict instabilities in plasma within a fusion reactor. This was achieved by constructing neural networks in various configurations trained on a dataset produced by the numerical model QuaLiKiz, a model which calculates the growth of turbulent instabilities in the plasma. Initially, an autoencoder was constructed to assess the compressibility of the QuaLiKiz data. Subsequently, the most relevant parameters for predicting the growth rates of instabilities on the electron scale were investigated. Following this, the combination of parameters to facilitate neural network analysis was explored. Finally, relationships between relevant parameters were analyzed via scatter plots. Results indicate that growth rates on the electron scale computed by QuaLiKiz have a low-dimensional representation in two data points. The key variables for representing electron-scale instabilities appear to be: electron density gradient, electron temperature gradient, collisionality, safety factor, magnetic shear, ion-to-electron temperature ratio and effective atomic number in the plasma. Furthermore, the analysis of the neural network can be aided by dividing its structure into different branches. With this method, relationships between variables affecting the instabilities in fusion plasma can be more easily established.

Keywords: Machine learning, neural networks, fusion plasma, QuaLiKiz, turbulent instabilities.

Förord

Vi vill tacka våra handledare Andreas Gillgren och Andrei Osipov för deras värdefulla stöd och vägledning under detta kandidatarbete. Deras feedback och mentorskap har varit avgörande.

Lista med förkortningar

Nedan följer en lista med förkortningar som används i rapporten, listade i alfabetisk ordning:

Adam	Adaptive Moment Estimation
DNN	Dense Neural Network
ETG	Electron Temperature Gradient
ITG	Ion Temperature Gradient
MAE	Mean Absolute Error
MSE	Mean Squared Error
ReLU	Rectified Linear Unit
TEM	Trapped Electron Mode

Innehåll

1

Introduktion

1.1 Bakgrund

Enligt International Energy Agency står kärnkraft idag för "cirka 10 procent av världens elproduktion" [1]. Principen som denna kärnkraft bygger på kallas fission och den nyttjar den energi som frigörs då atomkärnor klyvs. Förutom den stabila energitillförsel som fission erbjuder tillkommer många problem. Ett av dessa problem är det seglivade radioaktiva avfall som utgör processens restprodukt.

Ett annat problem är att fissionskraft är beroende av uran eller plutonium för att fungera. Dessa kärnbränslen är begränsade och farliga råvaror. Dessutom finns det en potentiell risk för härdsmälta i fissionsreaktorer. Reaktionen skenar då utom kontroll vilket kan få förödande konsekvenser, ett exempel på en sådan incident är Fukushima-olyckan 2011 [2].

Ett alternativt koncept är fusionskraft, vilket bygger på den energi som frigörs då atomer slås samman. Fusionskraft saknar många av nackdelarna hos fissionskraft. För det första består reaktionens huvudsakliga restprodukt av helium, som är en ofarlig ädelgas. För det andra drivs reaktionen av väteisotoperna deuterium och tritium istället för uran eller plutonium. Deuterium är stabilt och förekommer i naturligt vatten [3]. Tritium emellertid är radioaktivt men är en av de minst farliga radionukliderna då strålningen som avges är svag [4]. Utöver detta finns det ingen risk att fusionsreaktionen skenar då det krävs väldigt noggranna förhållanden för att upprätthålla reaktionen. Däremot kan kraftfulla urladdningar i fusionsplasmat ske vilket kan skada reaktorn drastiskt, detta påverkar dock endast reaktorn och utgör ingen fara för allmänheten [5].

Dagens problem är att lyckas utforma en fusionsreaktor som har tillräckligt hög verkningsgrad för att man ska kunna utvinna mer energi än vad som förbrukas för att upprätthålla reaktionen. Detta har visat sig vara ett väldigt komplext problem, men med hjälp av numeriska modeller kan man lättare analysera och utforska de extrema förhållanden som krävs för att kontrollerat överskrida den starka kärnkraften hos atomen.

Modeller av naturliga processer kan sammanvävas till en numerisk modell. Ett exempel på en av dessa numeriska modeller är QuaLiKiz som beräknar tillväxten

av turbulenta instabiliteter i plasmat som används vid fusion. Dessa instabiliteter styr till stor del transporten av partiklar och energin i plasmat. Instabiliteterna kan resultera i att partiklar läcker ut ur det inneslutande magnetfältet i reaktorn, vilket begränsar möjligheterna att upprätthålla plasmat [6]. Man har tidigare skapat surrogatmodeller av QuaLiKiz där neurala nätverk tränats med data från QuaLiKiz för att snabba upp tiden för att nå en prediktion [7]. Detta eftersom QuaLiKiz kräver en lång beräkningstid jämfört med tiden det tar att göra en prediktion med ett neuralt nätverk.

Ett problem med neurala nätverk är att bakgrunden till de prediktionerna som ges av modellen är svårförklarade. Det är inte enkelt att förstå exakt hur inputs påverkar outputs vilket har lett till att neurala nätverk ofta kallas för svarta lådor. För att kunna få inblick i vilka val som görs av det neurala nätverket krävs därmed analys av nätverkets arkitektur och hur denna påverkar modellens tolkbarhet. Att förstå hur en modell gör sina prediktioner är viktigt då det gör det lättare att lita på prediktionerna och underlättar felsökning av modellen. Således ämnar detta projekt att, genom att utforska sambandanden mellan in- och utdata i QuaLiKiz-datamängden, bidra med ökad förståelse inom fusionsforskning och underlätta vid framtida konstruktion av surrogatmodeller av QuaLiKiz.

1.2 Syfte

Projektet strävar mot att få en bättre förståelse för hur de maskininlärningsmodeller som används inom fusionsforskning gör sina prediktioner. Syftet är att modellerna ska bli mer tolkbara, vilket kan mynna ut i ny kännedom om fysiken som representeras av QuaLiKiz. Några frågor som behandlas i rapporten är:

- Hur mycket kan datan om instabiliteternas tillväxthastigheter som predikteras av QuaLiKiz komprimeras?
- Vilka inputparametrar är mest relevanta för prediktionen av tillväxthastigheterna?
- Hur kan olika inputparametrar kombineras i det neurala nätverket för att finna en nätverksarkitektur som är lättare att analysera men samtidigt ger ett lågt fel?
- Kan analys av parameterkombinationer ge insikt i relationerna mellan de fysikaliska storheterna och instabiliteternas tillväxthastighet?

1.3 Avgränsningar

QuaLiKiz beräknar instabiliteter vid spatiala skalor från jon- till elektronnivå. Detta arbete undersöker endast instabiliteternas tillväxthastigheter och huvudsakligen tillväxthastigheterna på elektronskalan.

Ytterligare en avgränsning är att endast tillväxthastigheten hos den starkaste instabiliteten som QuaLiKiz beräknar analyseras, även om QuaLiKiz också kan beräkna den näst starkaste instabiliteten.

2

Teori

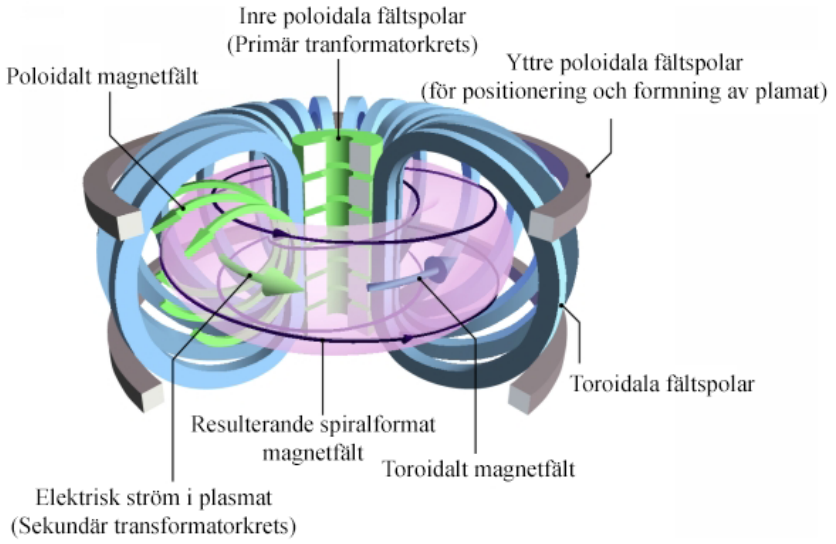
2.1 Fusion

Fusion innebär att två atomkärnor slås samman och bildar en tyngre kärna. Energin som behöver tillföras eller avges i reaktionen är skillnaden mellan bindningsenergierna hos atomerna före och efter reaktionen. För lätta atomkärnor ökar bindningsenergin med fler nukleoner vilket gör det möjligt att utvinna energi genom att slå samman två lätta atomer och skapa en tyngre och starkare bunden atomkärna [8].

Den vanligaste metoden för att skapa gynsamma förhållanden för kontrollerad fusion är magnetisk inneslutning av plasma. Kraften som partiklarna upplever är, med hänsyn tagen till laddade partiklars påverkan av Lorentz-kraften i magnetfält:

$$F = q\mathbf{v} \times \mathbf{B}, \tag{2.1}$$

där q är laddning, $\mathbf{v}$ är hastigheten och $\mathbf{B}$ är magnetfältet. Ekvation 2.1 medför att partikeln enkelt kan röra sig i samma riktning som magnetfältet eftersom kraften då är noll. Om den rör sig vinkelrätt mot fältet får den en cirkulär bana med fix radie. Av detta följer att laddningar rör sig spiralformat runt magnetiska fältlinjer.



Figur 2.1: Schematisk bild av tokamak som visar magnetfältens fältlinjer samt resulterande magnetfält [9, omarbetat], CC-BY 3.0.

En realisering som minskar partikelförlust är användning av speciellt utformade reaktorer med donut-form som kallas tokamaker. En schematisk bild av en tokamak visas i figur 2.1. Inuti tokamaken skapas ett torodialt magnetfält med hjälp av ström genom en rund solenoid utanpå tokamaken. Fältet blir svagare mot den yttre kanten av torusen, vilket skapar instabiliteter av partiklarnas omloppsbanor. Därav krävs ytterligare ett pålagt fält som är poloidalt. Effekten av båda fälten är partiklar som rör sig likt en helix runt torusen [8]. Dock sker avvikelser från en perfekt sådan bana som följer fältlinjerna på grund av exempelvis skjuvningshastigheten och magnetfältsgradienter, vilket kan leda till mikroinstabiliteter i plasmat [6].

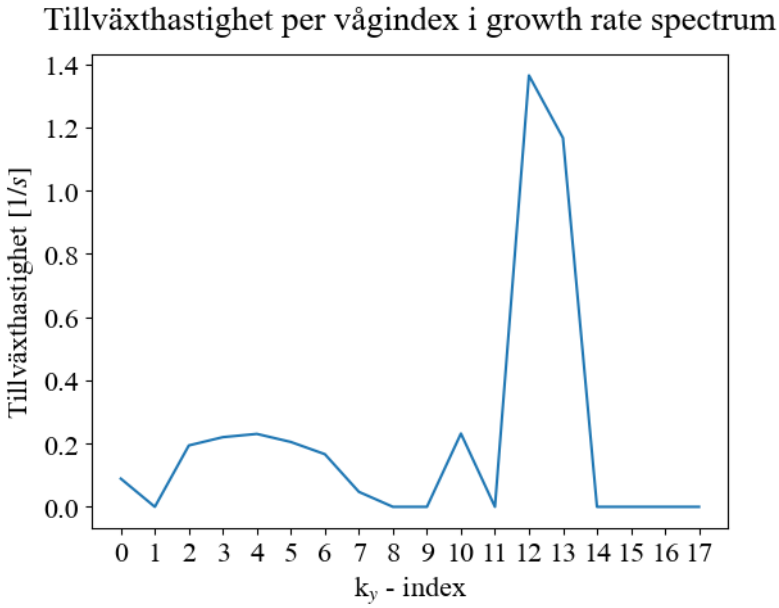
2.1.1 Instabiliteter i plasmat

För att studera och arbeta med tokamaker krävs en förståelse för den turbulenta transporten som uppkommer från mikroinstabiliteter i plasmat. Dessa instabiliteter styr till stor del transporten av partiklar och energi i plasmakärnan och begränsar möjligheterna att upprätthålla inneslutning av hög temperatur och densitet i plasmat [6]. Exempel på instabiliteter som leder till turbulent transport i plasmat är: *ion temperature gradient* (ITG), *electron temperature gradient* (ETG) samt *trapped electron mode* (TEM) [10]. ITG-instabiliteter uppkommer då temperaturen i plasmat inte är jämnt fördelad vilket skapar temperaturgradienter. När temperaturgradienten fluktuerar gör detta att partiklar med olika laddning separeras. Detta skapar ett elektriskt fält i den poloidala riktningen vilket gör att jonerna transporteras i plasmat. Denna transport av jonerna förstärker instabiliteten ytterligare på den yttre sidan av tokamaken. ETG-instabiliteter uppkommer på samma sätt från temperaturgradienten men med andra spatiala skalor och innebär att elektronerna transporteras istället för joner [11]. Eftersom elektronskalan är mycket mindre än jonskalan har oftast ITG-instabiliteten en mycket större påverkan på turbulensen än ETG. TEM-instabiliteten

uppkommer på grund av att magnetfältets styrka varierar i tokamaken. Detta gör att elektroner som inte har tillräckligt hög rörelseenergi inte klarar av att röra sig runt hela tokamaken utan istället rör sig fram och tillbaka på ena sidan av tokamaken [10].

2.2 QuaLiKiz

QuaLiKiz är en numerisk modell som modellerar instabiliteternas tillväxthastigheter, alltså hur snabbt mikorinstabiliteterna i ett tokamakplasma växer för olika vågnummer [12]. Dessa vågnummer är representerade som 18 k_y-index, i ett så kallat *growth rate spectrum*, i den outputdatamängd som utgjorde projektets test- och träningsdata. De första 9 av dessa 18 index representerar tillväxthastigheterna för de starkaste instabiliteterna i ITG-spektrumet medan resterande 9 index representerar tillväxthastigheterna för de starkaste instabiliteterna i ETG-spektrumet. Utöver dessa 18 index finns det ytterligare 54 index i outputdatamängden som representerar andra tillväxthastigheter och realfrekvenser i plasmat, men dessa används inte i detta arbete. Ett exempel på *growth rate spectrum* för ITG och ETG visas i figur 2.2.



Figur 2.2: Exempel på *growth rate spectrum* hämtad från outputdatamängden som används för träning av de neurala nätverken. Y-axelns enhet är normaliserad frekvens där s är sekunder. k_y - index 0-8 är ITG-spektrum och k_y - index 9-17 är ETG-spektrum. Det är endast dessa 18 index som behandlas i detta arbete.

De instabiliteter som modelleras i QuaLiKiz är ITG, ETG samt TEM [6]. Simuleringen baseras på numeriska lösningar av differentialekvationer som beskriver plasmat i tokamaken. Inputparametrarna till QuaLiKiz påverkar variablerna i differentialekvationerna och således lösningarna till dessa. QuaLiKiz inputparametrar visas i tabell 2.1.

Tabell 2.1: Inputparametrar i QuaLiKiz.

Inputvariabel	**Namn i QuaLiKiz**
Norm. tryckgradient	alpha (α)
Norm. elektrondensitetgradient	Ane (A_{ne})
Norm. jondensitetgradient för deuterium	Ani0 (A_{ni1})
Norm. elektrontemperaturgradient	Ate (A_{te})
Norm. jontemperaturgradient	Ati0 (A_{ti0})
Rotationsgradient	Autor (A_{utor})
ExB skjuvningshastighet	gammaE (γ_E)
Kollisionalitet *	logNustar (ν_*)
Toroidalt machtal †	Machtor (M_ϕ)
Norm. jondensitet för jonslag 1 ‡	normni1 (N_{ni1})
Säkerhetsfaktor §	q
Magnetisk skjuvning	smag ($\hat{s}$)
Jon-elektrontemperatur-ratio	Ti-Te0 (τ)
Norm. radiell position	x
Effektivt atomnummer i plasmat ¶	Z_eff (Z_{eff})

* Ett mått på hur ofta partiklarna kolliderar (byter riktning).
† Plasmats rotationshastighet i förhållande till ljudhastigheten.
‡ Jonslag 1 är lätta föroreningar med Z = 1-10, se ¶.
§ Förhållande mellan toroidalt och poloidalt magnetfält.
¶ Viktat medelvärde, ökat atomnumer härstammar från föroreningar från exempelvis reaktorväggen.

2.3 Maskininlärning

Maskininlärning innebär att man använder datorprogram som försöker lära sig av data. Programmen är skrivna med en algoritm som försöker hitta mönster i data eller förutsäga resultat utifrån den givna datan utan att varje specifikt scenario finns beskrivet i koden. Algoritmen tränas på datan och skapar sedan sin egen logik utifrån datan [13]. Maskininlärningsalgoritmen bygger på att man har en modell M som tar indatan $\vec{X}$ och genererar utdatan $\vec{Y}$ enligt $\vec{Y} = M(\vec{X}, \vec{w})$. Hur modellen översätter indatan till utdatan bestäms av modellens vikter, $\vec{w}$, som justeras genom träning av modellen [14].

För att kvantifiera hur väl en modell kan göra prediktioner används en kostnadsfunktion $C(\vec{w})$. Vanliga sådana är medelkvadratfelet (MSE),

$$MSE(\vec{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.2}$$

eller medelabsolutfelet (MAE),

$$MAE(\vec{w}) = \frac{1}{N}\sum_{i=1}^{N}|(y_i - \hat{y}_i)| \tag{2.3}$$

Här är y_i det korrekta värdet och $\hat{y}_i$ det predikterade värdet. Kostnadsfunktionen används både i träningen av modellen och för att mäta hur väl modellen fungerar på data den inte tränats på. När man tränar modellen vill man minimera kostnadsfunktionen. En vanlig optimeringsalgoritm för detta är *Gradient descent*. Algoritmen kan sammanfattas till tre steg enligt [13],

1. Börja med vektor med slumpade vikter $\vec{w}_0$.
2. Vid iteration n:
   i. Evaluera kostnadsfunktionens gradient med avseende på vikterna vid $\vec{w}_n$, $\nabla\vec{C}_n \equiv \nabla_{\vec{w}} C(\vec{w})|_{\vec{w}=\vec{w}_n}$
   ii. Välj en steglängd η_n. Hastigheten kan vara samma för alla iterationer eller minska enligt någon funktion.
   iii. Justera vikterna genom att gå i riktningen som har en negativ gradient: $\vec{w}_{n+1} = \vec{w}_n - \eta_n \nabla\vec{C}_n$.
3. Upprepa tills att $\nabla\vec{C}_n$ är mindre än något valt värde eller efter ett visst antal iterationer.

En annan vanlig optimeringsalgoritm är *Adaptive moment estimation* (Adam) som fungerar på samma sätt som *Gradient descent*, men den inkluderar även en momentparameter. Denna tittar på tidigare riktningar algoritmen har gått i för att prediktera ifall gradienten förväntas fortsätta i denna riktning och därmed öka steglängd för att effektivisera beräkningar. En för liten steglängd gör att det tar längre tid att hitta optimala vikter och modellen fastnar lättare i lokala minimum av kostnadsfunktionen. Ett lokalt minimum är inte nödvändigtvis det minsta modellfelet som kan uppnås utan målet är att hitta ett globalt minimum. Å andra sidan kan en för stor steglängd göra att modellen aldrig konvergerar, det vill säga att inget minimum hittas eftersom för drastiska ändringar sker. Optimeringsalgoritmer såsom Adam hjälper till att förhindra dessa problem, men det är inte garanterat att problemen ej uppstår ändå.

Datan som används för maskininlärningsmodeller delas ofta in i tre delar [13]:

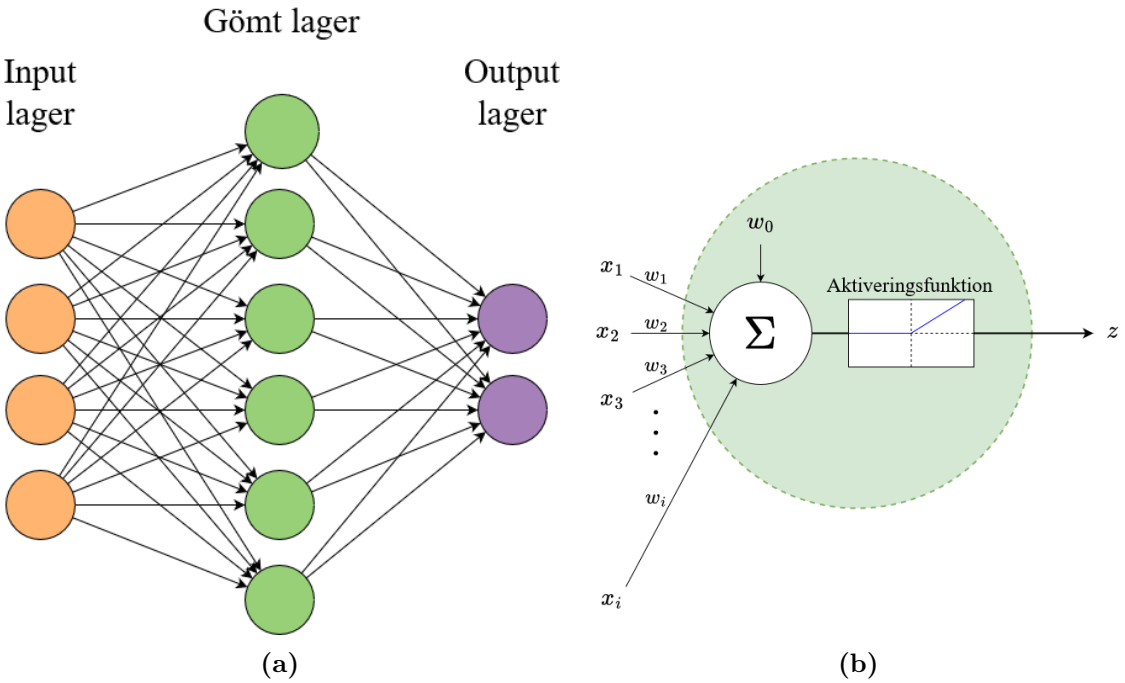
- Träningsdata, som används för att träna modellen.
- Valideringsdata, som används för att se hur bra modellen fungerar efter träningen och för att sedan justera hyperparametrar som exemepelvis inlärningshastighet.
- Testdata, som används som ett slutgiltligt test för hur bra prediktioner som modellen kan göra.

2.4 Neurala Nätverk

Detta arbete använder sig av en del inom maskininlärning som kallas för neurala nätverk. Principen bakom neurala nätverk grundar sig i hur biologiska system, exempelvis hjärnan, interagerar genom att skicka elektriska signaler mellan neuroner där kopplingarna modifieras alltefter som ny träning fås. Detta teoretiseras

med skapandet av en arkitektur som innehåller godtyckligt många lager av noder (artificiella neuroner), med funktioner som kopplar dem till varandra [13].

Figur 2.3 visar en schematisk bild på ett neuralt nätverk och en enskild nod i ett neuralt nätverk.



Figur 2.3: Figur a visar ett neuralt närverk med ett gömt lager.
Figur b visar en enskild nod i ett neuralt nätverk.

Varje nod genererar en signal z enligt,

$$z = f\left(\sum_{i=1}^{n} w_i x_i + w_0\right). \tag{2.4}$$

Noden applicerar aktiveringsfunktionen f på en summa av insignalerna $x_1, ..., x_n$ viktad med vikterna $w_1, ..., w_n$. I aktiveringsfunktionen används också en konstant term w_0, så kallad bias. Ett lager med M stycken noder genererar då signalerna,

$$z_j = f\left(\sum_{i=1}^{n} w_{ji} x_{ji} + w_{j0}\right), \tag{2.5}$$

där $j = 1, ..., M$. Signalerna som skapas i lagret kan sedan användas som insignaler i ett nytt lager med noder vilket genererar K stycken utsignaler,

$$y_k = f\left(\sum_{j=1}^{M} w_{kj} z_j + w_{k0}\right), \tag{2.6}$$

där $k = 1, ..., K$ [15]. Det neurala nätverket kan göras mer komplext genom att man lägger till fler lager mellan input- och outputlager. Dessa lager kallas för "gömda lager".

Aktiveringsfunktionen gör det möjligt för modellen att skapa icke-linjära signaler vilket möjliggör för nätverket att hantera mer komplex data. Valet av aktiveringsfunktion påverkar således hur modellen presterar och vilken aktiveringsfunktion som presterar bäst beror på datan som ska analyseras. En vanlig aktiveringsfunktion är ReLU,

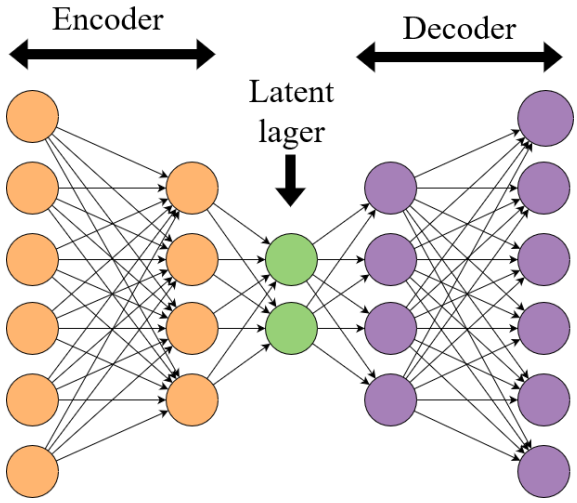
$$f_{ReLU}(x) = max(0, x), \tag{2.7}$$

som är 0 för negativa värden och linjär för positiva värden på x.

Neurala nätverk är mycket användbara då de kan fungera som flexibla funktionsapproximatorer. Nätverket blir väldigt bra på att prediktera den typ av data den tränas på vilket gör att man potentiellt kan träna en modell som ger pricksäkra prediktioner av en output från inputs trots att relationen mellan input och output är komplicerad.

2.4.1 Autoencoders

En typ av neurala nätverk som ofta används för att kunna lagra information med mindre minne är autoencoders, vars syfte är att återskapa data av samma typ som den tränas på. Figur 2.4 visar en schematisk bild på en autoencoder. Processen består av en *encoder* som komprimerar datan till ett visst antal latenta noder som är färre till antal än inputnoderna. Ju färre antal latenta noder, desto mer tvingas modellen att endast behålla information om de mest övergripande dragen. Därefter kommer en *decoder* vars uppgift är att rekonstruera datan. Målet är alltså att input ska vara lika med output, men inte för godtyckliga dataset utan helst bara för data som liknar datan den tränats på. Detta är viktigt för att modellen ska bli användbar, därför att den i så fall kan ge kännedom om tidigare okända egenskaper och samband i datan. Exempelvis kan antalet latenta noder som autoencodern lyckas återskapa datan på inom ett givet felintervall ge insikt om hur få parametrar som krävs för att beskriva de övergripande mönster som finns i datan [16].

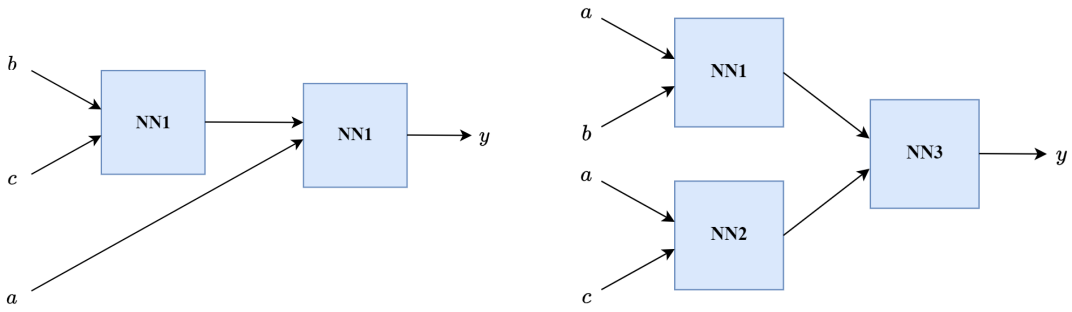


Figur 2.4: Schematisk bild av en autoencoder med två latenta noder och med ett gömt lager i encoder och decoder.

2.4.2 Grenmodeller

Generellt går det inte att se varför ett neuralt nätverk gör en viss prediktion. Betrakta det enkla sambandet $y = a(b + c)$ för olika inputs a, b och c. Ekvivalent kan det skrivas $y = ab + ac$. Ett neuralt nätverk som lyckas återskapa sambandet kan ha hittat antingen det första, andra, eller något ytterligare sätt. För att kunna analysera hur nätverket predikterar har det under projektets gång utvecklats en alternativ neural nätverksstruktur som givits namnet grenmodell.

Grenmodellen bygger på att man, istället för ett *Dense Neural Network* (DNN) där alla parametrar går in i samma neurala nätverk, delar upp nätverket i mindre neurala nätverk som enbart behandlar vissa av inputparametrarna. Exempelvis för sambandet ovan kan två olika konstruktioner skapas, vilka visas i figur 2.5: En grenmodell där b och c paras ihop till en output, som sedan blir input till en ihopparning med a likt Figur 2.5a. Alternativt en grenmodell där a paras ihop med både b och c till input i nästa del av nätverket som i figur 2.5b. Modellfelen borde bli i princip lika stora för dessa nätverk eftersom båda alternativen är likvärdiga sätt att uttrycka datan på.



(a) Grenmodell för $y = a(b + c)$ **(b)** Grenmodell för $y = ab + ac$

Figur 2.5: Två alternativa uppbyggnader av en grenmodell, med flera sammankopplade neurala nätverk, som predikterar y.

Naturligtvis är så inte alltid fallet att felen blir samma, vilket är beroende på det bakomliggande sambandet och på vilka sätt det kan representeras. Med denna insikt är ett sätt att angripa svarta-lådan-problemet att skapa grenmodeller som systematiskt parar ihop parametrar på olika sätt till en output och sedan hitta modellen med lägst fel.

Fördelen med en grenmodell jämfört med ett DNN är att nätverksanalys underlättas då man kan visualisera förhållandet mellan input-datan i varje gren och dess lokala output. Detta kan göras med scatter-plots som plottar två parametrar mot varandra med det predikterade värdena som färgskala, eftersom varje neuralt delnätverk har två inputs.

3

Metod och Delresultat

Detta kapitel presenterar de undersökingar som genomförts i denna studie. Inledningsvis presenteras de hyperparametrar som användes för maskininlärningsmodellerna. Därefter presenteras de metoder som använts samt delresultat från tidiga undersökningar. För samtliga datorberäkningar användes `Python`. Maskininlärningsmodellerna konstruerades med `Python`-biblioteket `Keras` i `Tensorflow`. Samtlig träning- och testdata normaliserades var för sig med `MinMaxScaler`.

3.1 Hyperparametrar

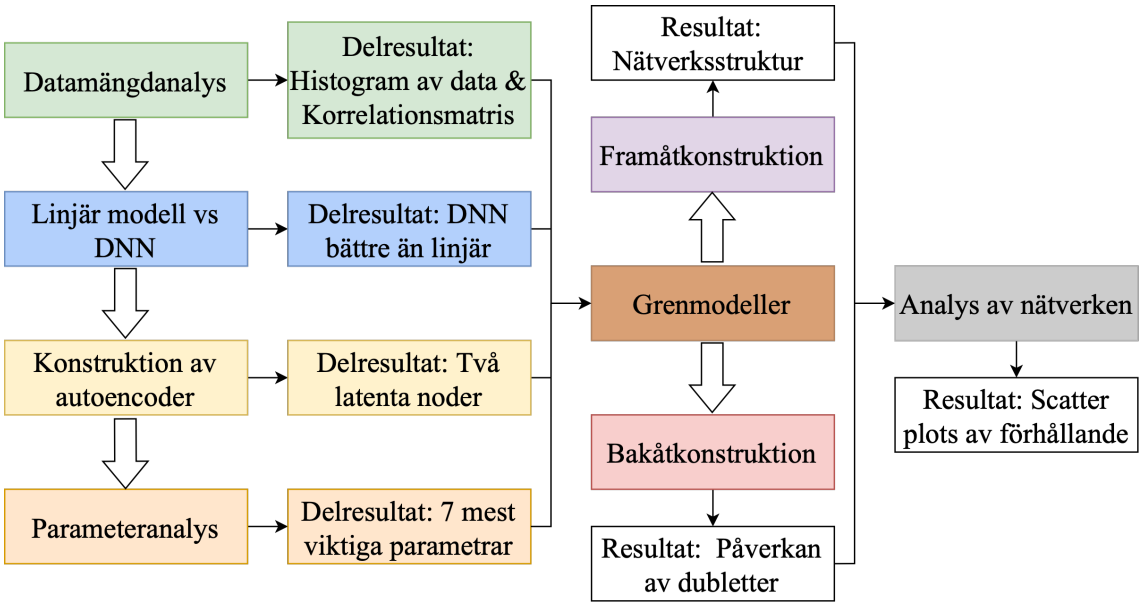
Vid konstruktion av samtliga modeller användes optimiseringsalgoritmen Adam och kostnadsfunktionen MSE. Adam är beräkningsmässigt effektiv samt lämplig för problem med mycket data. MSE valdes då den är bra på att straffa mycket avvikande värden eftersom kostnaden är kvadrerad. För validering valdes 100 000 slumpmässiga datapunkter ut från träningsdatan och de övriga 4 900 000 användes för själva träningen.

Batchstorleken, vilket representerar antalet datapunkter som observeras innan vikterna justeras, valdes till 500 och 500 000 datapunkter för undersökningen med autoencoder (se avsnitt 3.5) och 1000 datapunkter för övriga undersökningar.

Antalet träningsepoker, det vill säga hur många gånger varje individuell datapunkt används vid träning eller validering av modellen, var 1 epok för majoriteten av undersökningarna. Ett undantag gjordes i undersökningen med autoencoder i avsnitt 3.5, där både 1 epok och 150 epoker användes.

3.2 Metodupplägg

I figur 3.1 visas vilka steg som tagits under projektets gång. På grund av syftet har behovet funnits att tillämpa flera metoder som bygger på resultat från föregående metod.



Figur 3.1: Upplägg av projektets metoder och tillhörande resultat. Färgerna markerar delmetoderna samt deras tillhörande delresultat. De vita pilarna markerar den kronologiska ordningen i vilken metoderna utfördes. De svarta pilarna markerar delresultaten som producerats från delmetoderna samt de metoder som sedan baserats på dessa delresultat.

3.3 Datamängdanalys

Inledningsvis undersöktes datamängden från QuaLiKiz som bestod av tränings- och testdata på 5 respektive 3 miljoner mätningar. Datan av speciellt intresse vid analysen var *growth rate spectrum*, vilket bestod av 18 kolumner, varav de sista 9 innehöll datan för tillväxthastigheter på elektronskalorna. Träningsdatan hade 5 miljoner rader och testdatan 3 miljoner rader. Det plottades för varje k_y-index hur många procent av datan som var större än noll och maxvärdet för varje k_y-index, dessa figurer återfinns i appendix A.1. En liknande analys gjordes även för input-parametrarna, vilket bestod av 15 kolumner i gruppen *input*. En korrelationsmatris för alla parametrar skapades och histogram med fördelningen plottades för varje inputparameter. Korrelationsmatrisen och histogrammen återfinns i appendix A.2.

3.4 Jämförelse mellan en linjär modell och DNN

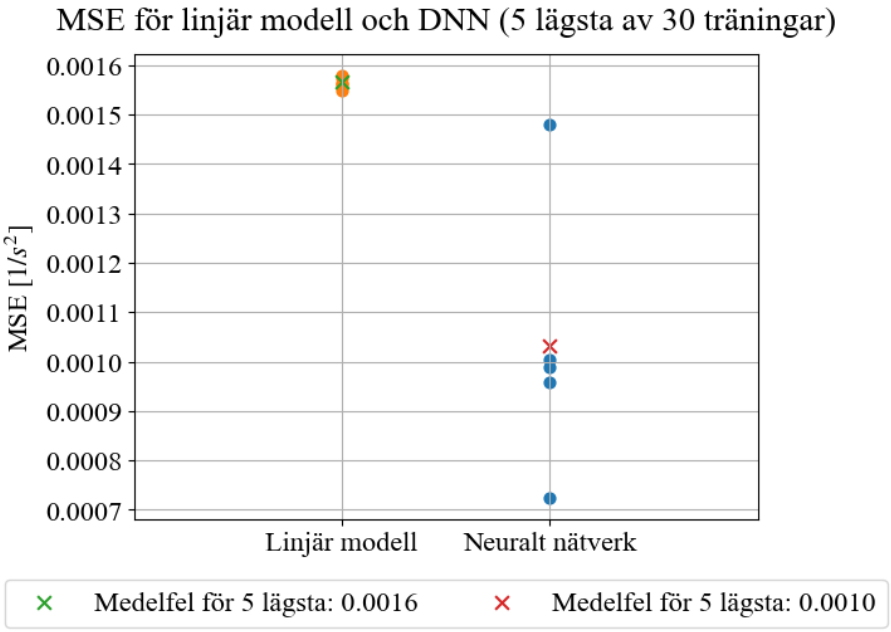
Då neurala nätverk generellt sett är mer komplexa och därav mer beräkningstunga än linjära modeller var det lämpligt att genomföra en jämförelse mellan en linjär modell

och ett neuralt nätverk. Som nämnt i avsnitt 1.3 undersöker detta arbete endast tillväxthastigheter på elektronskalorna. Därav undersöktes endast prediktionerna för k_y-index 9 till 17, vilket motsvar elektronskalor. Alla 15 inputparametrar togs med i prediktionerna. Om den linjära modellen skulle visa sig vara lika bra finns det ingen fördel med att använda ett neuralt nätverk för att prediktera instabiliteterna.

Inledningsvis skapades en linjär modell med ReLU på outputen genom att konstruera ett neuralt nätverk med endast ett lager, som då gav outputs med predikterade elektronskalor y_k på formen $y_k = f_{ReLU}(w_{k0} + w_{k1}x_1 + w_{k2}x_2 + ... + w_{k15}x_{15})$ enligt ekvation 2.7. Konstanterna w_{k0} och resterande w_{kj} motsvarar bias respektive tränade vikter, och x_j är inputparametrarna. Maskininlärning användes för den linjära modellen för att kunna applicera ReLU på outputen vilket ger en rättvis jämförelse, då exakt 0 kan predikteras lättare samt att värden under 0 är ointressanta då tillväxthastigheterna inte är negativa i QuaLiKiz-datamängden. Därefter konstruerades ytterligare ett neuralt nätverk med 2 gömda lager med 50 noder vardera. Slutligen tränades modellerna för att prediktera outputen med k_y-index 9-17 och MSE från testdatan mättes. Detta genomfördes 30 gånger och medelfelet från de 5 mätningarna med lägst fel noterades.

3.4.1 Delresultat från jämförelse

Jämförelsen mellan en linjär modell och ett DNN gav delresultaten i figur 3.2. Figuren visar de 5 bästa mätpunkterna från 30 träningar av en linjär modell och ett DNN. Resultaten tyder på att en linjär modell inte kan minimera felet till mindre än 0,0015 medan det är möjligt för ett DNN att minimera felet till mindre än 0,001. Detta indikerar att ett DNN är en bättre modell för att prediktera instabiliteterna i plasmat på elektronskalan jämfört med en linjär modell.



Figur 3.2: De 5 lägsta felen från 30 träningar av ett DNN och linjär modell.

3.5 Konstruktion av autoencoder

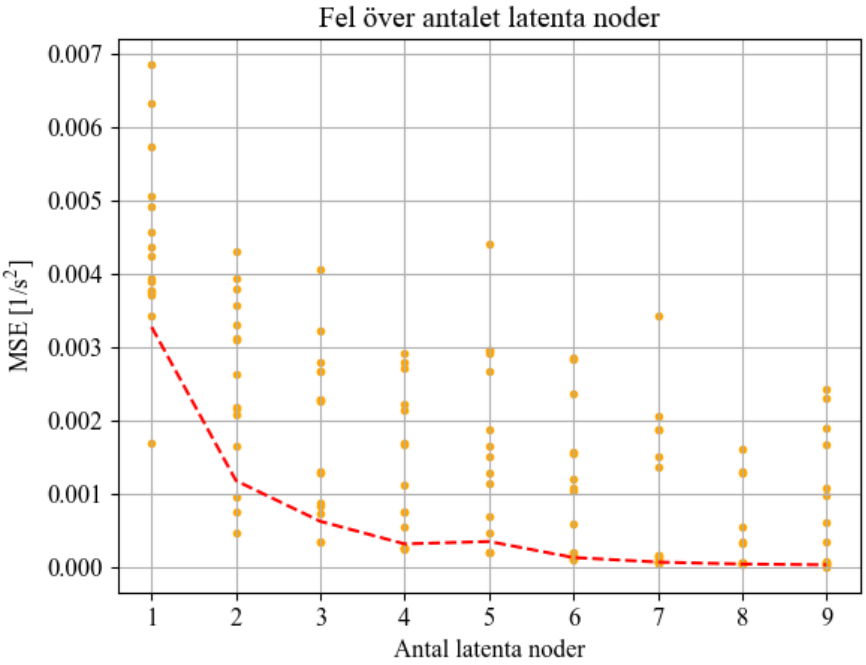
Några olika neurala nätverk i form av autoencoders skapades för att undersöka hur många latenta noder som krävdes för att kunna återskapa träningsdatan för *growth rate spectrum* med godtagbart fel. Testet användes för att ge en indikation om hur mycket det går att komprimera tillväxthastigheterna men ändå bibehålla den viktigaste informationen. Detta var särskilt viktigt för att veta vilken arkitektur neurala nätverk i senare undersökningar skulle ha (grenmodeller i avsnitt 3.7), då man inte kan använda färre grenar än minsta antalet möjliga latenta noder.

Initialt byggdes olika autoencoders utan gömda lager med 1-9 latenta noder. Hyperparametrarna som valdes var batchstorleken 500 och 1 träningsepok vilket ger 9800 träningsiterationer, alltså hur många gånger vikterna uppdateras. Därefter plottades valideringsfelet som funktion av antalet latenta noder för att kunna bestämma vilket antal som skulle användas för att utföra senare delar i projektet. Den optimala antalet latenta noder skulle vara så få som möjligt utan att felet blev för stort. För att minska effekten av slumpmässiga variationer genomfördes 15 träningar av autoencodern för varje antal latenta noder. Slutligen beräknades medelvärdet av MSE på de 5 träningarna med lägst fel.

Valideringsfelet plottades även mot antalet träningsepoker för att säkerställa att antalet träningsiterationer var tillräckligt för att modellen ska hinna minimera felet. Här användes hyperparametrarna: 500 000 som batchstorlek och 150 träningsepoker, vilket innebär 1470 träningsiterationer.

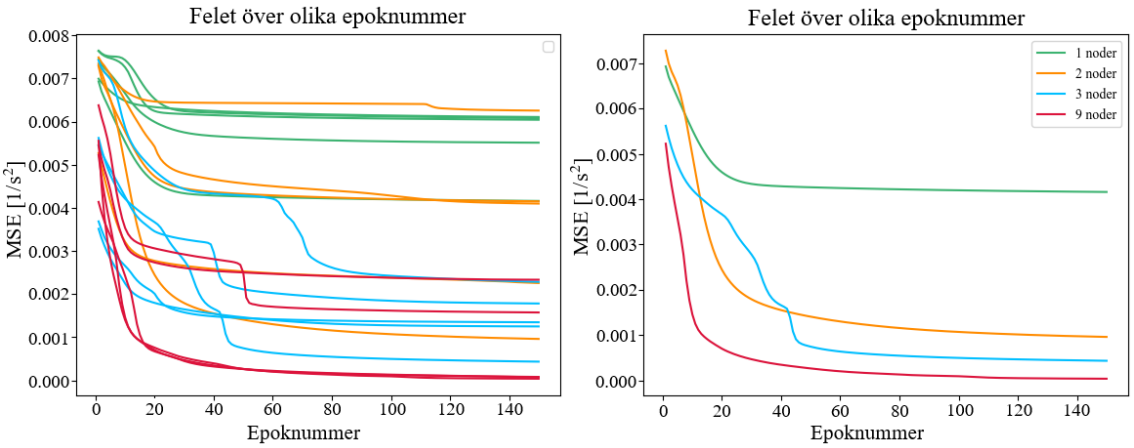
3.5.1 Delresultat från autoencoder

Valideringsfelet för olika antal latenta noder redovisas i figur 3.3 och tyder på att i endast 2 noder skulle behövas i det latenta lagret. Även om medelvärdet är något lägre för 3 noder verkar de bästa mätningarna tillräckligt lika för att 2 noder ska anses vara ett likvärdigt alternativ. Undersökningen genomfördes även för utdata med samtliga 18 k_y-index, det vill säga instabiliteterna vid både jon- och elektronskalan. Resultaten från denna undersökning återfinns i appendix B. Resultaten från undersökningarna med autoencoder indikerar att datan i *growth rate spectrum* kan reduceras från 18 (samtliga skalor) och 9 (elektronskalor) dimensioner till 3 respektive 2 dimensioner. Det finns alltså en lågdimensionell representation av datan, vilket är en förutsättning för att metoden för grenmodellen (se avsnitt 3.7) ska ge intressanta resultat. Detta eftersom det inte skulle vara möjligt att para ihop parametrar för reducering om det inte existerade en lågdimensionell representation.



Figur 3.3: Valideringsfelet efter träning för olika antal noder i det latenta lagret där varje mätning har genomförts 15 gånger. Orangea punkter visar MSE från varje mätning. Den streckade linjen representerar medelvärdet av de 5 bästa mätningarna för varje nodantal.

Denna slutsats finner ytterligare stöd i figur 3.4, där utvecklingen av valideringsfelet över träningsprocessens epoker redovisas. Även om 3 latenta noder ger ett mindre fel än 2 anses skillnaden vara försumbart liten. Felet verkar även stabilisera sig vilket indikerar på att 1 470 träningsiterationer räcker för att minimera felet. Detta tyder på att modellen som användes för resultaten i figur 3.3, som genomförde 9 800 träningsiterationer också hann minimera sitt fel. Båda valen av hyperparametrar (batchstorlek 500 000 och 500) ger ungefär samma MSE vilket indikerar att modellerna tränats lika bra.



(a) (b)

Figur 3.4: Valideringsfelet över träningens epoker för olika antal latenta noder. Träningen utgjordes av 1470 träningsiterationer. Mätningen genomfördes 5 gånger. Figur a visar samtliga mätningar och figur b visar det lägsta valideringsfelet för varje nod. Uppifrån och ner är antalet noder 1 (grön), 2 (orange), 3 (blå), samt 9 (röd).

3.6 Relevans hos parametrar

Denna undersökning fokuserar på k_y-värderna 9-17, alltså instabiliteter på elektronskalan. Detta innebär att vissa plasmaparametrar kan ha minimal påverkan på resultatet, då de har större påverkan på jonskalan. För att minska tiden för träning av nätverken, genom att endast fokusera på de parametrar som påverkar tillväxthastigheterna på elektronskalan vid senare mätningar, genomfördes en undersökning för att se vilka parametrar som kan försummas.

Inlednigsvis togs en parameter bort från inputdatan. Datan användes sedan för att träna ett neuralt nätverk med samma struktur som det neurala nätverket i avsnitt 3.4 (2 gömda lager med 50 noder vardera). Efter träning mättes MSE hos modellen med testdatan. Detta upprepades 15 gånger och medelvärdet på felet från de 5 mätningarna med lägst fel beräknades. Därefter genomfördes samma mätning då en annan parameter togs bort på samma sätt. Efter att samtliga 15 parametrar genomgått denna mätning erhölls den parameter som gav lägst medelfel vid borttagning. Denna parameter togs bort helt från inputdatan då den ansågs vara minst viktig för att prediktera tillväxthastigheterna. Därefter genomfördes samma process igen med den nya inputdatan med de resterande 14 parametrarna. Återigen togs den parameter som gav lägst medelfel bort från datan. Processen upprepades tills dess att endast 1 parameter återstod i inputdatan.

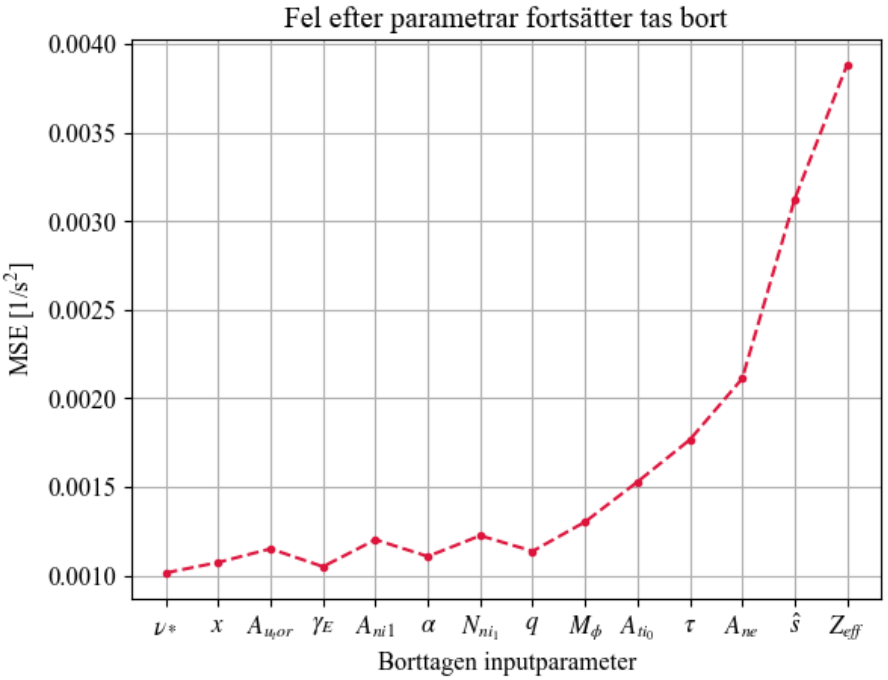
Denna totala mätning gjordes totalt tre gånger och resultatet analyserades för att kunna dra en slutsats om vilka parametrar som konsekvent var kvar i slutet samt vilka som togs bort tidigt.

Slutligen gjordes ytterligare mätningar på hur felet hos det neurala nätverket såg ut med endast de inputparametrar som ansågs vara potentiellt relevanta. Valet av dessa parametrar baserades på resultaten från de tre tidigare mätningarna. MSE på testdatan beräknades 30 gånger och medelvärdet av felet från de 5 bästa mätningarna beräknades. Medelfelet jämfördes med felet från samtliga inputparametrar för att se hur stor förändringen i felet var.

Efter att den bästa kombinationen av parametrar fastställts användes medelfelet för detta neurala nätverket som ett riktmärke att jämföra med i senare undersökningar, se avsnitt 3.7.

3.6.1 Delresultat för relevans av parametrar

I figur 3.5 redovisas resultatet från en av de tre mätningarna som genomfördes för att se relevansen hos de olika parametrarna. Denna mätning indikerar att A_{te}, Z_{eff}, $\hat{s}$, A_{ne}, τ samt A_{ti0} är de viktiga parametrarna. Detta stämmer bra överens med resultatet från de övriga mätningarna förutom att de indikerar att parametrar så som q och ν_* också är viktiga. Resultaten från övriga två mätningar återfinns i appendix C.



Figur 3.5: MSE av de 5 bästa mätningarna av totalt 15 där den utskrivna parametern har tagits bort. Detta innebär att alla parametrar till vänster om den utskrivna, inklusive denna, är borttagna. MSE är då taget ifrån ett neuralt nätverk som bygger på alla parametrar till höger om den utskrivna. Den sista punkten är MSE från ett nätverk med endast en parameter kvar, A_{te}, som är med i samtliga mätningar.

Eftersom A_{te}, Z_{eff} och $\hat{s}$ var de tre sista parametrarna som togs bort i samtliga mätningar misstänks dessa vara viktigast för att det neurala nätverket ska kunna prediktera tillväxthastigheten. Med 8 olika potentiellt viktiga parametrar genomfördes ett flertal olika mätningar för att bestämma felet på de olika konstellationerna. Resultaten från dessa redovisas i tabell 3.1. Att ta bort parametern A_{ti0} försämrar inte felet, så den anses därför inte längre vara viktig. Då parametrarna A_{ne} och ν_* tas bort, både var för sig och att båda tas bort samtidigt, ökar felet mer markant.

Tabell 3.1: MSE hos det neurala nätverket med olika inputparametrar. Värde är medelvärdet av 5 bästa mätningar av 30 totalt.

Inputparametrar	MSE $[10^{-3}]$
Samtliga 15 parametrar	1,20
A_{ne}, A_{te}, A_{ti0}, ν_*, q, $\hat{s}$, τ, Z_{eff}	0,95
A_{ne}, A_{te}, ν_*, q, $\hat{s}$, τ, Z_{eff}	0,85
A_{ne}, A_{te}, q, $\hat{s}$, τ, Z_{eff}	1,25
A_{te}, ν_*, q, $\hat{s}$, τ, Z_{eff}	1,24
A_{te}, q, $\hat{s}$, τ, Z_{eff}	1,32

Det slutgiltiga resultatet visar därmed att de viktiga parametrarna bör vara A_{ne}, A_{te}, ν_*, q, $\hat{s}$, τ och Z_{eff}. Riktmärket blir därmed 0,00085 för felet som kan uppnås med ett vanligt DNN. Att elektrondensitetgradienten A_{ne}, elektrontemperaturgradienten A_{te} och jon-elektrontemperatur-ration τ visade sig vara viktiga är rimligt då modellen predikterar på elektronskalorna. Ingen av de slutgiltiga parametrarna är specifika för joner enbart. Om det finns parametrar som fysikaliskt borde vara essentiella att ha med, kan en möjlig förklaring till att de inte var nödvändiga för modellen vara att det finns en stark korrelation till inkluderade parametrar. Exempelvis är korrelationen 0,81 mellan inkluderade $\hat{s}$ och ej inkluderade x, se korrelationsmatris i appendix A.2.

Anledningen till varför ett nätverk med färre antal parametrar kan ge ett lägre fel än ett nätverk med fler parametrar kan möjligtvis bero på att optimeringsfunktionen i det senare fallet tar hänsyn till renodlade jonparametrar, exempelvis A_{ti0}, som inte borde påverka från ett fysikaliskt perspektiv. Detta leder då till att träningsiterationer ödslas på att optimera vikter till parametrar som inte bidrar till ett lägre fel. Det blir därmed effektivt färre träningsiterationer i fallet med samtliga parametrar, vilket hade kunnat förklara varför ett lägre fel inte uppnås.

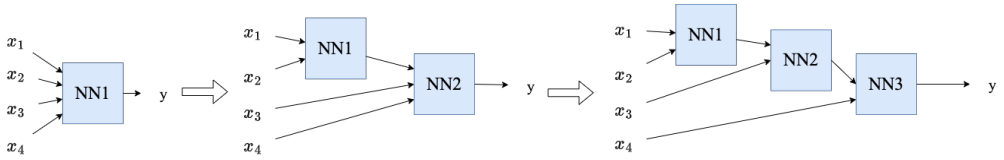
3.7 Grenmodeller

Med vetskap om hur många latenta noder som tillväxthastigheterna kunde reduceras till via autoencoder-testet erhölls ett miniminummer på hur många nyckelparametrar som krävdes för att beskriva outputen. För elektronskalorna kunde tillväxthastigheterna beskrivas med två latenta noder baserat på resultatet om antalet latenta noder, se figur 3.3 i avsnitt 3.5.1. För att undersöka *hur* ett neuralt nätverk predikterar

instabiliteterna undersöktes därmed hur parametrarna kan kombineras ihop till endast två reducerade parametrar som sedan kan användas för att prediktera outputen. Detta gjordes genom att skapa och jämföra grenmodeller bestående av förgreningar av flera neurala nätverk i olika konfigurationer genom två olika tillvägagångssätt: framåt- och bakåtkonstruktion.

3.7.1 Framåtkonstruktion

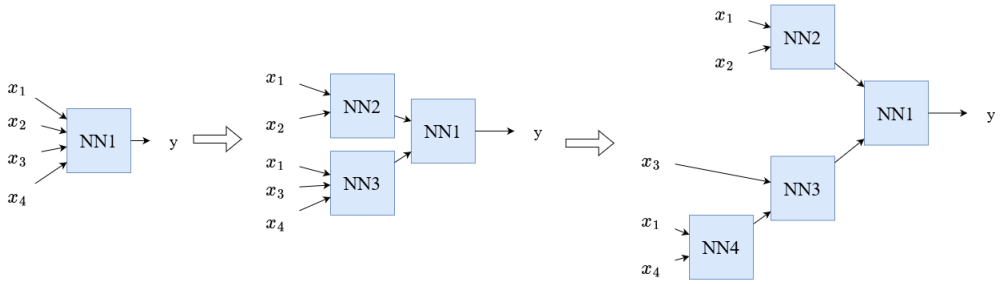
Principen för processen visas i figur 3.6. Inledningsvis skapades ett neuralt nätverk som använde de inputparametrar som visade sig vara relevanta för att beskriva tillväxthastigheterna på elektronskalan. Två av parametrarna parades ihop till en gren i nätverket och felet beräknades som medelvärdet av de 5 lägsta MSE av totalt 15 mätningar. Detta gjordes för samtliga kombinationer av ihopparningar. Den ihopparning som resulterade i lägsta fel låstes och den predikterade outputen från dess neurala nätverk sparades. Denna predikterade output ersatte de ihopparade parametrarna för att skapa en ny inputdata bestående av en färre parameter. Denna process upprepades sedan med den nya inputdatan tills alla inputparametrar hade reducerats till två latenta noder.



Figur 3.6: Stegvis framåtkonstruktion av grenmodell för reducering av parametrar. Parametrarna fördelas till olika neurala nätverk som sammankopplas för att prediktera en output.

3.7.2 Bakåtkonstruktion

Då föregående konstruktion bygger nätverket "framlänges" genom att fatta beslut om grenarnas konfiguration utifrån vilka ihopparningar av parametrar som ger lägst fel, så bygger följande konstruktion upp nätverket "baklänges" genom att kontinuerligt dela upp parametrarna i varje gren i två nya grenar. Med denna konstruktion kan det, till skillnad från med framåtkonstruktionen, tas hänsyn till dubbletter, alltså att parametrar kan förekomma i flera grenar. Förekomsten av parameterdubbletter är en möjlighet då varje fysikalisk storhet kan förekomma fler gånger i lösningen till differentialekvationerna som QuaLiKiz bygger på. Ett exempel på denna konstruktion visas i figur 3.7.



Figur 3.7: Stegvis bakåtkonstruktion av grenmodell för reducering av parametrar. Parametrarna delas upp till olika, sammankopplade, neurala nätverk.

3.7.3 Dubbletter i konstruktionen

Som ett första steg i bakåtmodellen analyserades hur stor påverkan dubbletter av parametrar har på grenmodellen, det vill säga när en eller flera parametrar går in i mer än en gren. Inputparametrarna delades upp i två grupper för att sedan användas för träning av ett delnätverk. Storleken på en grupp fick inte överstiga $n-1$ parametrar, där n är totala antalet unika inputparametrar. Detta eftersom om en grupp innehåller n parametrar är det inte längre en uppdelning. För $n = 7$ finns tre olika uppdelningar när dubbletter inte tillåts:

- 1 parameter och 6 parametrar
- 2 parametrar och 5 parametrar
- 3 parametrar och 4 parametrar

När man introducerar dubbletter ändras uppdelningarna. Beroende på hur många dubbletter som tillåts finns det olika antal kombinationer av parametrar som ingår i de olika grupperna. Totala antalet kombinationer blir som följande för $n = 7$:

Antal dubbletter	Antal kombinationer
0	63
1	217
2	315
3	245
4	105
5	21

Varje kombination undersöktes 15 gånger där medelvärdet av de 5 lägsta MSE sedan användes för att jämföra uppdelningar. Därefter introducerades en dubblett, vilket ökade antalet möjliga kombinationer, varvid undersökningen upprepades. Detta gjordes upp till totalt $n-2$ dubbletter. Slutligen jämfördes medelvärdet av MSE för den bästa uppdelningen för varje antal dubbletter för att se ifall fler dubbletter minskade felet tillräckligt mycket för att rättfärdiga en ökad komplexitet.

3.7.4 Analys av grennätverken

När en slutgiltig konstruktion för framåt- och bakåtmodellen hade upptäckts tränades nätverken inledningsvis 15 gånger och det lägsta felet från träningarna noterades. Därefter genomfördes nya träningar tills att ett nätverk nådde ett lika lågt fel som det lägsta felet från de 15 första träningarna. Genom detta nätverk analyserades relationen mellan inputparametrar och det predikterade värdet från de neurala nätverken som direkt tog in inputparametrarna. Detta gjordes med hjälp av scatterplots. Förhållandet mellan grenarnas inputs och output blev då tolkningsbart, till skillnad från DNN-modellen.

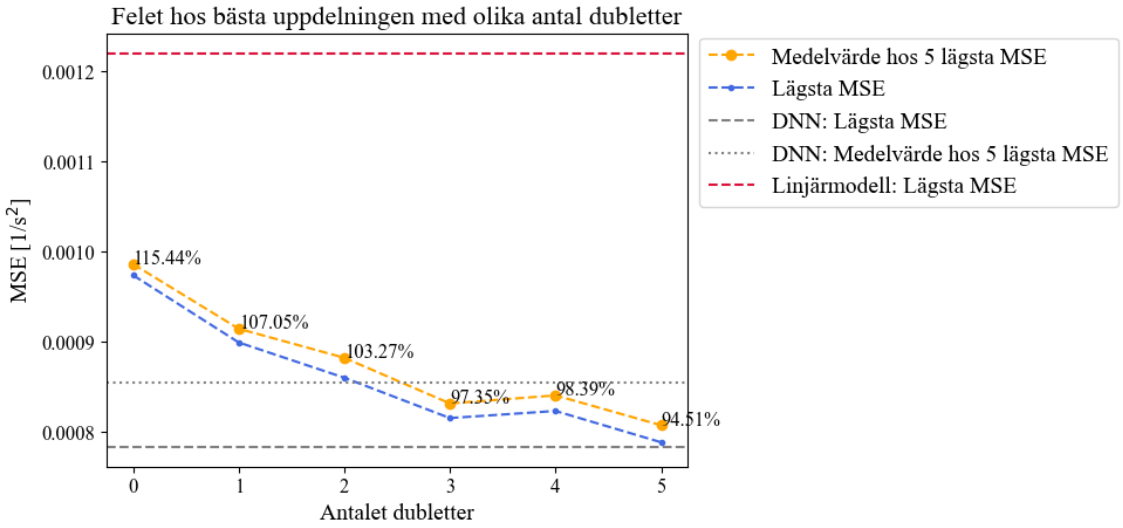
Notera att samtlig data samlades in från samma träning av grenmodellen eftersom det kan finnas variationer mellan olika träningar, även om det är samma grenmodell.

4

Slutresultat och Diskussion

4.1 Dubbletter hos inputparametrar

Felet från första steget i en bakåtkonstruktion med olika antal parametrar som tillåts gå in i flera delnätverk, s.k. dubletter, visas i figur 4.1. Notera att skillnaden mellan 0 dubbletter och 5 dubbletter är cirka $0,000198$ (minskning på $16,8\%$).



Figur 4.1: MSE hos den bästa uppdelningen med olika antal dubbletter, där andelen av riktmärkesvärdet (den prickade linjen) är utskrivet. De gråa linjerna är riktmärken från tidigare resultat från ett DNN. Den röda linjen är riktmärke från en linjär modell.

Resultaten tyder på att användning av dubbletter skulle ge en mer korrekt beskrivning av hur inputparametrarna relaterar till output, samt att generellt ger fler dubbletter ett lägre fel. Värt att notera är dock att fler dubbletter introducerar en högre komplexitet och försvårar tolkningen av hur inputparametrarna påverkar output. I detta arbete anses minskningen i felet inte vara tillräckligt stort för att rättfärdiga den ökade komplexiteten. Felet för noll dubbletter är fortfarande lågt och därmed kommer framtida mätningar ej behandla dubbletter.

Enligt resultaten som baseras på medelvärdet från de 5 bästa mätningarna av 15 stycken verkar fler dubbletter leda till att modellen når ett lägre fel med högre sannolikhet än ett vanligt DNN. En möjlig förklaring till detta är att ett DNN inte tvingas till en viss konstellation. Detta skulle kunna leda till att sannolikheten att ett DNN hamnar i ett lokalt minimum är högre än för en grenmodell som tvingas ha flera dubbletter.

4.2 Framåtkonstruktion

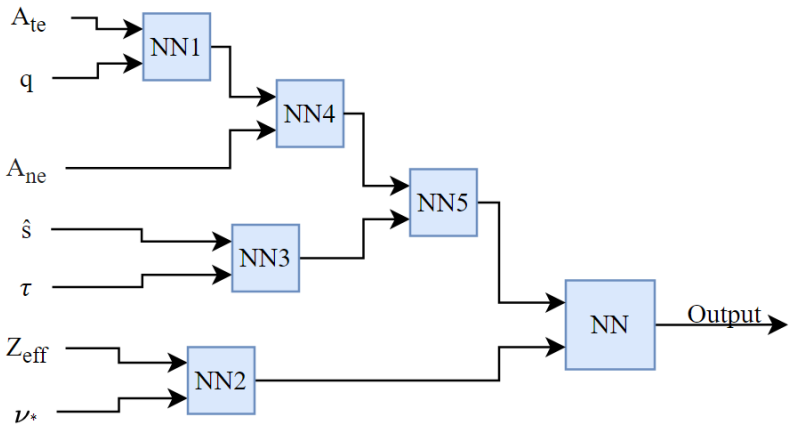
Figur 4.2 visar grenmodellen som skapades med hjälp av framåtkonstruktion, där ihopparningarna gjordes i ordningen NN1 till NN5. Denna konstruktion kan beskrivas matematiskt enligt:

$$output = NN(z_2, z_5)$$

$$\begin{aligned} z_1 &= NN1(A_{te}, q) \\ z_2 &= NN2(Z_{eff}, \nu_*) \\ z_3 &= NN3(\hat{s}, \tau) \end{aligned} \qquad \begin{aligned} z_4 &= NN4(z_1, A_{ne}) \\ z_5 &= NN5(z_4, z_3) \end{aligned} \tag{4.1}$$

Värt att notera att i de flesta ihopparningarna fanns det alternativa ihopparningar som gav ett fel som i princip var lika lågt som den valda ihopparningen. Detta tyder på att det finns flera matematiska modeller för att beskriva samma output vilket är rimligt då en matematisk ekvation kan formuleras på olika sätt (se figur 2.5 i avsnitt 2.4.2).

Notera även att parametrarna A_{te}, Z_{eff} och $\hat{s}$, alltså de som ansågs vara viktigast i avsnitt 3.6.1, är separerade så länge som möjligt. A_{te} och $\hat{s}$ paras ihop i det sista steget (NN5).

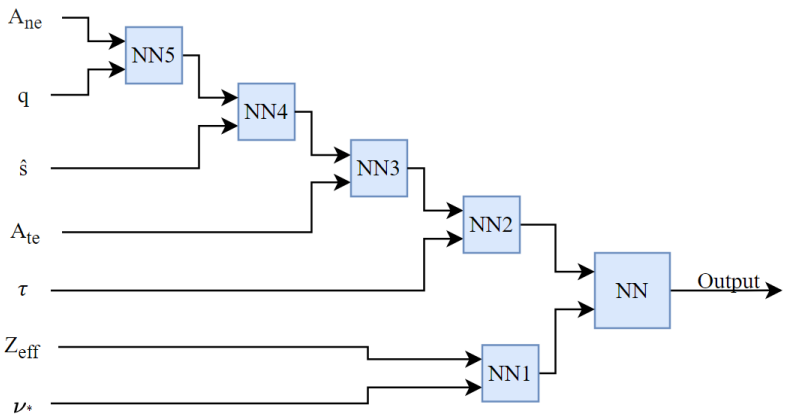


Figur 4.2: Den slutgiltiga konstruktionen av grenmodellen från framåtkonstruktionen.

4.3 Bakåtkonstruktion

Utan användning av dubbletter är uppdelningen som ger lägst fel att ha Z_{eff} och ν_* till en gren och att ha A_{ne}, A_{te}, q, $\hat{s}$ och τ till den andra. Detta stämmer överens med resultatet från framåtkonstruktionen (se figur 4.2). Däremot skiljer sig den slutgiltiga konstruktionen i hur A_{ne}, A_{te}, q, $\hat{s}$ och τ reduceras ner till en av de två nyckelparametrarna. Exempelvis slås A_{te} och $\hat{s}$ ihop tidigare än vad de gör i framåtmodellen. Figur 4.3 visar modellen från bakåtkonstruktionen (där ihopparningarna gjordes i ordningen NN1 till NN5) och kan beskrivas matematiskt enligt:

$$
\begin{aligned}
output &= NN(z_2, z_1) \\
z_1 &= NN1(Z_{eff}, \nu_*) \qquad & z_4 &= NN4(z_5, \hat{s}) \\
z_5 &= NN5(A_{ne}, q) & z_3 &= NN3(z_4, A_{te}) \\
& & z_2 &= NN2(z_3, \tau)
\end{aligned}
\tag{4.2}
$$



Figur 4.3: Den slutgiltiga konstruktionen av grenmodellen från bakåtkonstruktionen.

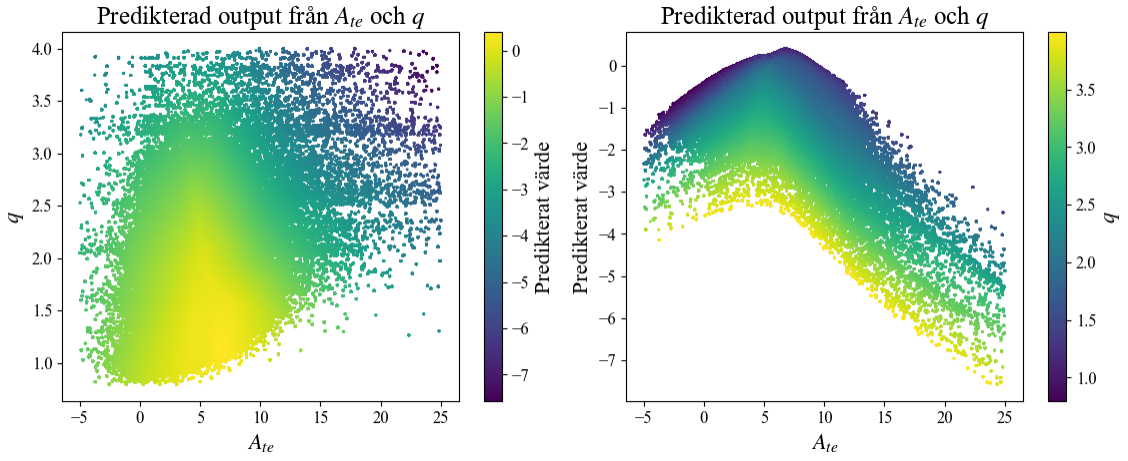
Även värt att notera att liknande framåtkonstruktionen så fanns det alternativa ihopparningar som gav ett fel som var nästintill lika lågt som för de valda ihopparningarna. Precis som för framåtkonstruktionen tyder detta ytterligare på att det finns flera matematiska modeller som kan beskriva output.

4.4 Korrelationer inom de neurala nätverken

I figurerna 4.4, 4.5 och 4.6 redovisas hur inputparametrarna i de neurala nätverken NN1, NN2 respektive NN3 från framåtkonstruktionen i figur 4.2 påverkar delnätverkets output (se appendix D.1 för scatterplots för resterande nätverk). I figurerna 4.7 och 4.8 redovisas nätverken NN1 och NN5 från bakåtkonstruktionen i figur 4.3 (se appendix D.2 för plots för resterande nätverk). Eftersom nätverket kan hamna i olika lokala minimum, trots att de har samma fel, kan sambandet variera något mellan olika träningar.

4.4.1 Framåtkonstruktion

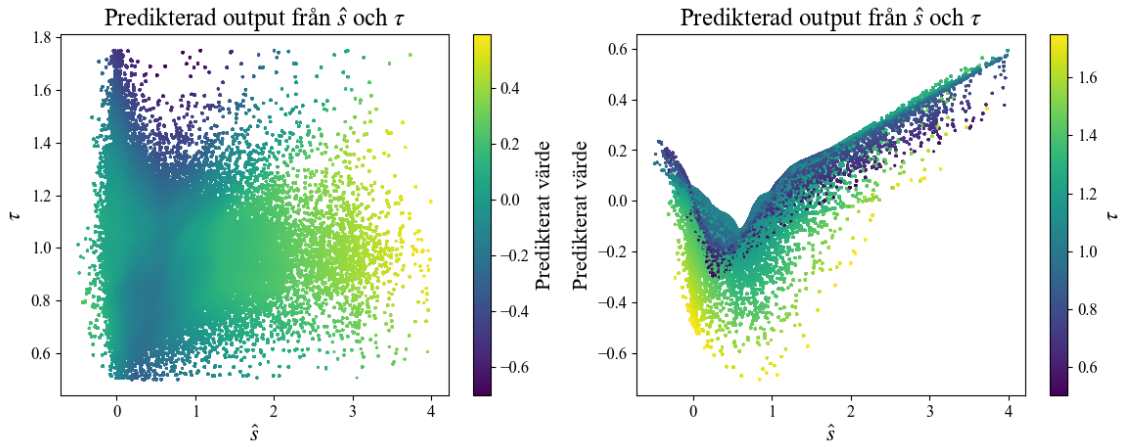
I figur 4.4 visas sambanden från NN1 i grenmodellen från framåtkonstruktionen (se figur 4.2). I NN1 verkar A_{te} ha ett styckvis linjärt samband med det predikterade värdet. Vid värden högre än $A_{te} \approx 5$ ändras lutningen på det linjära sambandet från positivt till negativt (se figur 4.4b). Värdet på q verkar ha en direktkorrelation med det predikterade värdet, eftersom då q ökar minskar det predikterade värdet.



(a) **(b)**

Figur 4.4: Relationen mellan inputparametrarna och det predikterade värdet från NN1 i grenmodellen i figur 4.2. Figur a visar A_{te} och q plottade mot varandra. Figur b visar A_{te} plottad mot det predikterade värdet.

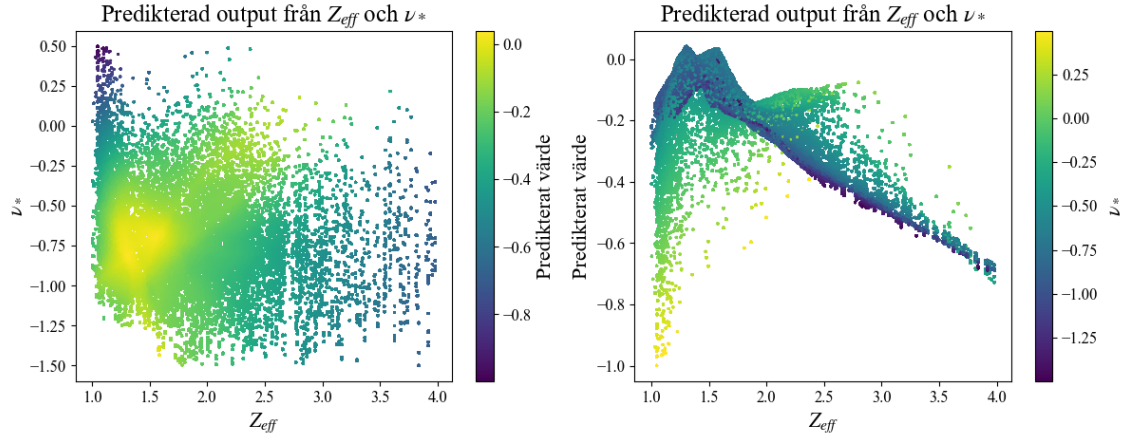
Figur 4.5 visar NN2 från grenmodellen i figur 4.2. Sambandet mellan $\hat{s}$ och det predikterade värdet kan även här klassas som styckvis linjärt och delas in i två regioner. Var gränsen mellan dessa är varierar med värdet på τ och är cirka $\hat{s} \approx 0,25$ för låga τ och $\hat{s} \approx 0,75$ för höga τ. Lutningen mellan de olika regionerna ändras från negativt till positivt och hur skarpa dessa lutningar är verkar bero på τ. Alltså vid höga τ har värdet på $\hat{s}$ en större påverkan på systemet.



(a) **(b)**

Figur 4.5: Relationen mellan inputparametrarna och det predikterade värdet från NN2 i grenmodellen i figur 4.2. Figur a visar $\hat{s}$ och τ plottade mot varandra. Figur b visar $\hat{s}$ plottad mot det predikterade värdet.

Slutligen visas NN3 från framåtmodellen i figur 4.6. Det predikterade värdet verkar uppnå ett maximum vid $\nu_* \approx -0,75$ och $Z_{eff} \approx 1,5$ (se figur 4.6a). Att röra sig från denna punkt minskar det predikterade värdet, däremot varierar lutningen beroende på riktning. Att hålla ν_* eller Z_{eff} konstant medan den andra ökar resulterar i en relativt brant lutning på minskningen hos det predikterade värdet. Att däremot öka båda resulterar i en relativt stagnant lutning där det predikterade värdet knappt ändras.

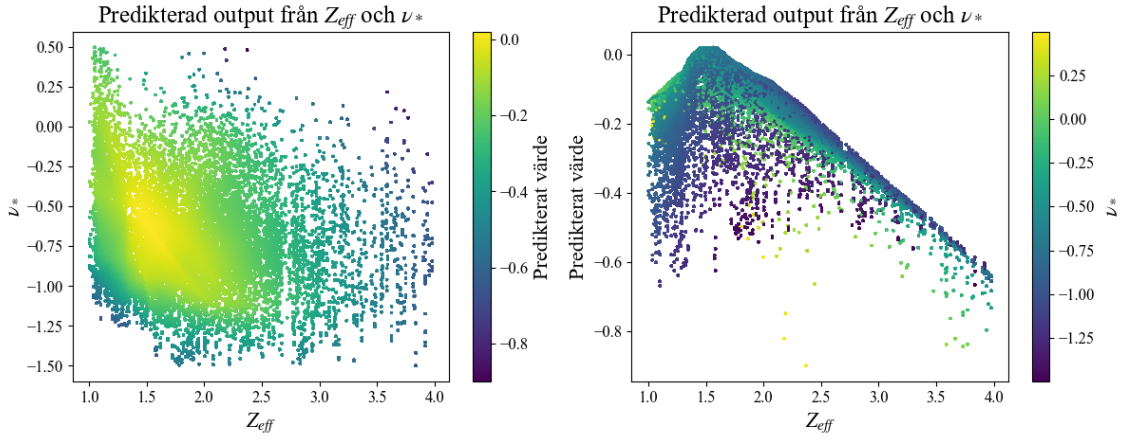


(a) **(b)**

Figur 4.6: Relationen mellan inputparametrarna och det predikterade värdet från NN3 i grenmodellen i figur 4.2. Figur a visar Z_{eff} och ν_* plottade mot varandra. Figur b visar Z_{eff} plottad mot det predikterade värdet.

4.4.2 Bakåtkonstruktion

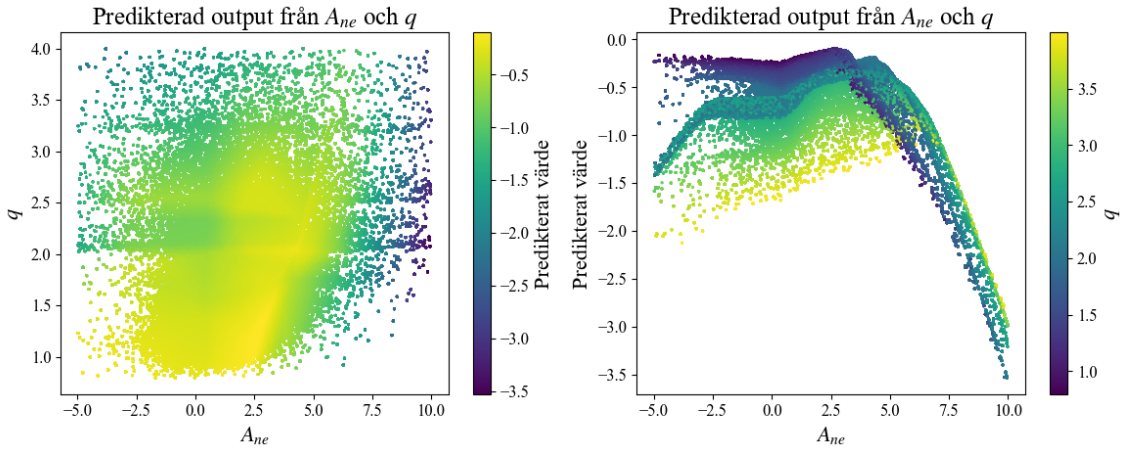
I figur 4.7 visas sambanden från NN1 i bakåtkonstruktionen (se figur 4.3). Denna ihopparning är den som överlappar med framåtkonstruktionen (se figur 4.6). Även om korrelationen från framåtkonstruktionen inte är identisk med bakåtkonstruktionen visar båda konstruktionerna ett liknande samband. Vid $\nu_* \approx -0,75$ och $Z_{eff} \approx 1,5$ antar outputen ett maximumvärde som sedan avtar runt omkring denna punkt. Till skillnad från framåtkonstruktionen verkar dock figur 4.7a tyda på att hålla konstant Z_{eff} medan ν_* ökar ger långsammast minskning av det predikterade värdet.



(a) **(b)**

Figur 4.7: Relationen mellan inputparametrarna och det predikterade värdet från NN1 i grenmodellen i figur 4.3. Figur a visar Z_{eff} och ν_* plottade mot varandra. Figur b visar Z_{eff} plottad mot det predikterade värdet.

Figur 4.8 visar NN5 från bakåtkonstruktionen i figur 4.3. Figur 4.8b kan delas in i två regioner där gränsen mellan dessa verkar bero på värdet på q. Vid låga q ligger gränsen på ungefär $A_{ne} \approx 3$ och vid höga q ligger gränsen på ungefär $A_{ne} \approx 7$. Då A_{ne} antar ett värde som är större än detta har A_{ne} ett tydligt linjärt samband med det predikterade värdet. I denna region så ökar även det predikterade värdet när värdet på q ökar. Om A_{ne} antar ett värde som är lägre än värdet på gränsen mellan regionerna blir grafen mer svårtolkad. Däremot verkar korrelationen fortfarande vara linjär, men har en positiv lutning till skillnad från den andra regionen. Även här verkar q påverka lutningen på grafen. Till skillnad från den andra regionen minskar det predikterade värdet om q ökar. Alltså vid låga q har värdet på A_{ne} en större påverkan på systemet.



(a) **(b)**

Figur 4.8: Relationen mellan inputparametrarna och det predikterade värdet från NN5 i grenmodellen i figur 4.3. Figur a visar A_{ne} och q plottade mot varandra. Figur b visar A_{ne} plottad mot det predikterade värdet.

5

Slutsats

Projektet har undersökt hur neurala nätverk predikterar QuaLiKiz beräkningar på tillväxthastigheter för instabiliteter i fusionsplasma. Detta gjordes genom att undersöka vilka inputparametrar som är viktiga för att beskriva tillväxthastigheterna samt genom att använda grenmodeller med både framåt- och bakåtkonstruktion. Resultaten tyder på att endast parametrarna A_{ne}, A_{te}, ν_*, q, $\hat{s}$, τ och Z_{eff} krävs för att prediktera tillväxthastigheterna på elektronskalan. Att endast använda dessa parametrar vid framtida beräkningar skulle kunna snabba upp processen eftersom en maskininlärningsmodell då inte behöver göra onödiga optimeringar. Dessutom visar resultaten att det möjligtvis existerar en enklare matematisk modell som beskriver tillväxthastigheterna än de differentialekvationer som används i QuaLiKiz.

Resultaten från grenmodellerna visar att det finns konstruktioner som ger bättre fel än många andra varianter, vilket är intressant av två anledningar: Dels fås information om hur parametrarna är relaterade till varandra, de parametrar som inte fungerar väl tillsammans i en gren innehåller antagligen mycket information som är viktig för fysiken. Speciellt intressant är att parametrarna Z_{eff} (effektivt atomnummer i plasmat) och ν_* (kollisionalitet) verkar gå att kombinera till en av de två latenta noderna som datan för tillväxthastigheten kan beskrivas ifrån. Dels har användandet av grenmodeller i detta projekt indikerat att det är en fungerande metod för att närma sig svarta-lådan-problemet.

5.1 Samhälls- och etiska aspekter

I en tid med växande användning av maskininlärning inom alla möjliga fält är det relevant att reflektera över såväl samhälleliga konsekvenser som etiska aspekter. Eftersom syftet med projektet är att använda maskininlärning för forskning om en förnybar energikälla anses inte konsekvenserna av dåligt fungerande modeller vara lika allvarliga som om de skulle användas för att exempelvis fastställa en dom eller att anställa personer. Däremot är det viktigt att ha i åtanke att modeller som är avsedda för ett visst syfte, kan användas som grund för något helt annat. Exempelvis skulle de framsteg som gjorts under detta projekt kunna bidra till utvecklingen av fler tolkbara modeller inom andra, mer etiskt kritiska fält.

5.2 Framtida projekt

Möjliga fortsättningar på projektet hade varit att mer utförligt undersöka relationerna mellan inputparametrarna i syfte att undersöka om det finns en enklare matematisk modell som kan användas för beräkning av instabiliteterna. Detta kan förslagsvis dels göras genom att ytterligare studera scatter-plotsen samt att ytterligare undersöka grenmodeller med fler dubbletter. Utöver detta skulle även den näst starkaste instabiliteten från QuaLiKiz kunna undersökas. Det hade även kunnat vara givande att studera både elektronskalorna och jonskalorna tillsammans istället för att bara undersöka en skala i taget.

Referenser

[1] B. Wanner och R. Taniguchi, *Nuclear energy*. 2023 [Online]. URL: https://www.iea.org/energy-system/electricity/nuclear-power (hämtad 2024-01-24).

[2] Strålsäkerhetsmyndigheten, *Kärnkraftsolyckor i världen*. 2022 [Online]. URL: https://www.stralsakerhetsmyndigheten.se/omraden/karnkraft/%5C%5Ckarnkraftsolyckor-i-varlden/ (hämtad 2024-01-26).

[3] P. Daya, *What is Deuterium?* 2023 [Online]. URL: https://www.iaea.org/newscenter/news/what-is-deuterium (hämtad 2024-02-01).

[4] United States Nuclear Regulatory Comission, *Frequently asked Questions About Liquid Radioactive Releases*. 2021 [Online]. URL: https://www.nrc.gov/reactors/operating/ops-experience/tritium/faqs.html (hämtad 2024-04-18).

[5] FusionWiki, *Disruption*. 2014 [Online]. URL: https://wiki.fusion.ciemat.es/wiki/Disruption (hämtad 2024-05-01).

[6] C. D. Stephens, X. Garbet, Jonathan, C. Bourdelle, K. L. van de Plassche och F. Jenko, "Quasilinear gyrokinetic theory: A derivation of QuaLiKiz," *Journal of Plasma Physics*, årg. 87, nr 4, 2021. DOI: https://doi.org/10.1017/S0022377821000763.

[7] J. Brosander, O. Lindberg, W. Rieck och M. Åqvist, "Surrogatmodell av QuaLiKiz för turbulenta instabiliteter i en tokamak," Chalmers, Insitutionen för rymd-, geo och miljövetenskap, 2023, kandidatrapport.

[8] B. Martin och G. Shaw, "Fusion," i *Nuclear and Particle Physics: An introduction*, Hoboken, John Wiley & Sons, 2019, kap. 9, avs. 2, s. 357–371.

[9] S. Li, H. Jiang, Z. Ren och C. Xu, "Optimal Tracking for a Divergent-Type Parabolic PDE System in Current Profile," *Abstract and Applied Analysis*, årg. 2014, nr 4, 2014. DOI: https://doi.org/10.1155/2014/940965.

[10] E. Fransson, "Simulation and assessment of particle transport in fusion plasmas," Chalmers, Insitutionen för rymd-, geo och miljövetenskap, 2021, licentiatavhandling.

[11] F. Eriksson, "Fast ions and turbulent particle transport in tokamaks," Chalmers, Insitutionen för rymd-, geo och miljövetenskap, 2019, doktorsavhandling.

[12] C. Bourdelle, J. Citrin, B. Baiocchi m. fl., "Core turbulent transport in tokamak plasmas: bridging theory and experiment with QuaLiKiz," *Plasma Physics and Controlled Fusion*, årg. 58, nr 1, 2015. DOI: https://doi.org/10.1088/0741-3335/58/1/014036.

[13] C. Forssén, *TIF385 Bayesian inference and machine learning*. 2022 [Online]. URL: https://cforssen.gitlab.io/tif385-book/content/Intro/preface.html (hämtad 2024-02-17).

[14] C. M. Bisop, "Introduction," i *Pattern Recognition and Machine Learning*, Cambridge, Springer, 2006 [Online], kap. 1, s. 1–5. URL: https://www.math.chalmers.se/Stat/Grundutb/CTH/mve187/1819/Bishop.pdf (hämtad 2024-02-19).

[15] C. M. Bisop, "Feed-forward Network Functions," i *Pattern Recognition and Machine Learning*, Cambridge, Springer, 2006 [Online], kap. 5, avs. 1, s. 227–232. URL: https://www.math.chalmers.se/Stat/Grundutb/CTH/mve187/1819/Bishop.pdf (hämtad 2024-02-19).

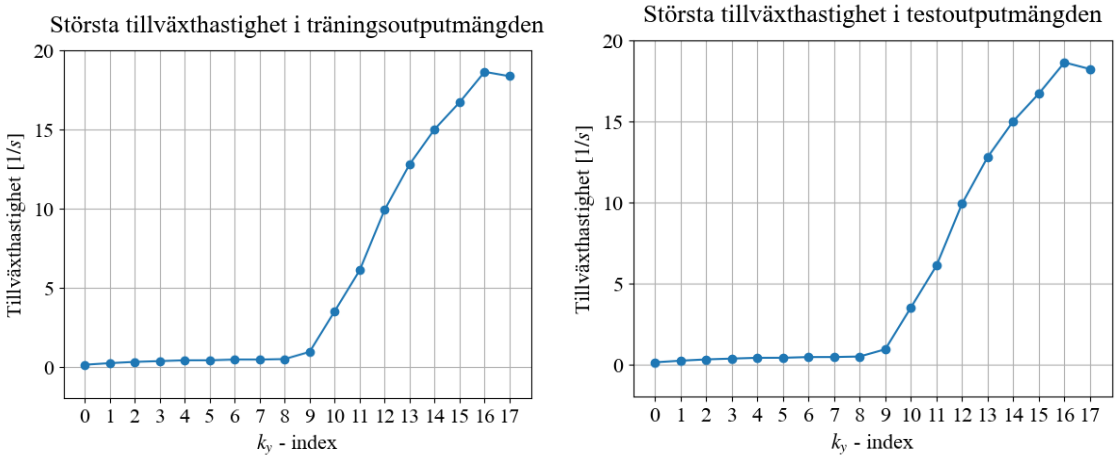
[16] I. Goodfellow, Y. Bengio och A. Courville, *Deep Learning*. MIT Press, 2016.

A

Datamängdanalys

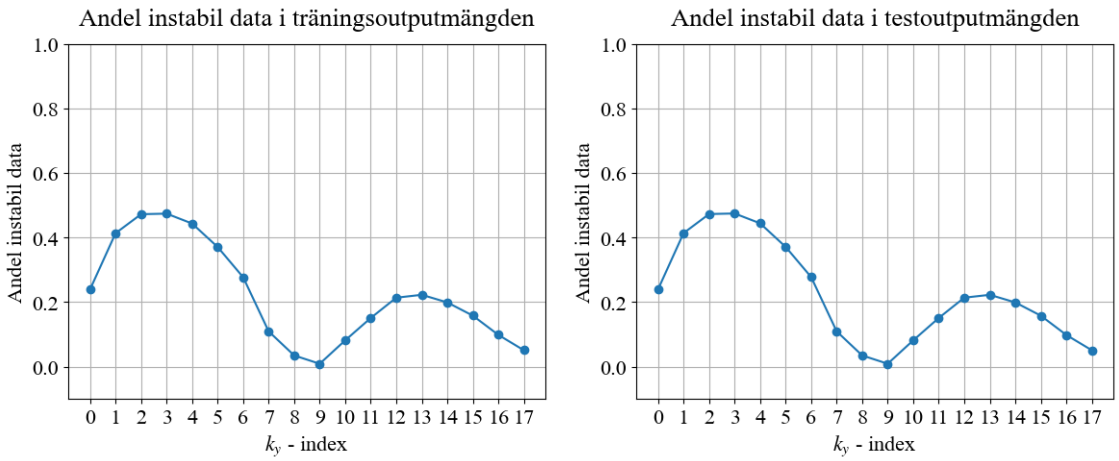
A.1 Datamängdanalys growth rate spectrum

I figur A.1 och A.2 visas delresultat från avsnitt 3.3. Trots att A.1a och A.1b ser nästan identiska ut är det marginell skillnad på grafernas värden. Detta på grund av att tränings- och testdatan är två uppdelningar av samma datamängd. Detsamma gäller figur A.2a och A.2b.



(a) **(b)**

Figur A.1: Figur A.1a visar de största tillväxthastigheterna för varje vågnummer (k_y - index) i träningsdatan. Figur A.1b visar motsvarande för testdatan.

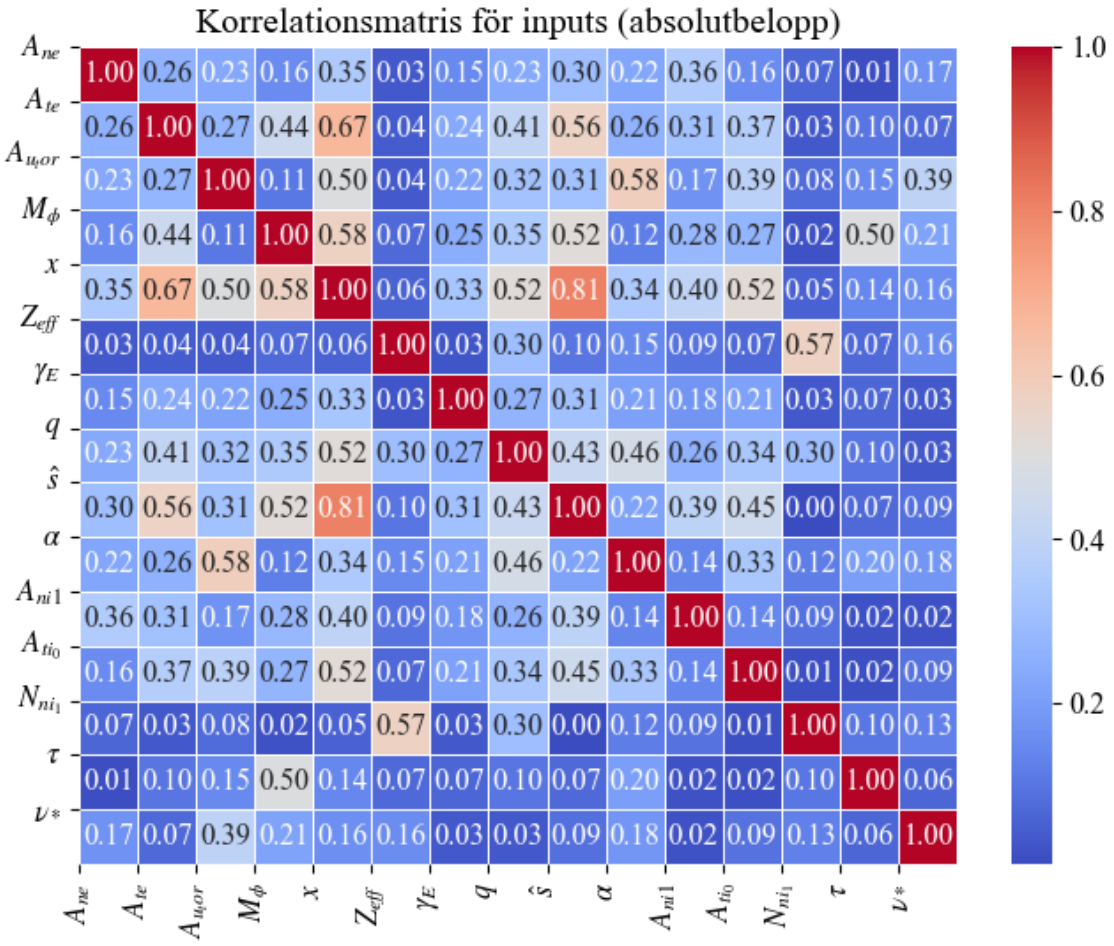


(a) **(b)**

Figur A.2: På y-axlarna visas andel instabil data i outputdatamängden, med instabil data menas tillväxthastigheter skilda från noll. Figur A.2a visar andelen instabil data för varje vågnummer (k_y - index) i träningsdatan. Figur A.2b visar motsvarande för testdatan.

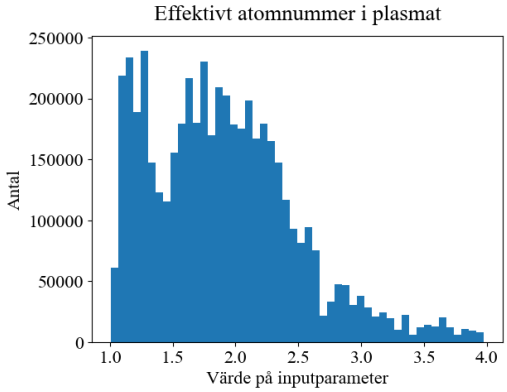
A.2 Datamängdanalys inputparametrar

A.2.1 Korrelationsmatris för inputparametrar

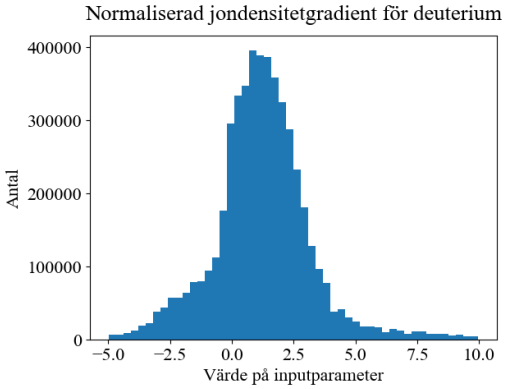


Figur A.3: Linjär korrelationsgrad mellan de olika parametrarna.

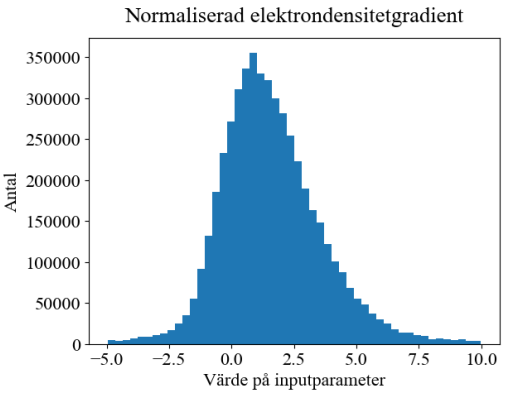
A.2.2 Histogram för inputparametrar i QuaLiKiz



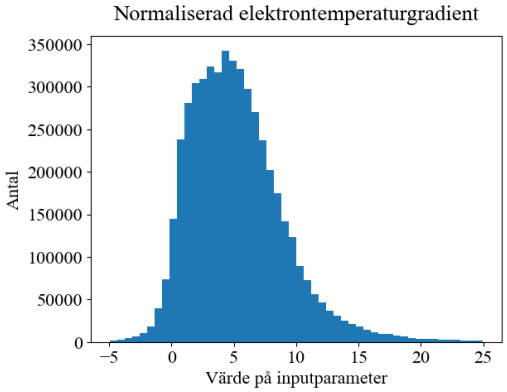
Figur A.4: Histogram över inputparametern: effektivt atomnummer i plasmat.



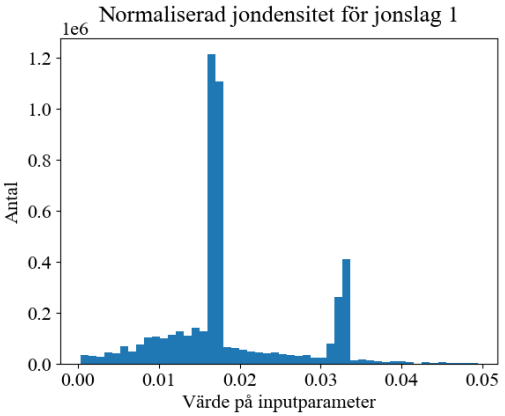
Figur A.5: Histogram över inputparametern: normaliserad jondensitetgradient för deuterium.



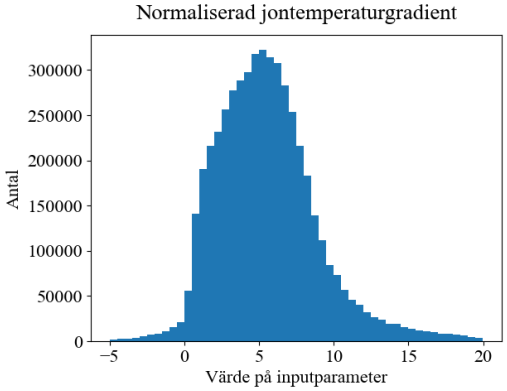
Figur A.6: Histogram över inputparametern: normaliserad elektrondensitetgradient.



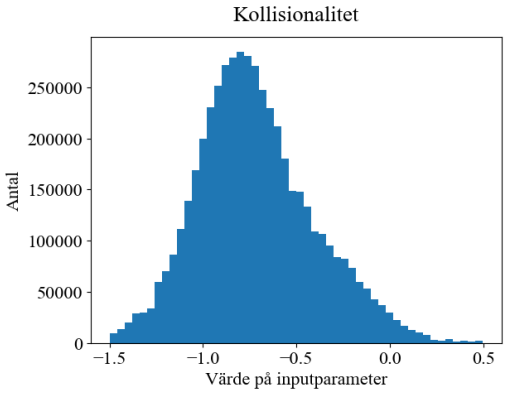
Figur A.7: Histogram över inputparametern: normaliserad elektrontemperaturgradient.



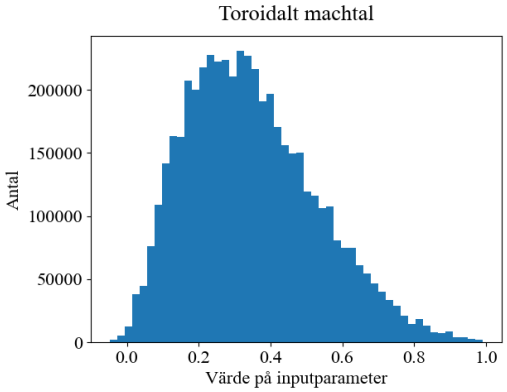
Figur A.8: Histogram över inputparametern: normaliserad jondensitet för jonslag 1.



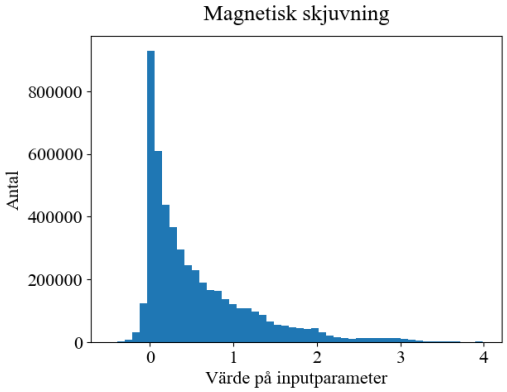
Figur A.9: Histogram över inputparametern: normaliserad jontemperaturgradient.



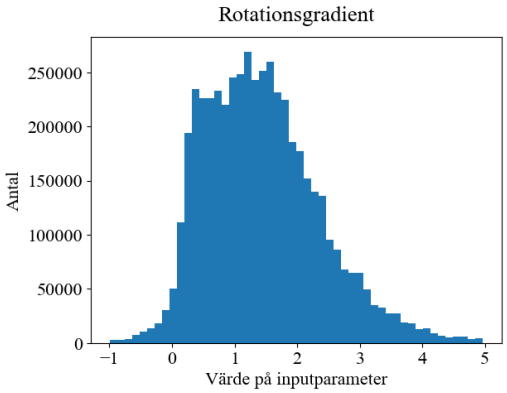
Figur A.10: Histogram över inputparametern: kollisionalitet.



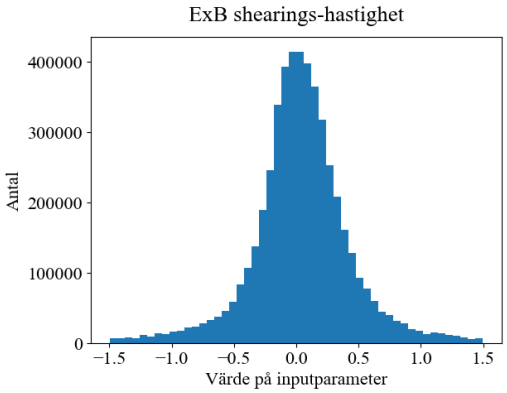
Figur A.11: Histogram över inputparametern: toroidalt machtal.



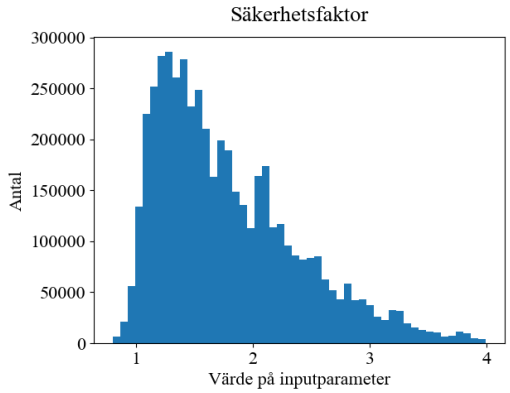
Figur A.12: Histogram över inputparametern: magnetisk skjuvning.



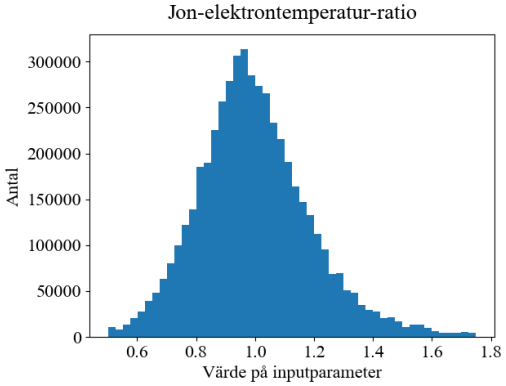
Figur A.13: Histogram över inputparametern: rotationsgradient.



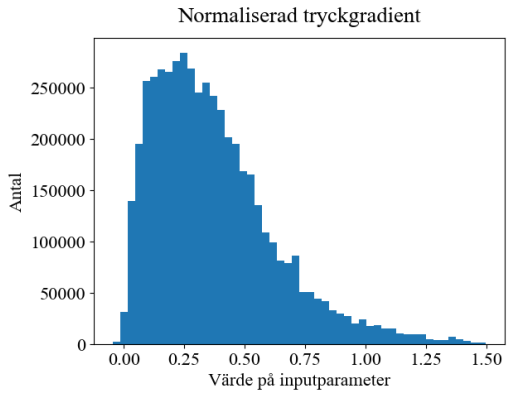
Figur A.14: Histogram över inputparametern: ExB shearings-hastighet.



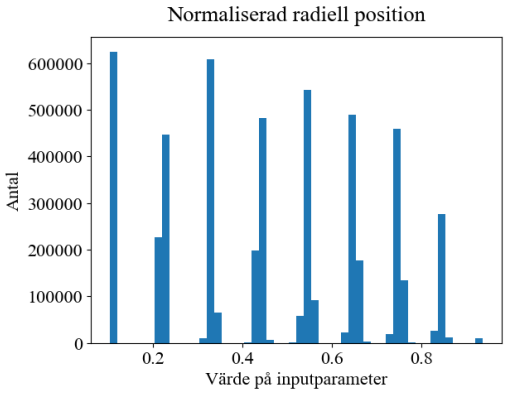
Figur A.15: Histogram över inputparametern: säkerhetsfaktor.



Figur A.16: Histogram över inputparametern: jon-elektrontemperaturratio.



Figur A.17: Histogram över inputparametern: normaliserad tryckgradient.

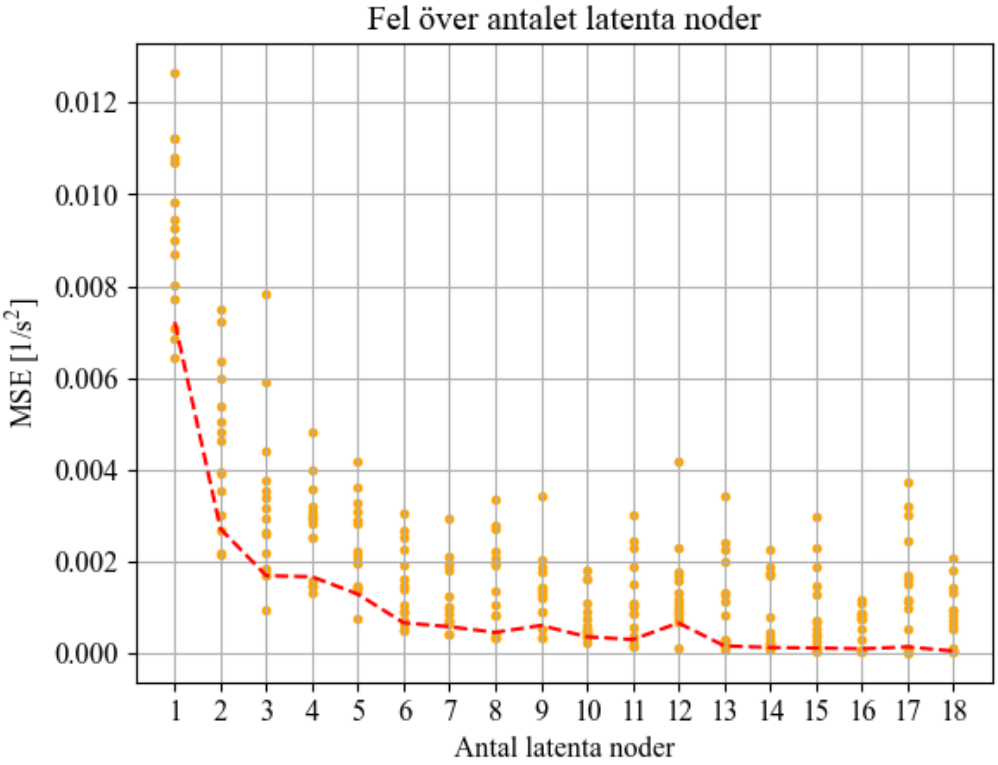


Figur A.18: Histogram över inputparametern: normaliserad radiell position.

B

Autoencoder för samtliga k_y

I figur B.1 har valideringsfelet för olika antalet noder i det latenta lagret beräknats och plottats. Observera att då antalet noder är fler än 2 börjar grafen stabilisera sig. Även om resultatet fortsätter förbättras anses förbättringen vara för liten för att rättfärdiga att öka antalet noder.

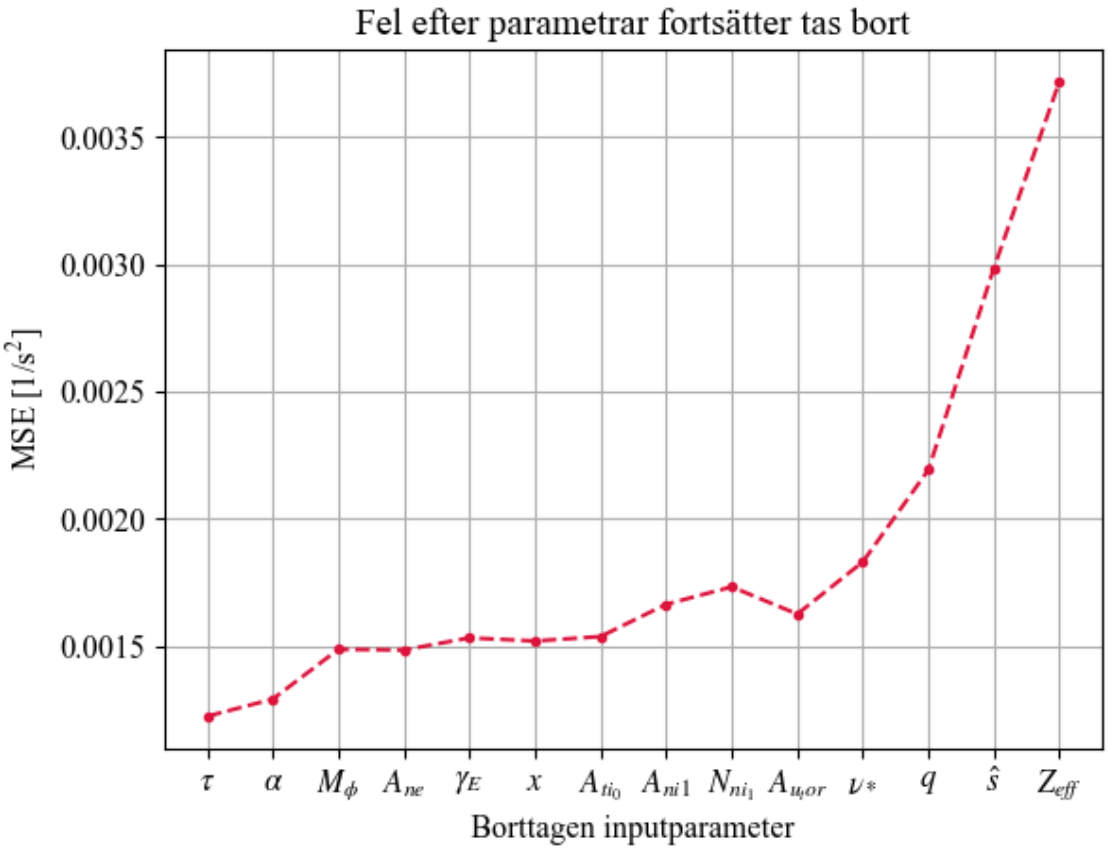


Figur B.1: Valideringsfelet efter träning för olika antal noder i det latenta lagret där varje mätning har genomförts 15 gånger. Linjen representerar medelvärdet av de 5 bästa mätningarna för varje nodantal.

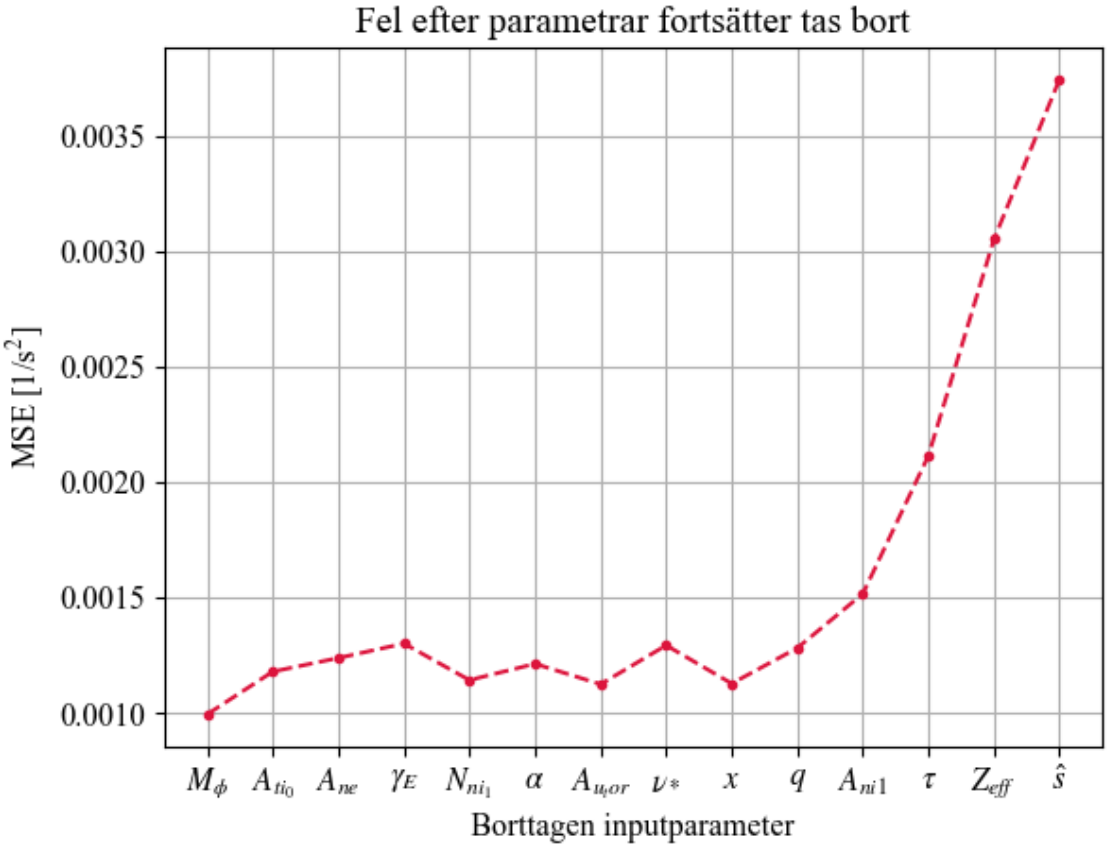
C

Reduceringsdata

C.1 Data från mätning av stor reducering



Figur C.1: Medelfelet av de 5 bästa mätningarna av totalt 15 där den utskrivna paramteren har tagits bort.

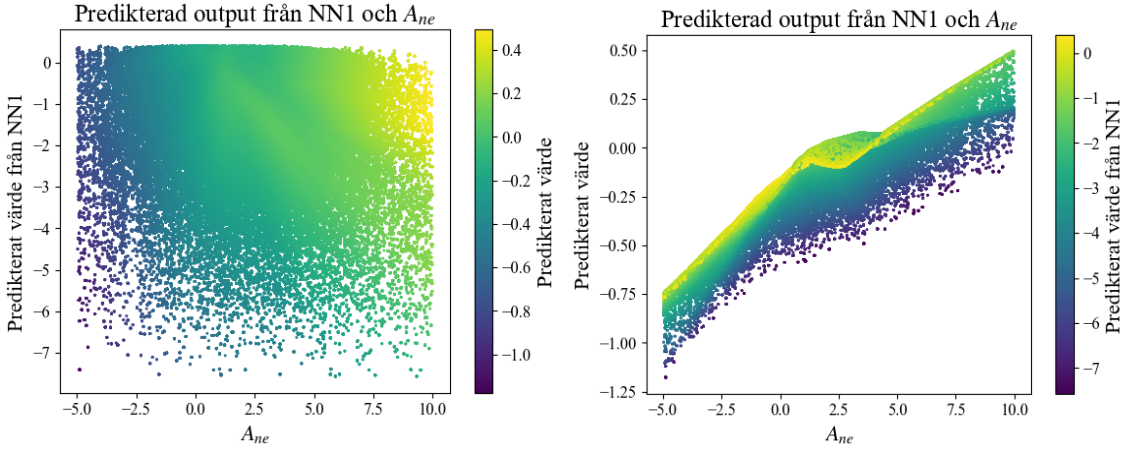


Figur C.2: Medelfelet av de 5 bästa mätningarna av totalt 15 där den utskrivna paramteren har tagits bort.

D

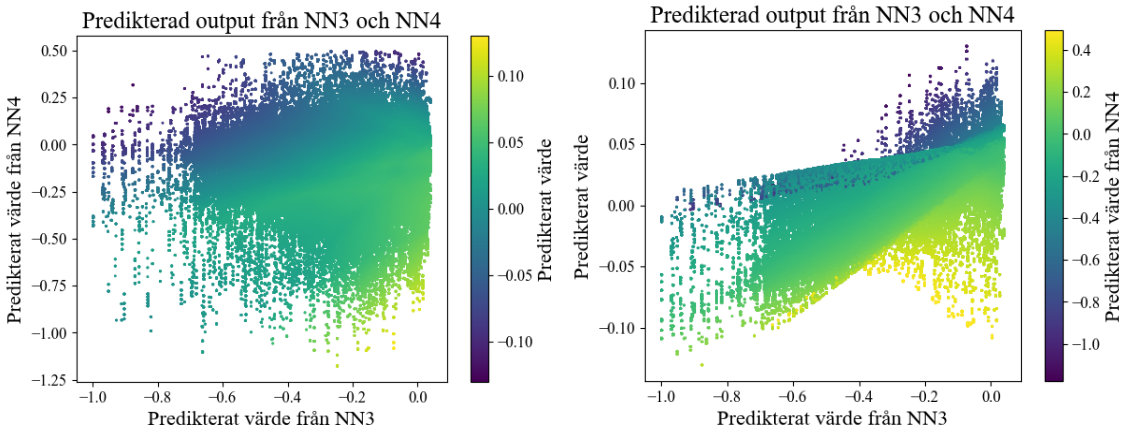
Korrelationer inom de neurala nätverken

D.1 Plots för samtliga neurala nätverk i framåtkonstruktionen



(a) **(b)**

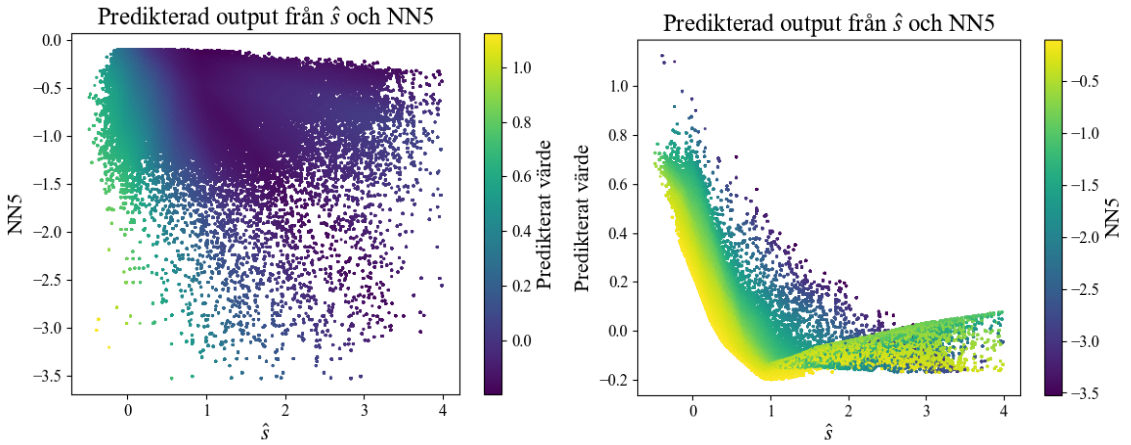
Figur D.1: Relationen mellan inputparametrarna och det predikterade värdet från NN4 i grenmodellen i figur 4.2. Figur (a) visar A_{ne} och output från NN1 plottade mot varandra. Figur (b) visar A_{ne} plottad mot det predikterade värdet.



(a) **(b)**

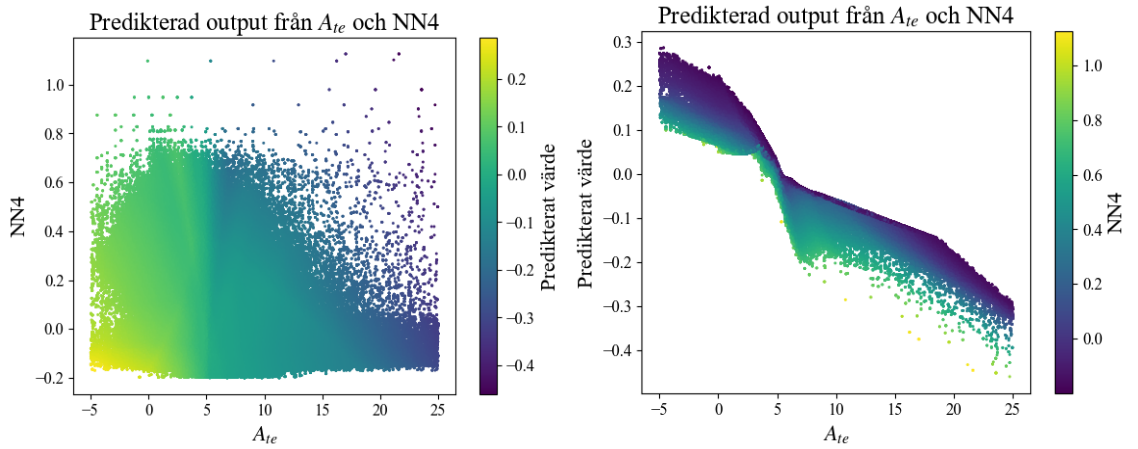
Figur D.2: Relationen mellan inputparametrarna och det predikterade värdet från NN5 i grenmodellen i figur 4.2. Figur (a) visar output från NN3 och output från NN4 plottade mot varandra. Figur (b) visar output från NN3 plottad mot det predikterade värdet.

D.2 Plots för samtliga neurala nätverk i bakåtkonstruktionen



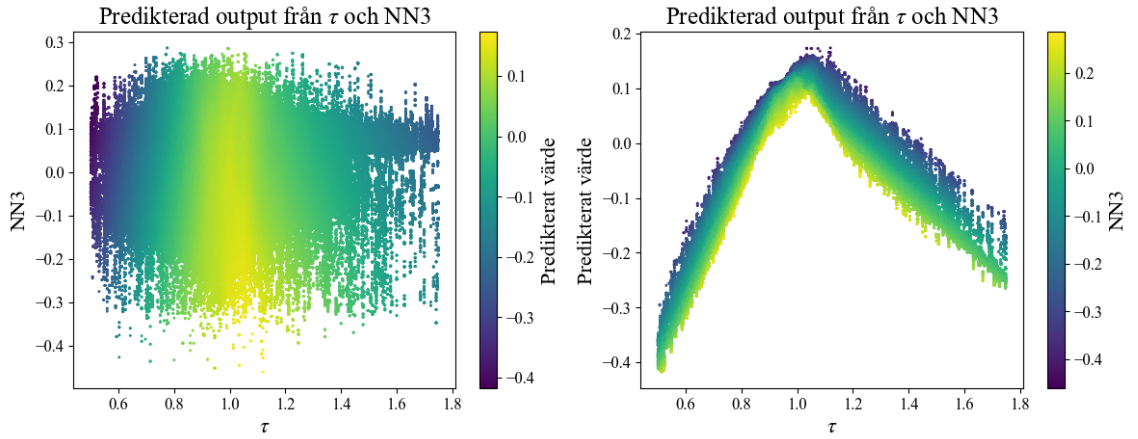
(a) **(b)**

Figur D.3: Relationen mellan inputparametrarna och det predikterade värdet från NN4 i grenmodellen i figur 4.3. Figur (a) visar $\hat{s}$ och output från NN5 plottade mot varandra. Figur (b) visar $\hat{s}$ plottad mot det predikterade värdet.



(a) **(b)**

Figur D.4: Relationen mellan inputparametrarna och det predikterade värdet från NN3 i grenmodellen i figur 4.3. Figur (a) visar A_{te} och output från NN4 plottade mot varandra. Figur (b) visar A_{te} plottad mot det predikterade värdet.



(a) **(b)**

Figur D.5: Relationen mellan inputparametrarna och det predikterade värdet från NN2 i grenmodellen i figur 4.3. Figur (a) visar τ och output från NN3 plottade mot varandra. Figur (b) visar τ plottad mot det predikterade värdet.

Rymd-, geo och miljövetenskap
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige
www.chalmers.se

CHALMERS