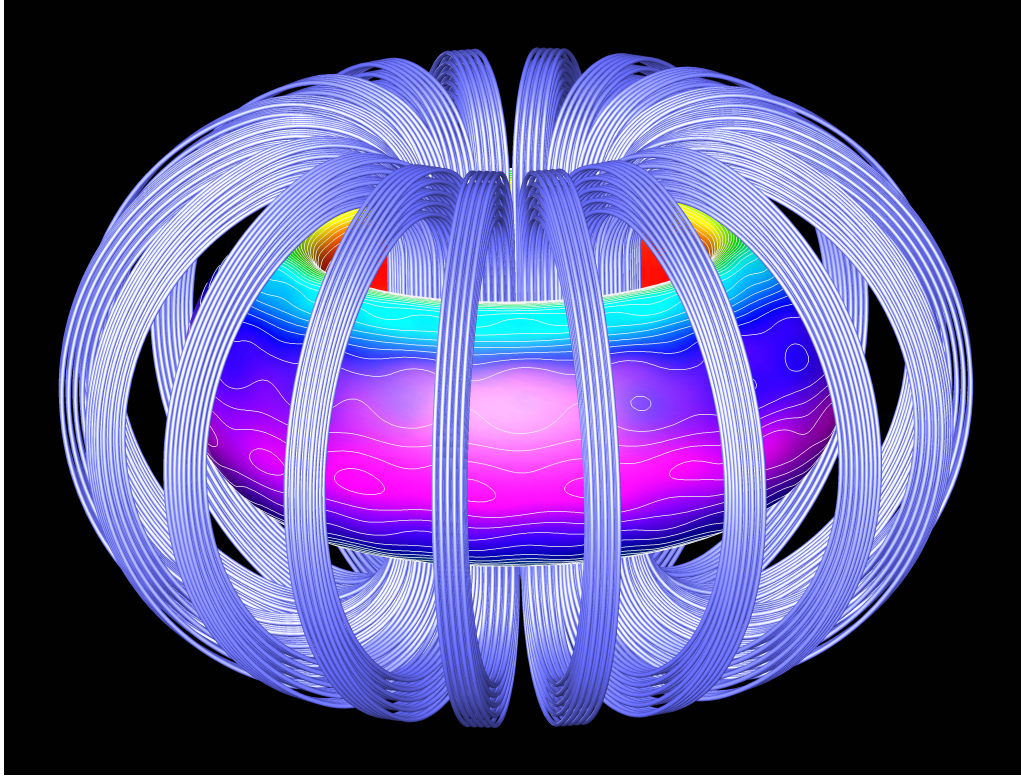




CHALMERS



Samband mellan input- och outputparametrar i QuaLiKiz-modellen

Undersökning av en numerisk modell som beräknar turbulenta instabiliteter i magnetiskt innesluten fusion

Kandidatarbete inom programmen Teknisk fysik och Globala system

Laura Flyckt, Max Green, Erik Olsson och Adam Orthog

Institutionen för rymd-, geo- och miljövetenskap

CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2024

www.chalmers.se

KANDIDATARBETE 2024

Samband mellan input- och outputparametrar i QuaLiKiz-modellen

Undersökning av en numerisk modell som beräknar instabiliteter i
magnetiskt innesluten fusion

Laura Flyckt, Max Green, Erik Olsson och Adam Orthag



CHALMERS

Institutionen för Rymd-, geo- och miljövetenskap
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2024

Samband mellan input- och outputparametrar i QuaLiKiz-modellen
Undersökning av numerisk modell som beräknar instabiliteter i magnetiskt innesluten fusion
Laura Flyckt, Max Green, Erik Olsson och Adam Orthag

© Laura Flyckt, Max Green, Erik Olsson och Adam Orthag, 2024.

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

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

Omslagsbild: 3D visualisering på plasma i jämvikt i ITER utifrån experimentell data, av Oakridgelabnews, 2009[1], CC BY 2.0

Skriven i L^AT_EX
Göteborg, Sverige 2024

Sammandrag

I ljuset av ett ökat globalt energibehov och nödvändigheten att övergå från fossila bränslen är användning av fusion som energikälla av stort intresse för mänskligheten. Framgångsrik användning av fusion skulle innebära att mänskligheten får tillgång till en ren och i princip ousinlig energikälla. För att fusion ska inträffa på jorden krävs det att fusionsbränslet befinner sig i plasmatillstånd med extremt hög temperatur, och således är fusion en energikrävande process. I en tokamak, en typ av fusionsreaktor, innesluts fusionsplasma magnetiskt. Den termiska och magnetiska energin i tokamaken är hög, vilket orsakar instabiliteter som ger upphov till turbulens och därmed energiförluster från plasmat. Genom att öka förståelsen för vad som orsakar instabiliteterna kan energiförlusterna minskas, vilket är eftersträvnadsvärt för att energiproduktionen från fusion i tokamaker ska bli nettopositiv.

Att simulera de instabiliteter som ger upphov till turbulens i ett fusionsplasma är beräkningsmässigt kostsamt, och QuaLiKiz-modellen har utvecklats för att minska beräkningsbelastningen. Även Qualikiz är dock relativt beräkningstung, och därför har surrogatmodeller baserade på neurala nätverk som tränas på data från QuaLiKiz utvecklats. Dessa är dock i allmänhet svårtolkade, eftersom det ofta inte går att förstå hur neurala nätverk gör prediktioner. I detta arbete skapas en ny surrogatmodell av QuaLiKiz, som kommer att kallas för en grenmodell. Grenmodellen består av ett antal sammanhängande neurala nätverk som tar emot två inputparametrar och producerar en output. Syftet med grenmodellen är att prediktera hur snabbt instabiliteter för jonskalor växer i ett fusionsplasma givet vissa inputvärden, samtidigt som grenmodellens struktur gör det möjligt att visualisera hur en specifik inputparameter påverkar tillväxthastigheterna. För att förenkla arbetet med grenmodeller skapas en autokodare, som visar att det går att komprimera de niodimensionella tillväxthastigheterna till två dimensioner utan signifikant informationsförlust. Dessutom utförs en reduktion av inputparameterrymden, från 15 till sex inputparametrar.

Abstract

In light of an increased global energy demand and the necessity to transition away from fossil fuels, the use of fusion as an energy source is of great interest to humanity. Successful utilisation of fusion would provide mankind with a clean and virtually inexhaustible energy source. For fusion to occur on Earth, the fusion fuel must be in a plasma state at extremely high temperatures, making fusion an energy-intensive process. In a tokamak, a kind of fusion reactor, fusion plasma is magnetically confined. The thermal and magnetic energy in the tokamak is high, which causes instabilities that in turn lead to turbulence and thereby energy losses from the plasma. By increasing the understanding of what causes these instabilities, energy losses can be reduced, thus contributing to the ultimate end of making energy production from fusion in tokamaks net positive.

Simulating the complex behaviors exhibited by a fusion plasma is computationally expensive, and the QuaLiKiz model has been developed to reduce the computational load. However, even QuaLiKiz is relatively computationally heavy, and therefore surrogate models based on neural networks trained on data from QuaLiKiz have been developed. These are generally difficult to interpret, as it is often not possible to understand how neural networks make predictions. In this thesis, a new surrogate model of QuaLiKiz, to be called a branch model, is created. The branch model consists of a number of interconnected neural networks that receive two input parameters and produce one output. The purpose of the branch model is to predict how quickly instabilities for ion scales grow in a fusion plasma given certain input values, while the structure of the branch model allows for visualisation of how a specific input parameter affects growth rates. To simplify the work with branch models, an autoencoder is created, which shows that it is possible to compress the nine-dimensional growth rates to two dimensions without significant information loss. Additionally, a reduction of the input parameter space is performed, from 15 to six input parameters.

Nyckelord: QuaLiKiz, autoencoder, maskinlärning, grenmodell, plasma, fusion, tokamak, turbulent transport, fusionsplasma.

Nomenklatur

Nedan följer en redovisning av den notation och de förkortningar som används i arbetet, samt korta beskrivningar av QuaLiKiz-modellens inputparametrar.

Förkortningar och variabler

\mathbb{R}	Reella endimensionella domänen
\mathbb{R}_0^+	Reella endimensionella domänen, tal ≥ 0
\mathbb{R}^2	Reella tvådimensionella domänen
ReLU	Rectified Linear Unit
Adam	Adaptive Moment Estimation
β	Inlärningshastighet
MSE	Medelkvadratfel
λ	Förlustfunktion

Inputparametrar

α	Normaliserad tryckgradient
A_{ne}	Normaliserad elektrondensitetgradient
A_{ni_1}	Normaliserad jondensitetsgradient för deuterium
A_{te}	Normaliserad elektrontemperaturgradient
A_{ti_0}	Normaliserad jontemperaturgradient
A_{utor}	Rotationsgradient
γ_E	ExB skjuvningshastighet
ν_*	Kollisionalitet med logaritmisk skala
M_ϕ	Toroidalt machtal
N_{ni_1}	Normaliserad jondensitetsart, jonslag 1
q	Säkerhetsfaktor

\hat{s}	Magnetisk skjuvning
τ	Jon-elektrontemperatur-relation
x	Normaliserad radiell position
Z_{eff}	Effektivt atomnummer i plasma

Innehåll

Nomenklatur	v
Figurer	ix
Tabeller	xi
1 Introduktion	1
1.1 Bakgrund	1
1.2 Syfte och frågeställning	2
2 Teori	4
2.1 Fusion	4
2.1.1 Magnetiskt innesluten funktion	5
2.1.1.1 Tokamaker	5
2.1.1.2 Gyrokinetisk rörelse	6
2.1.2 Drifter av partiklar i plasma	7
2.1.3 Turbulent transport	8
2.1.4 Instabiliteter	8
2.2 QuaLiKiz-modellen	9
2.2.1 Inputparametrar för Qualikiz	9
2.2.2 Outputparametrar för Qualikiz	9
2.3 Neurala nätverk	10
2.3.1 Uppbyggnad	10
2.3.2 Träning	12
2.3.3 Optimering och lokala minimum	12
2.3.4 Autokodare	13
2.3.5 Svarta lådor	14
2.3.6 Grenmodeller	15
3 Metoder och delresultat	17
3.1 Mätningsteknik	17
3.2 Förbehandling av data	17
3.2.1 Outputparametrar	17
3.2.2 Inputparametrar	18
3.3 Autokodare	19
3.4 Reduktion av inputparameterrymd	21
3.5 Grenmodeller	24

Innehåll

3.5.1	Generella metoder för grenmodellerna	24
3.5.2	Metod för att hitta en optimal grenmodell	26
3.5.2.1	Första reduktionen	26
3.5.2.2	Andra reduktionen	27
3.5.2.3	Sista reduktionen	28
3.5.2.4	Förgrening av de grovt indelade grenarna	29
4	Resultat och diskussion	32
4.1	Sammanfattning av delresultat	32
4.2	Diskussion av delresultat	32
4.2.1	Autokodare	32
4.2.2	Reduktion av parameterrymd	32
4.3	Resultat och diskussion av grenmodellen	33
4.4	Begränsningar i arbetet	41
4.5	Framtida arbete	41
5	Slutsatser	43
6	Sammanfattning	44
	Referenser	45
A	Grafer	I
B	Tabeller	VI

Figurer

2.1	Figur som visar vilka massnummer för atomer som fusion respektive fission kan producera energi [16]. Återgiven med tillstånd.	5
2.2	Bild över hur en tokamak är konstruerad. I figuren visas även de strömmar och magnetfält som verkar i tokamaken. Bild av S. Li, H. Jiang, Z. Ren och C. Xu, 2014 [21], CC BY 4.0.	6
2.3	Figuren visar tillväxthastigheter för alla k_y -index för en slumpmässigt utvald nollskild instans i träningsdatasetet. Den vertikala linjen separerar jonskalorna (till vänster) från elektronskalorna (till höger).	10
2.4	En möjlig struktur på ett neuralt nätverk. I denna representation finns n inputparametrar och två gömda lager med m noder i varje, som sedan predikterar en output o	11
2.5	En nod i ett gömt lager på ett neuralt nätverk som tar emot insignaler x_i , vikter w samt en bias b och predikterar outputparametrar.	12
2.6	Ett exempel på hur en autokodare kan se ut. Kodaren består av inputlagret och ett gömt lager, och den latent representationen är tvådimensionell. Avkodaren består sedan av ett gömt lager, och de återskapade inputvärdena.	14
2.7	Figur som visar en gren i en grenmodell. Ett neuralt nätverk har använts för att förutse o_1 genom x_1 och x_2	15
2.8	En möjlig struktur på en grenmodell, här med fem unika inputs x_1, \dots, x_5 varav o_1, \dots, o_4 är förgreningar av inputsen där den sista grenen ger outputen y	15
2.9	Med hjälp av grafer som dessa går det att få en uppfattning om hur inputparametrar relaterar till varandra och till outputprediktioner.	16
3.1	Figuren visar andelen nollskilda tillväxthastigheter för varje k_y -index. Till vänster om den vertikala linjen syns de k_y -index som relaterar till jonskalor, och till höger syns elektronskalorna.	18
3.2	Figuren illustrerar hur träningsdata- och testdatasetet har liknande uppdelning men olika antal datapunkter. Observera att träningsdatan i figuren även innefattar de värden som täcks av testdatan. Se figur A.2 för resterande inputparametrar.	19
3.3	Figur över förluster för testdata vid olika antal latent noder då 9 lösningsparametrar skulle predikteras.	20

3.4	Medelvärde av valideringsfel(MSE) jämfört med om någon av de olika parametrarna tas bort. Den parameter som ger lägst valideringsfel är rödmarkerad och togs bort för nästa körning. Se A.1 för figurer från alla körningar.	22
3.5	Figuren visar i vilken ordning de olika parametrarna har tagits bort där y-axeln är valideringsfel efter borttagen parameter och x-axeln är den borttagna parametern. Vid den första punkten har ingen parameter blivit borttagen.	22
3.6	Jämförelse med genomsnitt av valideringsfel(MSE) mellan den linjära modellen och det neurala nätverket	24
3.7	Strukturen för det neurala nätverket som predikterade tillväxthastigheterna. Nätverket hade tre gömda lager med 30 latent dimensioner. Pilarna representerar vikter.	25
3.8	Strukturen för det neurala nätverket som predikterade mellanliggande outputs. Nätverket hade två gömda lager med 20 latent dimensioner, och en outputvektor. Pilarna representerar vikter.	26
3.9	I figurerna visualiseras grenstrukturerna för de två första förgreningarna.	28
3.10	Den slutgiltiga grenmodellens struktur. NN_1 , NN_2 , NN_3 och NN_4 representerar G_1 , medan NN_4 och NN_5 representerar G_2	31
4.1	Figurerna visar samband mellan inputparametrarna γ_E , \hat{s} och output från NN_1 i figur 3.10.	34
4.2	Figurerna visar samband mellan x , output från NN_1 och output från NN_2	35
4.3	Figurerna visar samband mellan A_{ne} , output från NN_2 och output från NN_3	36
4.4	Figurerna visar samband mellan A_{ti_o} , output från NN_3 och output från NN_4	37
4.5	Figurerna visar samband mellan A_{ti_o} , τ och output från NN_5	38
4.6	Figurerna visar samband mellan A_{ti_o} , output från NN_3 och output från NN_4	39
4.7	4.5 och 4.6 visar plottar för sista övre grenen och sista nedre grenen i figur 3.10, medan 4.7 visar den slutliga plotten där tillväxthastigheter predikteras från de sista grenarna. I bilden är k_y -index = 0. För resterande k_y -index se figur A.3.	40
A.1	Grafer som visar minskande antal parametrar använda för att åter skapa outputs.	I
A.1	Grafer som visar minskande antal parametrar använda för att åter skapa outputs.	II
A.2	Histogram över inputparametrarnas värden i både träningsdata- och testdatasetet	III
A.2	Histogram över inputparametrarnas värden i både träningsdata- och testdatasetet.	IV
A.3	Predikterade tillväxthastigheter för k_y -index 1...8.	V

Tabeller

3.1	Medelvärde för MSE för modeller där sex respektive fem inputparametrar använts för att prediktera tillväxthastigheterna.	25
3.2	Grenstrukturer och medelkvadratfel MSE_i för de fyra bästa samt den sämsta strukturen med fem inputparametrar i båda grenarna, samt standardavvikelse σ_i för dessa. Kombinationerna är sorterade efter MSE_i , och den bästa kombinationen är markerad med fetstil. Resultat för resterande konfigurationer visas i tabell B.1.	27
3.3	Medelkvadratfel MSE_i för den bästa och sämsta grenstrukturen med fem inputparametrar i en gren och fyra i den andra samt standardavvikelse σ_i för dessa. Kombinationen med lägst MSE är markerad med fetstil. Resultat för resterande konfigurationer visas i tabell B.2.	27
3.4	Grenstrukturer och medelkvadratfel MSE_i för strukturerna med fyra inputparametrar i båda grenarna, samt standardavvikelse σ_i för dessa. Kombinationerna är sorterade efter MSE_i , och den bästa kombinationen är markerad med fetstil.	28
3.5	Grenstruktur och medelkvadratfel MSE_i för strukturerna med fem inputparametrar i en gren och tre i den andra, samt standardavvikelse σ_i för dessa. Kombinationerna är sorterade efter MSE_i , och den bästa kombinationen är markerad med fetstil.	29
3.6	Medelkvadratfel MSE_i för den bästa och den sämsta strukturen där en parameter brutits loss från grenen med fem inputparametrar, samt σ_i för dessa. Den bästa kombinationen är markerad med fetstil. Resultat för resterande konfigurationer visas i tabell B.5.	29
3.7	Medelkvadratfel MSE_i för strukturerna där två parametrar brutits loss från grenen med fem inputparametrar, samt standardavvikelse σ_i för dessa.	30
3.8	Medelkvadratfel MSE_i för strukturerna där tre parametrar brutits loss från grenen med fem inputparametrar, samt standardavvikelse σ_i för dessa. Resultat för resterande konfigurationer redovisas i tabell B.6.	30
3.9	Medelkvadratfel MSE_i för den bästa och sämsta modellen där en parameter brutits loss från grenen med tre inputparametrar, samt standardavvikelse σ_i för dessa. Resultat för resterande konfigurationer visas i tabell B.7.	30

B.1	Medelkvadratfel MSE_i för de tre bästa modellerna med fem inputparametrar i båda grenarna samt standardavvikelser σ_i för dessa. Tabellen är sorterad efter MSE_i , där index 1 har lägst MSE_i	VI
B.2	Medelkvadratfel MSE_i för de tre bästa modellerna med fem inputparametrar i en gren och fyra i den andra samt standardavvikelser σ_i för dessa. Tabellen är sorterad efter MSE_i , där index 1 har lägst MSE_i . VII	VII
B.3	Medelkvadratfel MSE_i för de tre bästa modellerna med fyra inputparametrar i samma gren samt standardavvikelser σ_i för dessa. Tabellen är sorterad efter MSE_i , där index 1 har lägst MSE_i	VII
B.4	Grenstruktur och medelkvadratfel MSE_i för strukturerna med fem inputparametrar i en gren och tre i den andra, samt standardavvikelser σ_i för dessa. Kombinationerna är sorterade efter MSE_i , och den bästa kombinationen är markerad med fetstil.	VII
B.5	Medelkvadratfel MSE_i för de kombinationer där en parameter brutits loss från grenen med fem inputparametrar, samt σ_i för dessa. Tabellen är sorterad efter MSE_i , där index 1 har lägst MSE_i	VIII
B.6	Medelkvadratfel MSE_i för strukturerna där tre parametrar brutits loss från grenen med fem inputparametrar, samt standardavvikelser σ_i för dessa. Tabellen är sorterad efter MSE_i , där index 1 har lägst MSE_i	VIII
B.7	Medelkvadratfel MSE_i för strukturerna där en parameter brutits loss från grenen med tre inputparametrar, samt standardavvikelser σ_i för dessa. Tabellen är sorterad efter MSE_i , där index 1 har lägst MSE_i . .	VIII

1

Introduktion

Energiproduktionen från stjärnor, inklusive solen, är en förutsättning för liv på jorden. Denna energiproduktion grundas i en process inom kärnfysik benämnd fusion, vilket är den process där två lättare atomkärnor slås ihop och bildar en tyngre kärna, en reaktion som frigör stora mängder energi [2]. Fusionsprocessen utgör en potentiellt stor källa till ren energi, och under årtionden har forskare strävat efter att återskapa fusion i kontrollerade former på jorden[3]. En framgång med att göra detta skulle möjliggöra etableringen av större fusionsenergianläggningar, som i sin tur skulle kunna använda sig av en ny energikälla som är miljövänlig, säker och långsiktigt hållbar [4].

Det finns flera metoder för att återskapa fusionsreaktioner på jorden, varav tokamakerna är den mest välutvecklade reaktordesignen [5]. En tokamak är en konstruktion med torusform, där kraftfulla magnetfält innesluter ett fusionsplasma. För att fusion ska inträffa på jorden behöver fusionsbränslets temperatur höjas till över 100 miljoner grader Celsius [6]. I en tokamak åstadkoms detta genom att först värma plasmat ohmiskt genom ström och sedan använda tekniker som “neutral-beam injection” och/eller “ion/electron cyclon resonance heating” [5] [7]. Uppvärmningen leder till att fusionsbränslet hamnar i ett plasmatillstånd. Den extrema temperaturen och bränslets plasmatillstånd skapar förhållanden där atomkärnor kan kollidera och sammanfogas, en process som frigör en signifikant mängd energi. En förståelse för plasmats dynamik inom dessa konstruktioner är essentiellt för att framgångsrikt genomföra fusion i tokamaker [5].

1.1 Bakgrund

Experimentella fusionsreaktorer finns på många platser på jorden, och en vanlig konstruktion av dessa är tokamaker. [8]. I tokamaker används vanligtvis tunga isotoper av väte som bränsle. Tokamakens magnetfält och strömmar driver laddade partiklar i bränslet. På grund av att plasmat i sig är strömledande och dessutom påverkas av många faktorer i en tokmak så kommer det att uppvisa komplexa beteenden [5].

I vissa fall uppför sig plasmat, och därmed även dess värmefflöde, turbulent, vilket gör att deras beteenden blir oförutsägbara. [9]. Turbulensen kan leda till att både partiklar och värme lämnar plasmat; detta är energi som går förlorad i fusionsprocessen, vilket sänker verkningsgraden för fusionsreaktorn. Därför hänger möjligheten att använda fusion som energikälla till stor del på att kunna förutspå den turbu-

lenta transporten i plasmat. Det är relevant att kunna beräkna hur instabiliteter växer i ett fusionsplasma och leder till den ovan nämnda turbulensen, men ekvationerna som styr detta går hittills inte att lösa analytiskt [10]. Med anledning av detta används numeriska modeller, där QuaLiKiz-modellen är ett exempel. QuaLiKiz beräknar tillväxthastigheter av instabiliteter i plasmat, vilka ger en indikation på när turbulens uppstår [10]. Modellen har 15 inputparametrar som motsvarar de mest relevanta storheterna som påverkar tillväxthastigheter för instabiliteter i ett fusionsplasma. QuaLiKiz outputparametrar är själva tillväxthastigheterna. Om en tillväxthastighet är noll är plasmat stabilt, medan det är instabilt om tillväxthastigheten är över noll. Högre tillväxthastigheter korrelerar med en större instabiliteter.

QuaLiKiz är en relativt beräkningstung numerisk modell, och simuleringar tar därmed längre tid än vad som ofta är önskvärt. För att effektivisera användningen av QuaLiKiz och kunna tillämpa nya metoder för att undersöka hur inputparametrarna i QuaLiKiz påverkar modellens tillväxthastigheter, så kan en surrogatmodell användas. Syftet med surrogatmodellen är att emulera QuaLiKiz och generera tillräckligt precisa outputprediktioner på ett effektivt sätt. Med surrogatmodell menas här ett neuralt nätverk som har tränats på data från QuaLiKiz-modellen. Modeller bestående av neurala nätverk lider dock av att de är mycket svåra att tolka - det går i allmänhet inte att veta hur ett neuralt nätverk kommer fram till sina beslut [11]. Att veta varför saker sker är viktigt inom forskning, och därför finns ett behov av att öka tolkbarheten för surrogatmodellen. Ett sätt att göra detta är att bryta ner modellen i flera sammanhängande, mindre neurala nätverk som tillsammans skapar en större struktur. En sådan struktur av neurala nätverk kommer i detta arbete att kallas grenmodell. Alltså, en grenmodell består av flera neurala nätverk; varje neuralt nätverk tar emot två inputparametrar och ger ut en outputparameter. Genom att dela upp modellen på detta sätt blir det möjligt att visualisera processen från inputdata till outputdata.

1.2 Syfte och frågeställning

Syftet med denna rapport är att skapa en surrogatmodell, en så kallad grenmodell, av QuaLiKiz med ett reducerat antal inputparametrar, och sedan försöka finna samband mellan dessa inputparametrar och tillväxthastigheter för instabiliteter i fusionsplasma i denna modell. De frågor som kommer att besvaras är följande:

- QuaLiKiz har outputparametrar som motsvarar tillväxthastigheter i 18 dimensioner, varav nio representerar jonskalor och är de som är relevanta i detta arbete. Går det att skapa en lågdimensionell representation av jonskalorna? Om det är möjligt, hur få dimensioner kan jonskalorna komprimeras till utan att en signifikant mängd information förloras? Svaret på denna fråga kommer att ge en indikation på hur mycket inputparametrarna kan komprimeras i grenmodellen.
- Vilka inputparametrar har störst betydelse för att prediktera tillväxthastigheterna för jonskalorna? Svaret på denna fråga kommer att ge värdefull insikt i sig, samtidigt som det bestämmer vilka inputparametrar som ska ingå i gren-

modellen.

- Vad är den bästa strukturen på en grenmodell givet de villkor som bestäms genom svaren på föregående frågor? Vilka samband kan hittas mellan input- och outputparametrarna i grenmodellen?

2

Teori

I följande sektion förklaras den underliggande teorin bakom fysiken samt maskinlärning som använts i kandidatarbetet.

2.1 Fusion

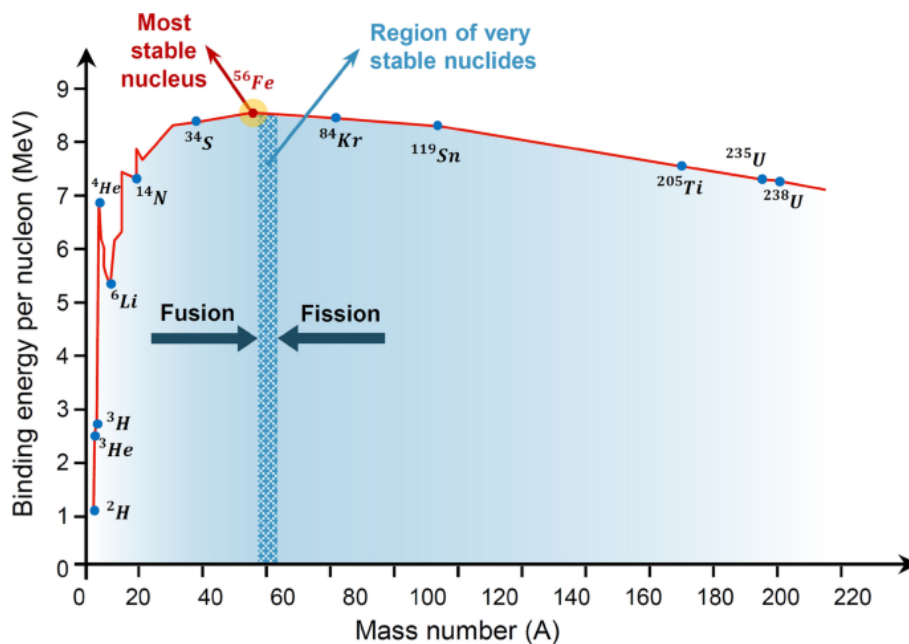
Fusion är en kärnfysikalisk process där två atomkärnor slås samman och bildar en tyngre kärna [2]. Ett exempel på en fusionsprocess är när deuterium och tritium reagerar med varandra och slås samman[12]:



Energi frigörs vid fusion när de två sammanslagna atomkärnorna uppvisar en lägre sammanlagd massa än de individuella kärnorna före reaktionen, vilket är fallet ovan [2]. Energifrigörningen förklaras på grund av massdefekten och genom Einsteins mass-energi-ekvivalensprincip [13]:

$$E = mc^2 \quad (2.2)$$

Förklaringen till att en massdefekt kan uppstå trots att antalet nuklider är oförändrat före och efter reaktionen är eftersom olika kärnkonfigurationer har olika bindningsenergi [14]. All sammanslagning av två atomkärnor som leder till en massdefekt innebär att fusion inträffar [5]. Detta gäller för samtliga nuklider lättare än järn. För nuklider tyngre än järn leder en sammanslagning av atomkärnor istället till en massökning, och energitillförsel krävs för att möjliggöra reaktionen. I motsats till detta innebär fission att en tyngre atom istället delar på sig till två eller flera lättare atomer med mindre total massa efteråt, vilket frigör energi. Fission kan ses som den omvända processen till fusion för tyngre element [15].



Figur 2.1: Figur som visar vilka massnummer för atomer som fusion respektive fission kan producera energi [16]. Återgiven med tillstånd.

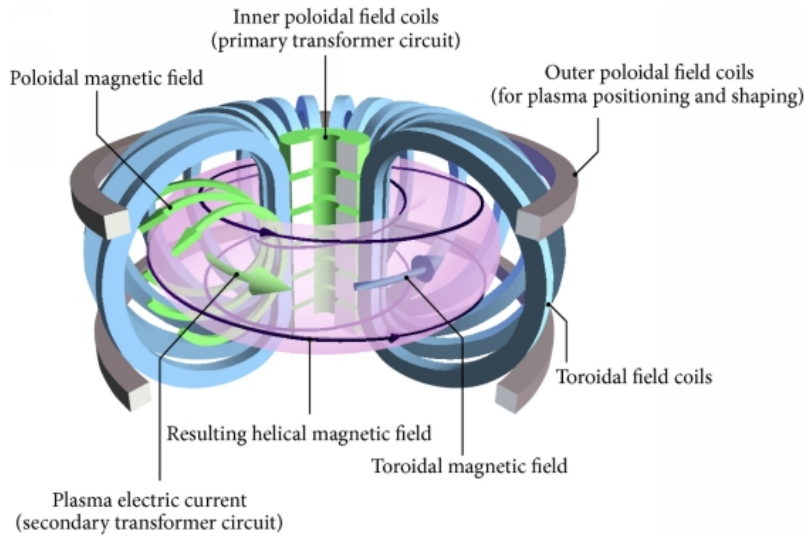
En förutsättning för att fusion ska inträffa är att atomerna är i plasma-tillstånd [2]. Detta tillstånd uppnås när en gas värms upp till höga temperaturer, vilket orsakar att atomernas elektroner frigörs och rör sig fritt från kärnan. När temperaturen är tillräckligt hög får atomkärnorna så mycket rörelseenergi att de kan kollidera trots den elektrostatiska repulsionen som ges av Coulombs lag [17]. Hög rörelseenergi på atomkärnorna är därför en förutsättning för att fusion ska äga rum. I solen uppnås rätt förutsättningar för fusion gå grund av den höga gravitationen vid solens kärna. På jorden är en metod att uppnå rätt förutsättningar för fusion genom att använda magnetisk inneslutning [18].

2.1.1 Magnetiskt innesluten funktion

Magnetiskt innesluten fusion har diskuterats sedan innan 1960-talet [19]. Följande delar ger teori om hur denna process görs möjlig och fungerar.

2.1.1.1 Tokamaker

Ordet tokamak kommer ursprungligen från det ryska uttrycket för 'torodial kam-mare med magnetiska spolar' [20]. En tokamak är en konstruktion formad som en torus där magneter används för att driva plasmat av väteisotoper vilket visas i figur 2.2. Det lila området i figuren är plasmat som genereras och styrs av magneterna i tokamaken.



Figur 2.2: Bild över hur en tokamak är konstruerad. I figuren visas även de strömmar och magnetfält som verkar i tokamaken. Bild av S. Li, H. Jiang, Z. Ren och C. Xu, 2014 [21], CC BY 4.0.

plasmata som används i tokamaker skapas av väteisotoper som värms upp till extrema temperaturer med hjälp av strömmar, magnetfält och tillsatsvärme i form av exempelvis högenergipartiklar eller radiovågor[22]. I tokamaker följer partiklar gyrokinetiska rörelser.

2.1.1.2 Gyrokinetisk rörelse

När en ström sätts igång så inducerar den ett magnetfält. Detta magnetfält kan beskrivas genom Ampères lag, som lyder:

$$\nabla \times \vec{B} = \mu_0 \vec{J} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t}, \quad (2.3)$$

där ∇ är nablaoperatoren, \vec{B} är magnetfältet, μ_0 är permeabiliteten i vakuum, \vec{J} är strömmen, ϵ_0 är permittiviteten i vakuum och $\frac{\partial \vec{E}}{\partial t}$ är elfältets förändring över tid [23]. $\frac{\partial \vec{E}}{\partial t}$ är i vissa fall noll i en tokamak vilket gör att en term försvinner och magnetfältet enkelt kan beskrivas med följande ekvation:

$$\nabla \times \vec{B} = \mu_0 \vec{J}. \quad (2.4)$$

När en laddad partikel rör sig genom detta magnetfält så påverkas den av en kraft, Lorentzkraften, som ges av följande ekvation:

$$\vec{F} = q (\vec{E} + \vec{v} \times \vec{B}), \quad (2.5)$$

där q är partikelns laddning, \vec{E} är det elektriska fältet som partikeln färdas genom, \vec{v} är partikelns hastighet och \vec{B} är det magnetiska fält som partikeln färdas genom.

Då elfältet ibland är noll eller försumbart jämfört med $\vec{v} \times \vec{B}$ i en tokamak kan uttrycket för Lorentzkraften förenklas till:

$$\vec{F} = q\vec{v} \times \vec{B}. \quad (2.6)$$

Kraften är då riktad vinkelrätt mot hastigheten av partikeln och magnetfältet. I figur 2.2 går det att se två magnetfält som agerar. Dessa utövar varsin kraft på plasmat av partiklar för att hålla det i önskad form och bana runt tokamaken. Med antagandet att plasmat rör sig runt centrum av konstruktionen så skapar det poloidala magnetfältet en kraft på partiklarna som pekar in i plasmat vilket gör att plasmat förhåller sig i relativt cirkulära skivor. Det toroidala magnetfältet skapar en kraft som leder till rörelse runt centrum av plasmacylindern. Eftersom krafterna alltid är vinkelräta med hastigheten så är den konstant och ingen rörelseenergi förloras. Totalt resulterar detta i att plasmapartiklarna rör sig runt tokamaken i en sorts spiralbana längst med fältlinjerna i figur 2.2 [24]. Utöver detta roterar partiklarna runt fältlinjerna med en radie som kallas Larmorradien. Den defineras enligt:

$$r_L = \frac{mv_{\perp}}{|q|B}, \quad (2.7)$$

där m är massan hos en partiklarna, v_{\perp} är hastigheten vinkelrät mot fältlinjerna, $|q|$ är beloppet av laddningen hos partikeln och B är magnetfältet som partikeln färdas genom. Larmorradien bestämmer flera fysikaliska egenskaper hos plasmat så som mikroinstabiliteter som kan uppstå i plasmat. Dessa mikroinstabiliteter är det som huvudsakligen står för de energiförluster som finns i optimerade fusionsreaktorer [25].

2.1.2 Drifter av partiklar i plasma

Instabiliteter i energi- och partikeltransporten kan leda till både energiförluster i form av att partiklar flyger ut ur plasmat, och i värsta fall katastrofala fel som tvingar tokamaken att avbryta fusionsprocessen [26]. Det finns flera faktorer som påverkar stabiliteten, där en av dem är drifter av partiklar i plasmat. Med drifter menas att partiklarna rör sig på ett annorlunda sätt än önskat utifrån den teori som hittills har förklarats. Eftersom partiklarna är laddade så upplever de krafter från el- och magnetfält enligt tidigare stycke. Variation av dessa kan dock leda till partikeldrifter och asymmetrier i plasmat, vilket betyder att partiklarnas rörelser skiljer sig från magnetlinjerna som visas i figur 2.2. De mest markanta drifterna när det kommer till plasmat i tokamakar är $\vec{E} \times \vec{B}$ -, ∇B -, och kurvaturdrift [27].

$\vec{E} \times \vec{B}$ -drift är driften som uppstår hos en partikel i närvaro av en konstant elektrisk kraft $q\vec{E}$. Även om elfältet generellt sett är extremt litet, vilket är varför det räknades bort i avsnitt 2.1.1.2, så skapar de ändå lite drift. Driften beskrivs av:

$$\vec{v}_E = \frac{\vec{E} \times \vec{B}}{B^2}, \quad (2.8)$$

där \vec{E} är elfältet och \vec{B} är magnetfältet. Eftersom driften är oberoende på laddning och massa så kommer både elektroner och joner ha samma drift. Därför skapar denna

drift inte en ström av elektroner genom joner eller vice versa, utan det resulterar i en flödes- och positionsförändring av hela plasmat [27].

Då magnetfältet i tokamaken inte går att hålla helt homogent, vilket kommer från att elfältet existerar på små skalor och inte är konstant, så uppstår en drift som baserar sig i gradienten av magnetfältet, ∇B -drift. På grund av inhomogenitet av magnetfältet så påverkas en partikel olika beroende på position i plasmat, och därifrån kommer driften. För en partikel i plasmat skrivs driften som

$$\vec{v}_{\nabla B} = \frac{mv_{\perp}^2}{2qB^3} \vec{B} \times \nabla B, \quad (2.9)$$

där m är massan hos partikeln, v_{\perp} är hastigheten vinkelrät mot magnetfältslinjen, q är laddningen hos partikeln, \vec{B} är magnetfältet och ∇B är gradienten hos magnetfältet [27].

När en partikel färdas längst krökade magnetfältslinjer så ändras hela tiden hastigheten vinkelrät, \vec{v}_{\perp} , och parallellt, \vec{v}_{\parallel} , med magnetfältslinjerna. Detta skapar en centrifugalkraft som leder till kurvaturdrift [27]. Driften beskrivs enligt:

$$\vec{v}_c = \frac{mv_{\parallel}^2}{qB^2} \vec{B} \times \nabla_{\parallel} \vec{b}, \quad (2.10)$$

där $\nabla_{\parallel} \vec{b}$ är kurvaturen av magnetfältslinjerna.

2.1.3 Turbulent transport

Turbulent transport innebär transport av partiklar i en gas, vätska eller plasma baserat på kaotisk rörelse [28]. I en tokamak står turbulent transport för majoriteten av värme- och partikeltransporten. För plasma sker turbulent transport på grund av mikroinstabiliteter i plasmat såsom ändringar av den elektrostatiska potentialen, densitet eller temperatur [5]. Mikroinstabiliteterna kan i sin tur uppkomma som en följd av $\vec{E} \times \vec{B}$ -driften.

2.1.4 Instabiliteter

Det finns flera relevanta instabiliteter i plasmat och en del är mikroinstabiliteterna. De viktigaste bland dem är i sin tur jontemperaturgradienten, elektrontemperaturgradienten och fångat elektronläge ("trapped electron mode") [29].

När ∇B -driften som förklarades i 2.1.2 agerar på plasmat så struktureras joner och elektroner olika i plasmat vilket ger upphov till temperaturgradienter hos både joner och elektroner. Tillräckligt stora temperaturgradienter skapar förutsättningar där mikroinstabiliteter kan växa i plasmat och skapa inhomogeniteter av partiklarna. Dessa inhomogeniteter i fälten gör att partiklarna i plasmat kan påverkas av krafter som skickar ut partiklarna ur plasmat, vilket är energiförluster [30]. Detta är hur både jontemperaturgradienten och elektrontemperaturgradienten fungerar. Fångat elektronläge innebär att elektroner har fastnat i en del av plasmat där fältet är svagare och inte färdas runt tokamaken.

2.2 QuaLiKiz-modellen

För att förstå hur partiklar i fusionsplasma beter sig behövs förståelse för mikroinstabiliteter i plasmat, eftersom dessa ger upphov till turbulens som i sin tur driver transport av värme, partiklar och momentum i tokamakens kärna. Denna förståelse fås bäst genom att simulera de olinjära kinetiska ekvationer som styr vilka transporter instabiliteterna ger upphov till. Problemet med sådana simulationer är att de är beräkningsmässigt kostsamma, vilket är anledningen till att QuaLiKiz-modellen utvecklats [10] [31].

Qualikiz är en kvasilinjär¹ numerisk modell som, bland annat, gyrokinetiskt beräknar egenvärdena för turbulenta mikroinstabiliteter i magnetisk innesluten fusion. QuaLiKiz-modellen har 15 inputparametrar, vilka anses vara de mest relevanta storheterna kopplade till magnetisk innesluten fusion. Den har även 64 outputparametrar, varav de 18 första representerar tillväxthastigheter och är de parametrar som kommer att undersökas i projektet.

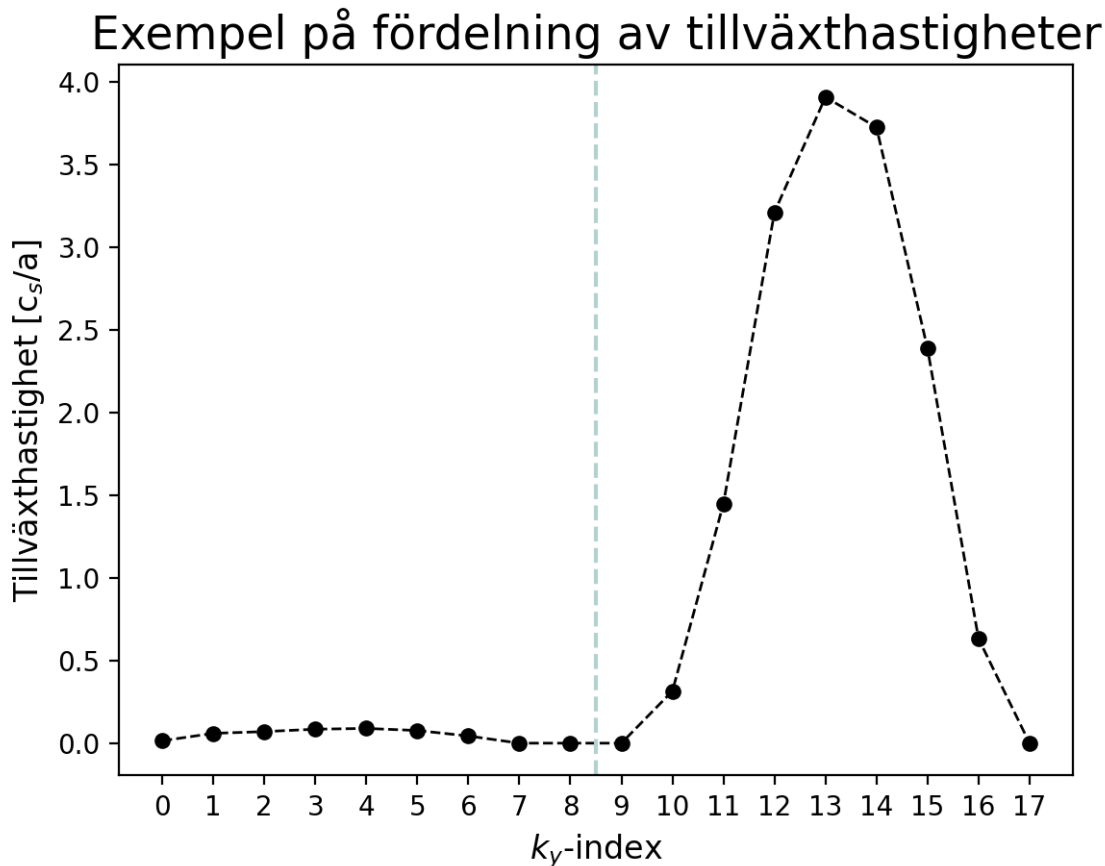
2.2.1 Inputparametrar för Qualikiz

Den numeriska modellen Qualikiz approximerar lösningar på differentialekvationer utifrån 15 inputparametrar. Dessa 15 inputparametrar representerar de mest relevanta storheterna som är verksamma fusion. Inputparametrarna beskrivs kortfattat under avsnittet Nomenklatur.

2.2.2 Outputparametrar för Qualikiz

De relevanta outputparametrarna för detta projekt är tillväxthastigheter som motsvarar den reella delen av egenvärdeslösningen för 18 olika spatiala skalor. De spatiala skalorna motsvarar vågnummer där de nio första vågnummer är associerade till jon-skalor och de nio senare vågnummer är associerade till elektron-skalor:

¹För Qualikiz-modellen betyder kvasilinjär att den förenklar beräkningarna för att bli mindre beräkningstung. En helt icke-linjär modell skulle behöva simulera plasmats varenda rörelse.



Figur 2.3: Figuren visar tillväxthastigheter för alla k_y -index för en slumpmässigt utvald nollskild instans i träningsdatasetet. Den vertikala linjen separerar jonskalorna (till vänster) från elektronskalorna (till höger).

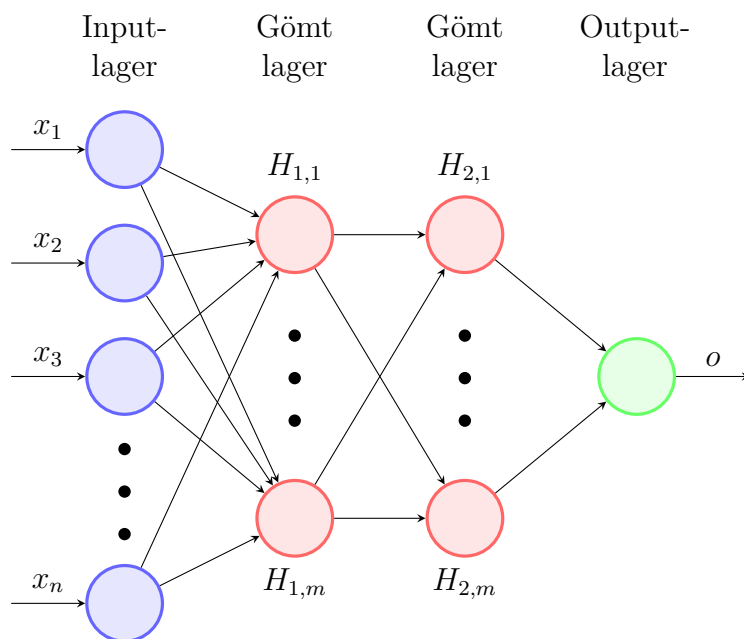
Tillväxthastigheterna visar alltså hur snabbt instabiliteten i magnetiskt innesluten fusion för olika spatiala skalor växer.

2.3 Neurala nätverk

Maskininlärning är ett fält inom artificiell intelligens där det fundamentala konceptet bakom maskininlärning är att använda stokastiska beräkningar för att skapa modeller som kan lära sig av och använda data för att utföra uppdrag [32]. Neurala nätverk är ett område inom maskininlärning som syftar till att skapa analyserande och förutsägande nätverk. Ett artificiellt neuralt nätverk försöker imitera hur biologiska neurala nätverk i organismers hjärnor fungerar och lär sig. Hädanefter kommer neurala nätverk att syfta på artificiella neurala nätverk [33].

2.3.1 Uppbyggnad

Ett neuralt nätverk består av ett inputlager, ett outputlager och ett eller flera latentlager, varje lager med ett bestämt antal latent noder se figur 2.4. En nod



Figur 2.4: En möjlig struktur på ett neuralt nätverk. I denna representation finns n inputparametrar och två gömda lager med m noder i varje, som sedan predikterar en output o .

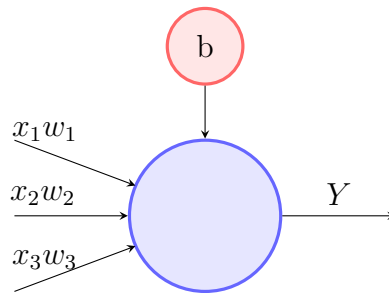
tar emot information bestående av så kallade vikter och bias från alla noder i det föregående lagret. Informationen manipuleras enligt en bestämd aktiveringsfunktion innan vikter och bias skickas vidare till nästa lager. Vikterna kan ses som vektorer som är associerade med specifika noder, och deras värden signalerar hur starkt kopplad en nod är till föregående noder. Biasen är en skalär som höjer eller sänker vikternas värden innan de manipuleras av aktiveringsfunktionen [34]. Detta kan, för en nod, uttryckas som:

$$y = \sum_{i=1}^m w_i \cdot x_i + b, \quad (2.11)$$

där w_i är vikterna associerade med värdena x_i på de föregående noderna och b är biasen [33]. Aktiveringsfunktionen manipulerar vikt- och biasvektorn och är det slutgiltiga steget innan en ny signal skickas ut ur noden. Den kan se olika ut beroende på vad som är syftet med manipulationen, men en av de vanligaste aktiveringsfunktionerna är ReLU. ReLU utför följande operation på de värden den tar in:

$$Y = \begin{cases} y, & \text{om } y > 0 \\ 0, & \text{annars} \end{cases}, \quad (2.12)$$

där y är värdet som går in i aktiveringsfunktionen, och Y är värdet som kommer ut [35]. Ekvation 2.11 och 2.12 kan tillsammans visualiseras med figur 2.5 som visar hur en nod tar emot inputdata i form av vikter och bias, och skickar ut en output y .



Figur 2.5: En nod i ett gömt lager på ett neuralt nätverk som tar emot insignaler x_i , vikter w samt en bias b och predikterar outputparametrar.

2.3.2 Träning

Neurala nätverk tillämpas på ett stort antal områden, exempelvis kan de utföra regressionsuppgifter. För att göra detta måste de tränas på data som är relevant för den aktuella uppgiften. För regressionsuppgifter innehåller träningsdata inputdata och outputdata, och den förlustfunktion β som används är oftast medelkvadratfel, MSE, som beskrivs nedan. Inputdata förs genom nätverket och skapar en prediktion, varefter det predikterade värdet jämförs med faktisk outputdata. Baserat på hur stort nätverkets fel är ändrar noderna sina vikter och bias för att förbättras inför nästa iteration [36]. MSE beräknar medelvärdet av felet i kvadrat och visas i ekvation 2.13. Variablerna n , V_i och P_i står för antalet prediktioner, det verkliga värdet respektive det predikterade värdet.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (V_i - P_i)^2 \quad (2.13)$$

Faktorer som bestämmer hur nätverket tränas som är oberoende av träningen kallas för hyperparametrar. Exempel på hyperparametrar är antalet noder i ett lager, hur många gånger (epoker) nätverket ska tränas på all data samt hur mycket av all data som ska föras genom nätverket innan vikter och bias uppdateras (batchstorlek) [33]. Ofta tränar nätverket på stora mängder kända input- och outputdata. Genom att göra detta blir nätverket bättre och bättre på att prediktera outputvärden baserade på inputparametrar. Om nätverket tränas för mycket på endast träningsdata så kan det bli överanpassat till träningsdatan, och därmed bli sämre på att prediktera generella fall. På grund av detta läggs en viss del av datasetet oftast undan: Denna kallas valideringsdata. Samtidigt som nätverket tränar på träningsdata, så testas det hur bra det är på att prediktera valideringsdata. Detta skapar en uppfattning om hur bra nätverket klarar andra fall än just de som nätverket tränar till.

2.3.3 Optimering och lokala minimum

Ett neuralt nätverk arbetar genom att försöka optimera en funktion på så kort tid som möjligt, vilket görs genom att använda en optimeringsalgoritm. En grundläggande optimeringsalgoritm kallas för "gradient descent" som, genom att förflytta sig stegvis längs en funktions gradients negativa riktning, hittar ett lokalt minimum

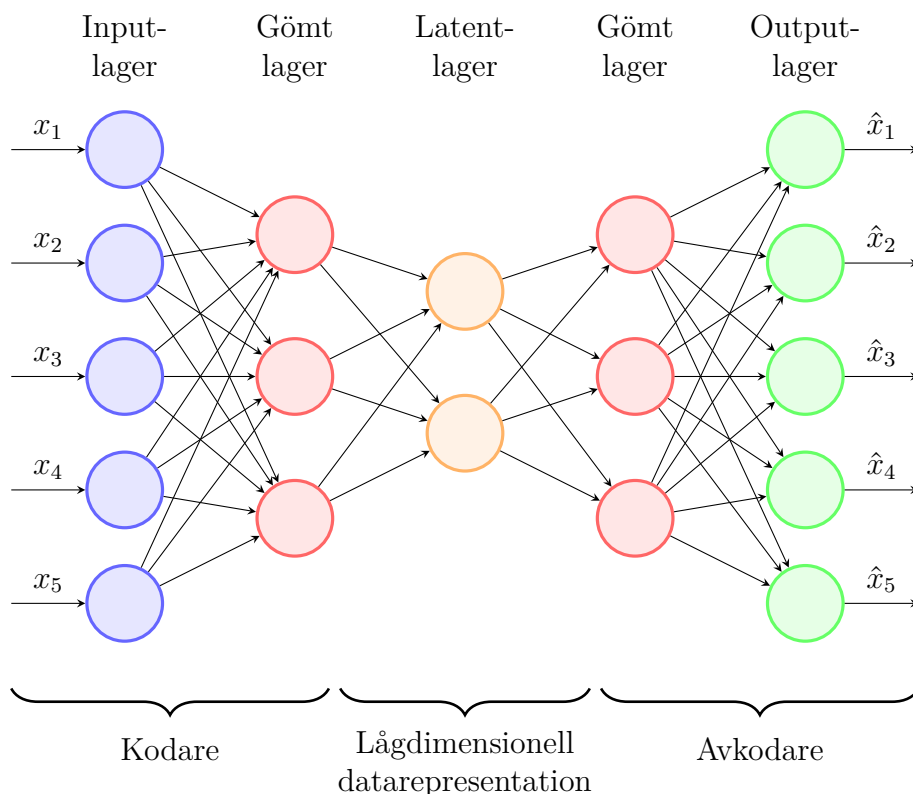
[37]:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \beta \nabla \lambda(\mathbf{x}_i), \quad (2.14)$$

där $\nabla \lambda(\mathbf{x})$ är gradienten av funktionen λ med avseende på vektorn \mathbf{x} . Algoritmen är välanvänd inom maskininlärning och $\nabla \lambda$ representerar gradienten av förlustfunktionen λ med avseende på ett neuralt nätverks vikter och bias \mathbf{x} . Syftet med algoritmen är att minimera förlustfunktionens värde genom att uppdatera nätverkets vikter och bias. Ett problem med optimeringsfunktionen är att den kan fastna i ett lokalt minimum, vilket förhindrar det neurala nätverket från att hitta den globala minimipunkten [37]. För att komma runt detta problem har anpassade versioner av "gradient descent"-algoritmen utvecklats, där Adam är en version. Adam använder sig av en adaptiv lärhastighet, vilket i praktiken innebär att steglängden inte endast beror på β utan även på ett antal andra parametrar [38].

2.3.4 Autokodare

En autokodare är en typ av neuralt nätverk vars syfte är att komprimera multidimensionell data i färre dimensioner utan att förlora en signifikant mängd information [39]. Autokodaren består av tre delar vilka är en enkodare, ett latent lager och en avkodare. Enkodaren komprimerar inputdatan i färre dimensioner och avkodaren återskapar inputdatan från dessa noder. Mellan dessa två lager finns ett latent lager där det latent lagret representerar det lager där inputdatan är som mest komprimerad, alltså med lägst antal dimensioner. Inputdatan blir sedan rekonstruerad från dess komprimerade form med avkodaren. Att kunna komprimera inputparametrar i färre noder är användbart i flera sammanhang. Istället för att skicka stora mängder data som tar utrymme och tid, kan den komprimerade datan skickas tillsammans med decodern [40]. Ett exempel på hur en enkel autokodare kan se ut visas i figur 2.6.



Figur 2.6: Ett exempel på hur en autokodare kan se ut. Kodaren består av inputlagret och ett gömt lager, och den latent representationen är tvådimensionell. Avkodaren består sedan av ett gömt lager, och de återskapade inputvärdena.

En autokodare som tränats på en viss mängd data kan användas på ny data av samma typ inom samma område utan att behöva tränas om då den har tränats till att vara en generell autokodare för den typen av data.

2.3.5 Svarta lådor

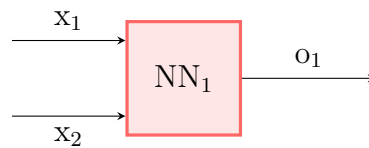
Det som är unikt med maskininlärningsalgoritmer är deras förmåga att lära sig av data och uppdatera sin struktur baserat på de mönster och samband som hittas av algoritmerna själva. De beteenden som uppvisas av en maskininlärningsmodell, som ett neuralt nätverk, är i allmänhet av en komplex natur med flertalet olinjära samband. Detta gör att modeller ofta är för komplexa för att en människa ska kunna spåra hur specifika inputparametrar relaterar till outputparametrar [41]. På grund av detta jämförs neurala nätverk ofta med svarta lådor. Det som går in och ut ur det neurala nätverket är tydligt, medan metoden på insidan är svårtolkad [41].

Att besluten som ett neuralt nätverk tar ofta inte kan förklaras ger upphov till både kunskapsluckor och etiska implikationer; när beslut som påverkar individer eller samhällsekonomin ska tas behöver de kunna underbyggas. Av denna anledning kan det finnas fördelar med att försöka göra en modell mer förklarlig [41]. Ett potentiellt sätt att öka en modells förklarlighet är bryta ner den i mindre beståndsdelar i förhoppningen om att bilda en uppfattning om hur modellens inputparametrar

relaterar till dess outputparametrar [11].

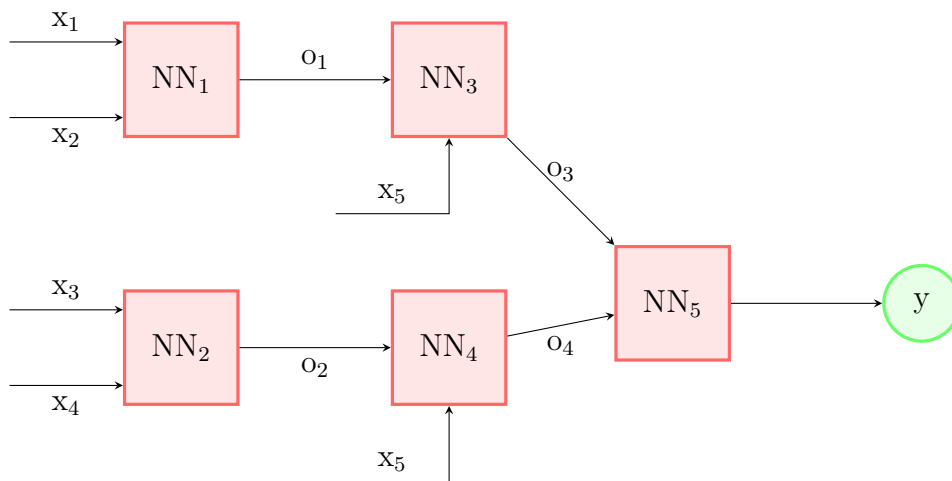
2.3.6 Grenmodeller

En grenmodell är en metod som kan användas för att para ihop olika inputparametrar och öka tolkbarheten av neurala nätverk. Grenmodeller består av en sammanhängande struktur av neurala nätverk, där varje neuralt nätverk tar in två inputparametrar och producerar en outputparameter. Denna struktur gör det möjligt att visualisera in- och outputparametrar i varje gren, och därmed till viss del förstå hur ett neuralt nätverk kommer fram till sina prediktioner. Ett exempel på hur en grenmodell kan se ut visas i figur 2.7.



Figur 2.7: Figur som visar en gren i en grenmodell. Ett neuralt nätverk har använts för att förutse o_1 genom x_1 och x_2 .

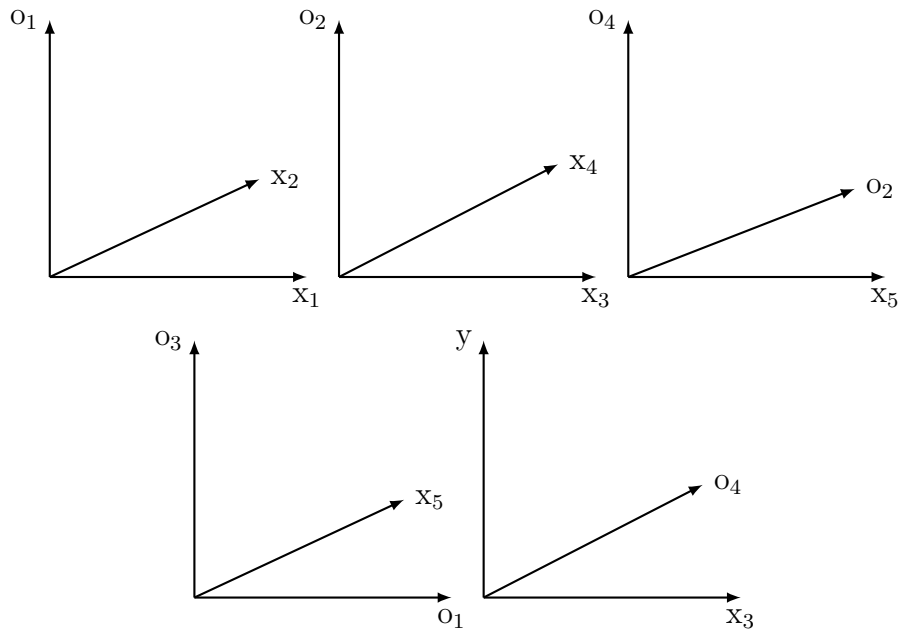
Istället för att sedan skicka in alla inputparametrar i det ursprungliga neurala nätverket så byts två inputparametrar ut mot ett mindre neuralt nätverks output. Detta görs för varje neuralt nätverk i grenmodellen fram till sista steget där outputparametrarna predikteras. Hur detta kan se ut visualiseras i figur 2.8.



Figur 2.8: En möjlig struktur på en grenmodell, här med fem unika inputs x_1, \dots, x_5 varav o_1, \dots, o_4 är förgreningar av inputsen där den sista grenen ger outputen y .

Genom att undersöka om det lilla neurala nätverket kan komprimera inputvariablerna bra och om helheten fungerar lika bra som innan så går det att ta fram hur mycket inputparametrarna är relaterade till varandra. Denna typ av nätverk kan indikera ifall vissa inputparametrar inte behöver mätas då de kan skapas av andra inputparametrar. Om de små nätverken utvecklas vidare enligt figur 2.8 så kan det

även visa mer exakt hur inputparametrarna relaterar till varandra, vilket är en förbättring då vanliga neurala nätverk inte visar detta. Figur 2.9 visar hur 3D plottar kan se ut för att hitta samband mellan inputparametrar och dess outputprediktioner.



Figur 2.9: Med hjälp av grafer som dessa går det att få en uppfattning om hur inputparametrar relaterar till varandra och till outputprediktioner.

3

Metoder och delresultat

3.1 Mätningmetodik

Alla mätningar som beskrivs i metoder har följt en grundläggande struktur gällande antalet mätningar som gjorts. I syfte att undvika att använda resultat från modeller som hamnat i icke-optimala lokala maxima, tränades varje modell flera gånger. Därefter valdes medelvärdet av förlustfunktionen för de bästa modellerna ut. På så sätt säkerställdes resultat av en mer robust karaktär än om varje modell endast hade tränats en gång. Denna metod gör att sannolikheten för att drabbas av problemet med lokala minima som beskrivs i 2.3.3 minskar.

Vidare har förlustfunktionen MSE använts vid all träning. Att valet föll på MSE beror på att funktionen lämpar sig för regressionsuppgifter [36]. Dessutom fanns en poäng i sig med att genomgående använda samma förlustfunktion för alla mätningar, eftersom det förenklade jämförelser av resultaten.

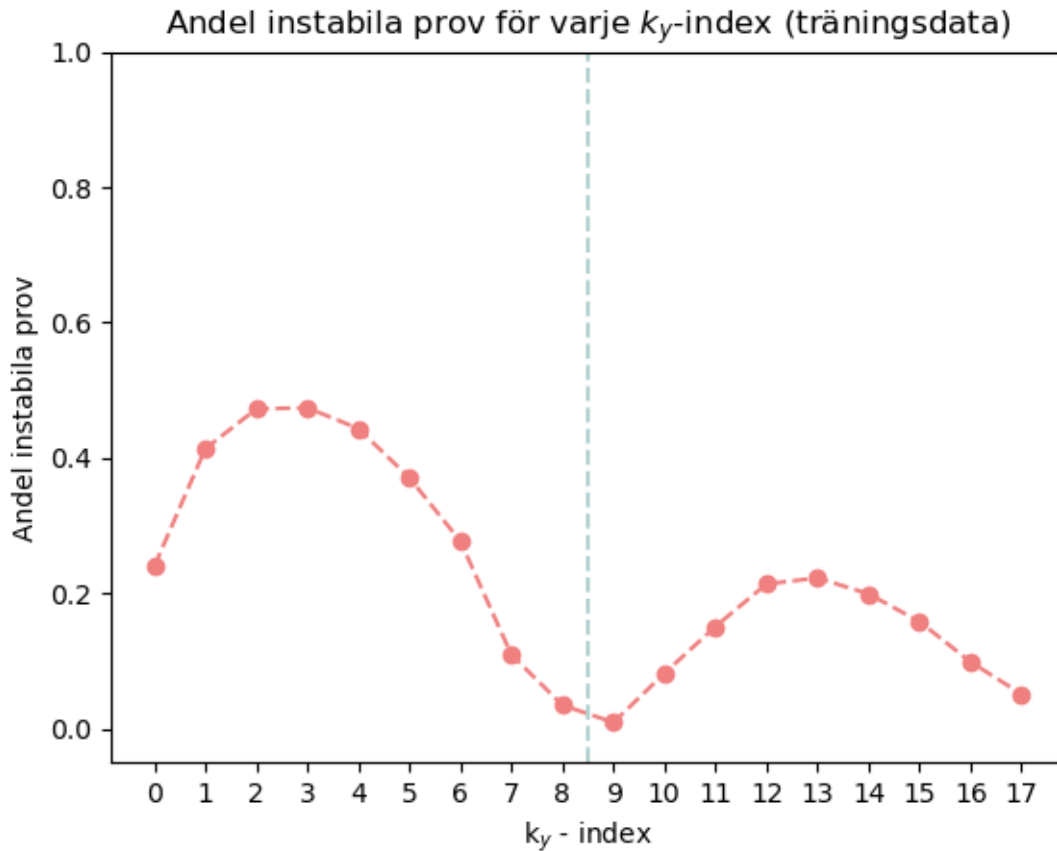
3.2 Förbehandling av data

De data som skulle undersökas var uppdelade i två delar: ett tränings- och valideringsdataset, och ett testdataset. Innan någon form av maskininlärning introducerades i projektet visualiserades datan på olika sätt, för att få ett grepp om hur den var distribuerad och hur det eventuellt skulle kunna komma att påverka resultaten.

3.2.1 Outputparametrar

Outputparametrarna var de parametrar som undersöktes först. Totalt innehöll dataseten 64 outputparametrar, varav endast de 18 första representerade tillväxthastigheter och var relevanta för projektet. Av denna anledning filtrerades parametrarna 19 till 64 ut innan datans egenskaper undersöktes. Dessa outputparametrar som representerar tillväxthastigheter kommer att motsvara k_y -index.

Prov där alla tillväxthastigheter hade värdet noll kallades för stabila prov, eftersom instabiliteter inte växer i plasmat om tillväxthastigheten är noll [14]. Nollskilda prover kallades för instabila prover. Andelen nollskilda prover för varje k_y -index i tränings- och valideringsdatasetet visualiseras i figur 3.1.

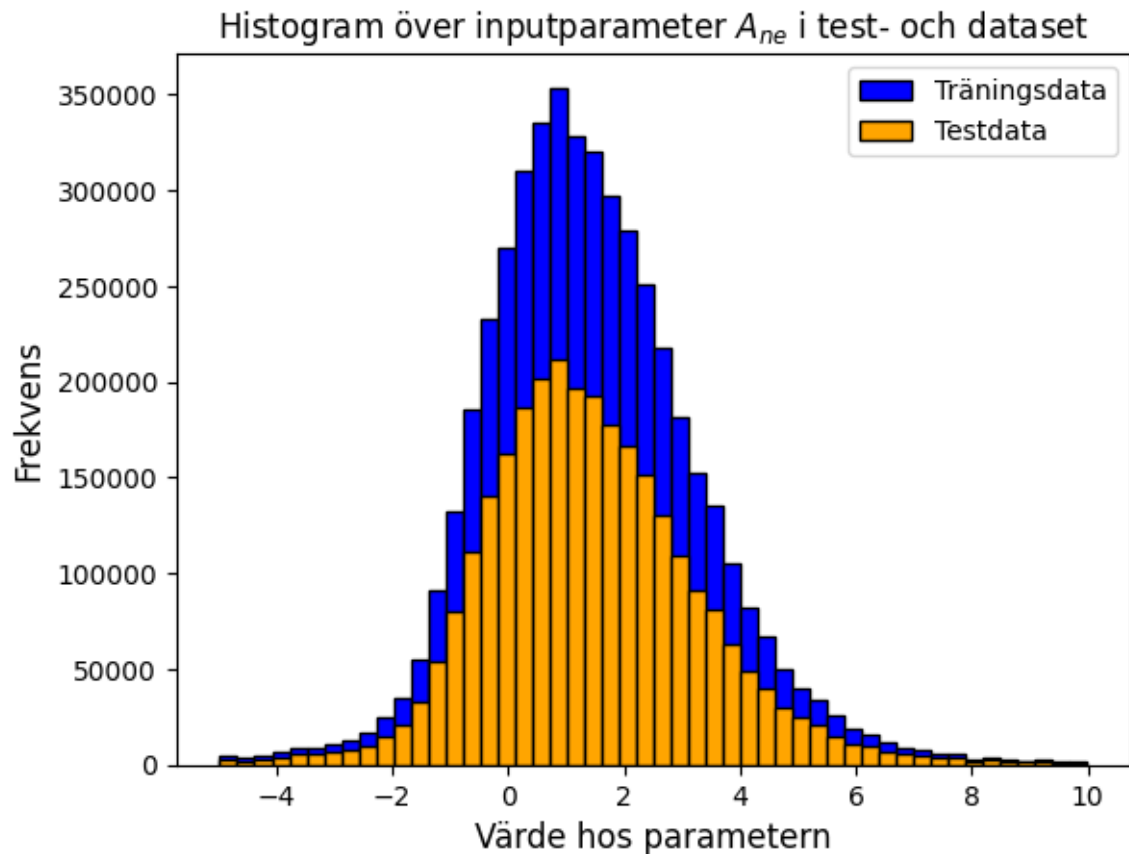


Figur 3.1: Figuren visar andelen nollskilda tillväxthastigheter för varje k_y -index. Till vänster om den vertikala linjen syns de k_y -index som relaterar till jonskalor, och till höger syns elektronskalorna.

Det gjordes även en jämförelse mellan distributionen av outputparametrarna i både tränings- och testdatasetet. Alla k_y -index i de två dataseten var relativt lika fördelade. Vidare var tillväxthastigheternas värden positiva (≥ 0) och skalades linjärt så att de erhöll värden mellan noll och ett. Detta gjordes för att anpassa värdena till de beräkningar som skulle utföras av maskininlärningsmodellerna.

3.2.2 Inputparametrar

De 15 inputparametrarna visualiserades med hjälp av histogram, för att ta reda på deras respektive distributioner i tränings- och testdataseten. Inputparametrarna och deras distributioner varierade inte nämnvärt mellan dataseten, se figur 3.2. Detta visade att dataseten var uppdelade på ett balanserat sätt.



Figur 3.2: Figuren illustrerar hur träningsdata- och testdatasetet har liknande uppdelning men olika antal datapunkter. Observera att träningsdatan i figuren även innefattar de värden som täcks av testdatan. Se figur A.2 för resterande inputparametrar.

Inputparametrarna antog både positiva och negativa värden, och för att behålla denna egenskap skalades de linjärt så att de erhöll värden mellan -1 och 1.

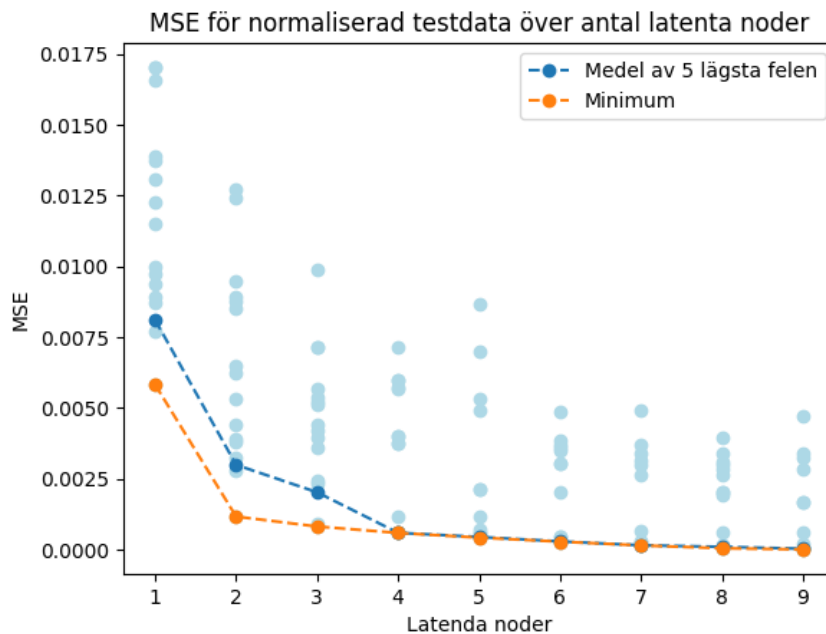
3.3 Autokodare

Efter att all data analyserats och normaliserats skapades en autokodare. Syftet med autokodaren var att undersöka hur mycket det gick att komprimera outputparametrarna utan att gå miste om viktiga egenskaper hos datan. Denna information var avgörande för när grenmodellen senare konstruerades, i och med att grenmodellen komprimerar inputdatan. Eftersom autokodaren visar att all nödvändig information för att återskapa outputdatan kan bevaras även i en komprimerad form, ger dess resultat en indikation på hur mycket inputdatan kan komprimeras i grenmodellen. Att resultatet av autokodare kan användas i grenmodellen bygger på att inputdatan kan med god precision prediktera outputdatan [42]. Då detta arbete fokuserade på jonskalorna så gjordes autokodare för de nio tillväxthastigheter som representerade jonskalorna.

På grund av att avsikten med användningen av autokodare inte var att uppnå en så exakt rekonstruktion som möjligt, utan att ta reda på vid vilken dimension på den latent vektorn som förlustfunktionen blev tillräckligt liten, så användes en väldigt enkel modell. Modellen bestod av ett lager som tog emot inputdata och ett som rekonstruerade dessa. Däremellan lades ett latentlager som i figur 2.6. Det latent lagret representerar den n -dimensionella vektor som skulle representera de komprimerade tillväxthastigheterna.

Aktiveringsfunktionen ReLU användes både mellan inputlagret och det latent lagret samt mellan det latent lagret och rekonstruktionslagret. Anledningen till att det var passande att använda ReLU som aktiveringsfunktion för rekonstruktionslagret var att tillväxthastigheterna endast bestod av tal i \mathbb{R}_0^+ . Vidare definierades följande hyperparametrar: batchstorlek = 4096; $\beta = 10^{-3}$; antal epoker = 10.

Modellen som beskrivits ovan tränades med varierande dimensioner på det latent lagret. Dimensionerna på det latent lagret varierades mellan ett och nio. Därefter jämfördes medelvärdet av felet mot antalet latent noder. Det var inte modellen som hade det minsta felet som eftersöktes, utan den som hade en så lågdimensionell latent vektor som möjligt och samtidigt ett tolerabelt värde på förlustfunktionen. Figur 3.3 visar felet mot antalet latent noder.



Figur 3.3: Figur över förluster för testdata vid olika antal latent noder då 9 lösningsparametrar skulle predikteras.

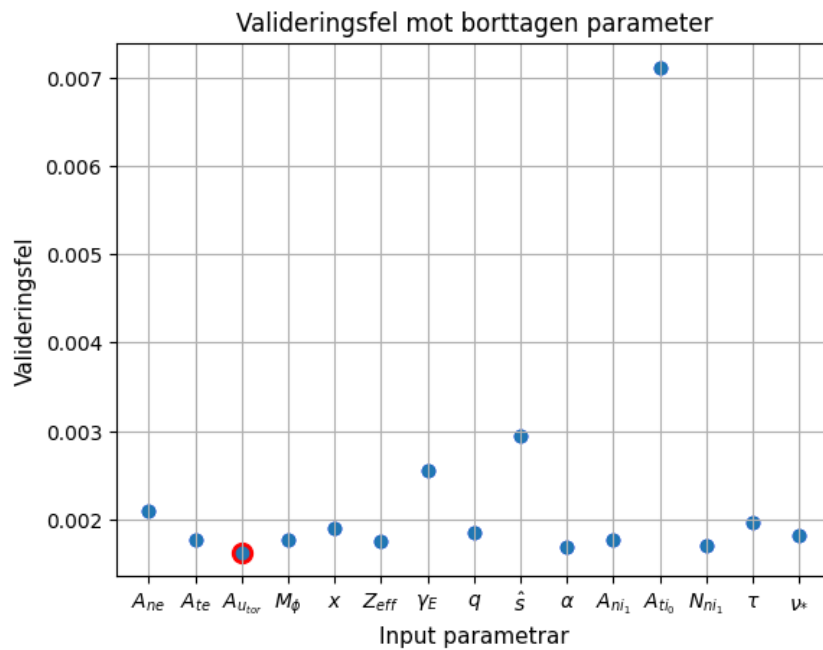
Felet hos den modell med lägst fel sjunker snabbt mellan att ha en nod jämfört med två, sedan börjar det plana ut. Därför valdes två latent noder som ett bra antal för att kunna representera datan.

3.4 Reduktion av inputparameterrymd

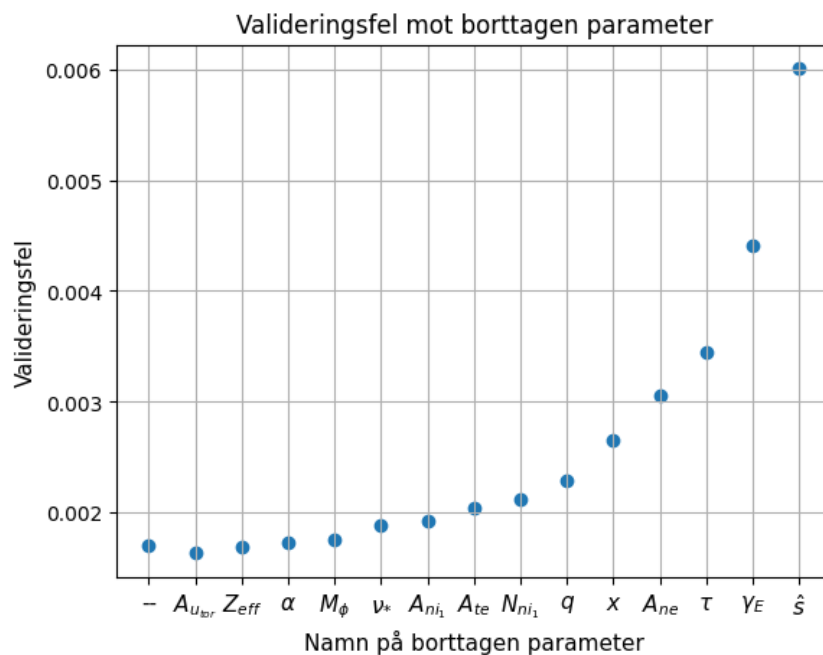
Med 15 inputparametrar fanns det ett väldigt stort antal möjliga grenmodeller, samtidigt som en grenmodell med så många parametrar kan bli svår att analysera och diskutera. Av den orsaken fanns det ett behov av att minska antalet inputparametrar i modellen innan arbetet med grenmodeller påbörjades, vilket gjordes genom att skapa ett neuralt nätverk som predikterade tillväxthastigheterna med hjälp av inputparametrarna.

Ett neuralt nätverk bestående av två dolda lager med 100 latent noder i varje lager och ReLU som aktiveringsfunktion skapades med mål att hitta de inputparametrar som påverkade valideringsfelet minst, och sedan plocka bort dessa en och en. Alla körningar av detta nätverk gjordes med en epok och 1000 i batchstorlek. Först använde nätverket alla 15 inputparametrar för att återskapa de första nio tillväxthastighetsspektrumen, där valideringsförlusten för denna modell sparades som ett basvärde på hur bra inputparametrarna är på att återskapa tillväxthastigheterna.

Samma nätverksarkitektur användes sedan för att prediktera tillväxthastigheterna med hjälp av endast 14 inputparametrar. Vid varje körning togs en ny inputparameter bort, och därefter jämfördes valideringsfelet för mätningarna. Den inputparameter som påverkade valideringsfelet minst då den inte användes blev bortplockad. Denna procedur upprepades genom att en av de resterande 14 parametrarna togs bort, och så vidare tills det endast fanns en inputparameter kvar. Innan en parameter blev bortplockad gjordes mätningarna fem gånger där medelvärdet av de två bästa körningarna valdes ut med motivering enligt 3.1. I figur 3.4 visualiseras MSE för valideringsdata.



Figur 3.4: Medelvärde av valideringsfel(MSE) jämfört med om någon av de olika parametrarna tas bort. Den parameter som ger lägst valideringsfel är rödmarkerad och togs bort för nästa körning. Se A.1 för figurer från alla körningar.



Figur 3.5: Figuren visar i vilken ordning de olika parametrarna har tagits bort där y-axeln är valideringsfel efter borttagen parameter och x-axeln är den borttagna parametern. Vid den första punkten har ingen parameter blivit borttagen.

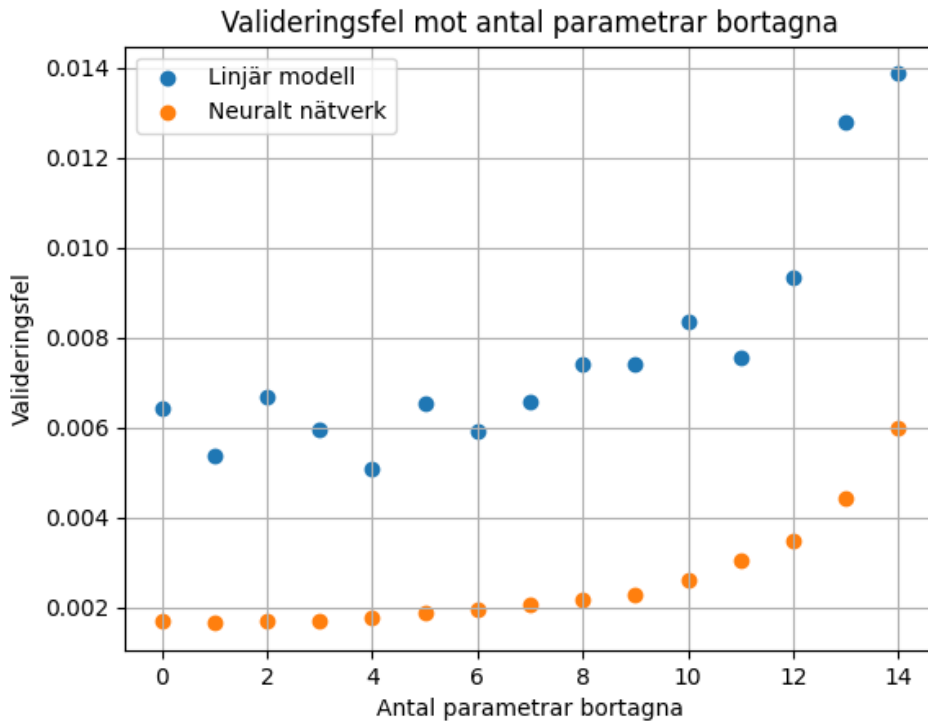
Figur 3.4 visar hur $A_{u_{tor}}$ är parametern som har minst påverkan på felet och därav minst viktig för att kunna återskapa tillväxtspektrumen medan A_{ti_0} har betydligt

större påverkan än någon annan parameter. Om man jämför figur 3.4 med figur A.1n kan man se att valideringsfelet blir 0.007 om alla parametrar utom A_{ti_o} används för att prediktera tillväxtspektrumen medan om endast A_{ti_o} används blir felet 0.006. Detta visar att parametrarna har väsentligt olika stor påverkan för att skapa outputen. Figur 3.5 beskriver 'vägen' i vilket de olika parametrarna har tagits bort och då även hur viktiga¹ de är för att återskapa tillväxtspektrum.

Utifrån detta resultat valdes de sex bästa parametrarna att användas för grenmodellen då resterande hade en relativt låg påverkan på valideringsfelet. De är alltså: x , A_{ne} , τ , γ_E , \hat{s} , A_{ti_o} .

Sedan skapades en linjär modell där samma metodik som beskrivits ovan för det neurala nätverket upprepades. Syftet med detta var att undersöka om det var nödvändigt att använda neurala nätverk, eller om en linjär modell kunde prediktera tillväxthastigheterna lika bra. Figur 3.6 visar skillnaden i valideringsfel. Det neurala nätverket har lägre valideringsfel oavsett antalet parametrar som används. Detta visar att det neurala nätverket är bättre på att prediktera tillväxthastigheterna än det linjära. I och med det är det relevant att använda grenmodeller för att undersöka parametrarna.

¹Värt att notera att parametrarna har tagits bort i ordningen där de har minst påverkan när de används tillsammans med alla tidigare parametrar. Detta betyder att en parameter som tas bort inte nödvändigtvis är den som har minst påverkan då alla parametrar används eftersom parametrarna påverkar varandra till någon grad.



Figur 3.6: Jämförelse med genomsnitt av valideringsfel(MSE) mellan den linjära modellen och det neurala nätverket

3.5 Grenmodeller

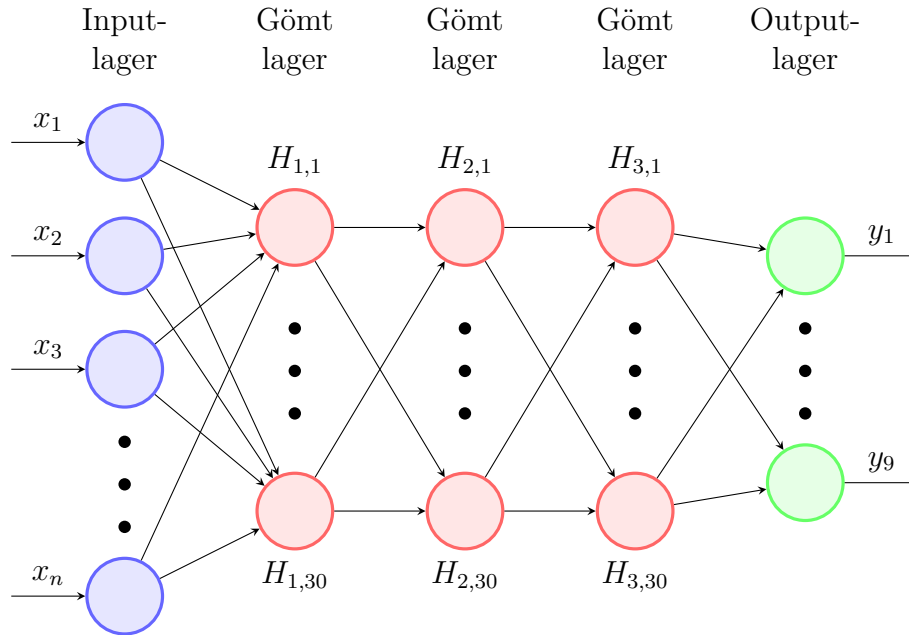
När de viktigaste inputparametrarna definierats påbörjades arbetet med att förgrena dem. I avsnitt 3.4 redovisas att sex inputparametrar valdes ut för den slutgiltiga analysen, och det var dessa som sedan ingick i grenmodellerna. Vidare uppdragas det i avsnitt 3.3 att det är möjligt att komprimera tillväxthastigheterna för jonskalorna från nio till två dimensioner utan att förlora en signifikant mängd information. Utifrån denna information togs beslutet att inputparametrarna i det grenade nätverket skulle komprimeras till en vektor i \mathbb{R}^2 innan tillväxthastigheterna predikterades. Detta innebar i praktiken att inputparametrarna skulle delas upp i två grövre grenar.

Syftet med användningen av grenmodeller var att hitta en kombination av inputparametrar som stoppades in parvis i neurala nätverk och producerade endimensionell outputdata enligt figur 2.8. Detta gjordes för att frigöra möjligheten att ta reda på hur inputparametrarna relaterade till tillväxthastigheterna genom att visualisera input- och outputdata från dessa nätverk.

3.5.1 Generella metoder för grenmodellerna

Alla konfigurationer av grenstrukturer kördes tio gånger, varefter de tre modellerna med lägst MSE för valideringsdata valdes ut, och jämförelser gjordes baserat

på medelvärdet av MSE för dessa modeller. Samma sak gäller för referensmodellerna. Vidare tränades varje modell under tre epoker, med batchstorlek = 1024 och $\beta = 10^{-3}$. En ytterligare gemensam nämnare var att delnätverket som predikerade tillväxthastigheterna alltid var uppbyggda som i figur 3.7. Dessutom användes ReLU som aktiveringsfunktion mellan alla lager, förutom för outputprediktionerna där linjär aktivering användes.



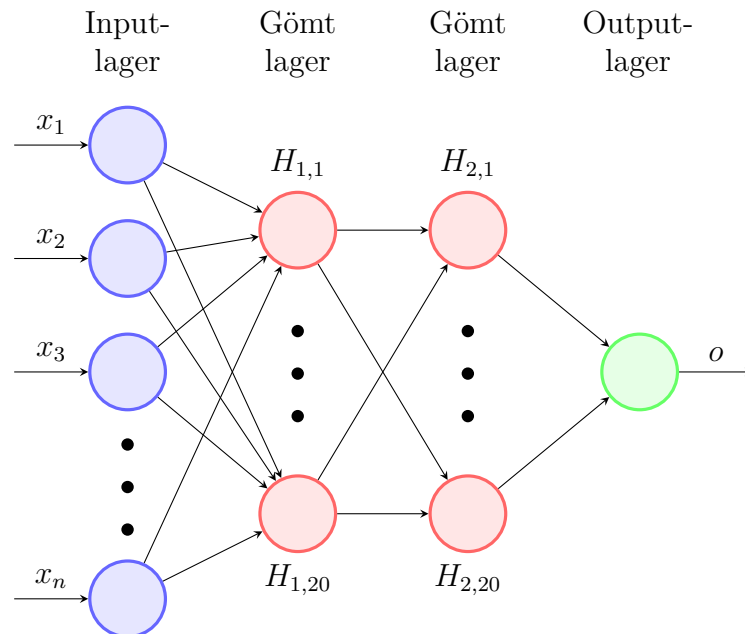
Figur 3.7: Strukturen för det neurala nätverket som predikerade tillväxthastigheterna. Nätverket hade tre gömda lager med 30 latent dimensioner. Pilarna representerar vikter.

Målet med grenmodellerna var att hitta en struktur som var så simpel som möjligt, utan att göra avkall på noggrannheten i för hög grad. Som referens för vad som ansågs vara tillräckligt hög noggrannhet användes en modell där inputparametern x tagits bort. I tabell 3.1 redovisas medelvärdet av MSE standardavvikelsen σ för referensmodellen samt för en modell där alla sex inputparametrar använts för att prediktera tillväxthastigheterna. Även en skattning av standardavvikelsen σ för MSE användes för alla modeller. Denna bestod av standardavvikelsen för de tre bästa modellerna i varje konfiguration, och användes i syfte att uppskatta om skillnaden i fel mellan den bästa modellen och de andra var statistiskt signifikant. I följande avsnitt avser “standardavvikelse” denna skattning.

Referensmodell	Medelvärde MSE	Standardavvikelse σ
Alla 6 inputparametrar använda	$2.1749 \cdot 10^{-3}$	$4.1 \cdot 10^{-6}$
x borttagen	$2.5061 \cdot 10^{-3}$	$8.4 \cdot 10^{-7}$

Tabell 3.1: Medelvärdet för MSE för modeller där sex respektive fem inputparametrar använts för att prediktera tillväxthastigheterna.

Det nätverk som predikerade grenarnas outputdata hade till en början samma struktur som nätverket som predikerade tillväxthastigheter. Träningen var dock tidskrävande, och precisionen ökade inte nämnvärt då ett simplare nätverk användes för grenarna. Av denna anledning togs beslutet att minska komplexiteten något för detta nätverk, se figur 3.8.



Figur 3.8: Strukturen för det neurala nätverket som predikerade mellanliggande outputs. Nätverket hade två gömda lager med 20 latent dimensioner, och en outputvektor. Pilarna representerar vikter.

3.5.2 Metod för att hitta en optimal grenmodell

Det första som gjordes var att försöka hitta en struktur med två grenar, med så få inputparametrar som möjligt i varje gren. När detta hade gjorts delades inputparametrarna i de två grövre grenarna upp i fler nätverk. I följande avsnitt beskrivs först hur reduktionen till två grövre grenar utfördes, och därefter hur den slutgiltiga grenstrukturen hittades.

3.5.2.1 Första reduktionen

Den första förgreningen gjordes genom att dela upp de sex inputparametrarna i två högar med fem parametrar i varje, se figur 3.9a. Grenarna, som här avser de små neurala nätverken, fick inte vara identiska, och dessutom fick en parameter x_n endast förekomma en gång i varje gren. Givet dessa villkor fanns det $\binom{6}{5} = 6$ unika kombinationer för en gren, och 15 unika sätt att kombinera dem parvis. I tabell 3.2 visas medelvärdet av MSE för de fyra förgreningar som hade lägst MSE samt för den med högst MSE. I denna och följande tabeller i detta avsnitt benämns grenarna som $G_{n,i,j}$, där n representerar vilken förgrening som gjorts och i hänvisar till en

specifik grenstruktur. Index j definierar vilken gren på rad i som är aktuell; om två mindre grenar härstammar från samma, större gren kommer $j = k1, \dots, kn$.

Tabell 3.2: Grenstrukturer och medelkvadratfel MSE_i för de fyra bästa samt den sämsta strukturen med fem inputparametrar i båda grenarna, samt standardavvikelser σ_i för dessa. Kombinationerna är sorterade efter MSE_i , och den bästa kombinationen är markerad med fetstil. Resultat för resterande konfigurationer visas i tabell B.1.

i	$(G_{1,i,1}), (G_{1,i,2})$	Medelvärde MSE_i	σ_i
1	$(\mathbf{A}_{ne}, \mathbf{x}, \gamma_E, \hat{\mathbf{s}}, \mathbf{A}_{ti_0}), (\mathbf{A}_{ne}, \mathbf{x}, \hat{\mathbf{s}}, \mathbf{A}_{ti_0}, \tau)$	$2.2894 \cdot 10^{-3}$	$5.7 \cdot 10^{-6}$
2	$(A_{ne}, x, \gamma_E, \hat{\mathbf{s}}, A_{ti_0}), (A_{ne}, \gamma_E, \hat{\mathbf{s}}, A_{ti_0}, \tau)$	$2.2902 \cdot 10^{-3}$	$5.3 \cdot 10^{-6}$
3	$(A_{ne}, x, \gamma_E, \hat{\mathbf{s}}, \tau), (A_{ne}, x, \hat{\mathbf{s}}, A_{ti_0}, \tau)$	$2.2917 \cdot 10^{-3}$	$6.4 \cdot 10^{-6}$
4	$(A_{ne}, x, \gamma_E, \hat{\mathbf{s}}, \tau), (A_{ne}, \gamma_E, \hat{\mathbf{s}}, A_{ti_0}, \tau)$	$2.2933 \cdot 10^{-3}$	$6.0 \cdot 10^{-6}$
...
15	$(A_{ne}, x, \gamma_E, \hat{\mathbf{s}}, A_{ti_0}), (A_{ne}, x, \gamma_E, A_{ti_0}, \tau)$	$2.4955 \cdot 10^{-3}$	$9.6 \cdot 10^{-7}$

Det framgår att struktur 1, där $A_{ne}, x, \hat{\mathbf{s}}$ och A_{ti_0} fanns i båda grenarna gav lägst MSE. Strukturen med högst MSE hade en gren som var identisk med en gren i den bästa förgreningen, medan $\hat{\mathbf{s}}$ var utbytt mot γ_E i den andra grenen. Vidare gällde för 2, 3 och 4 att $|MSE_1 - MSE_i| < \sigma_i$. Resultat för alla 15 konfigurationer kan ses i appendix B.

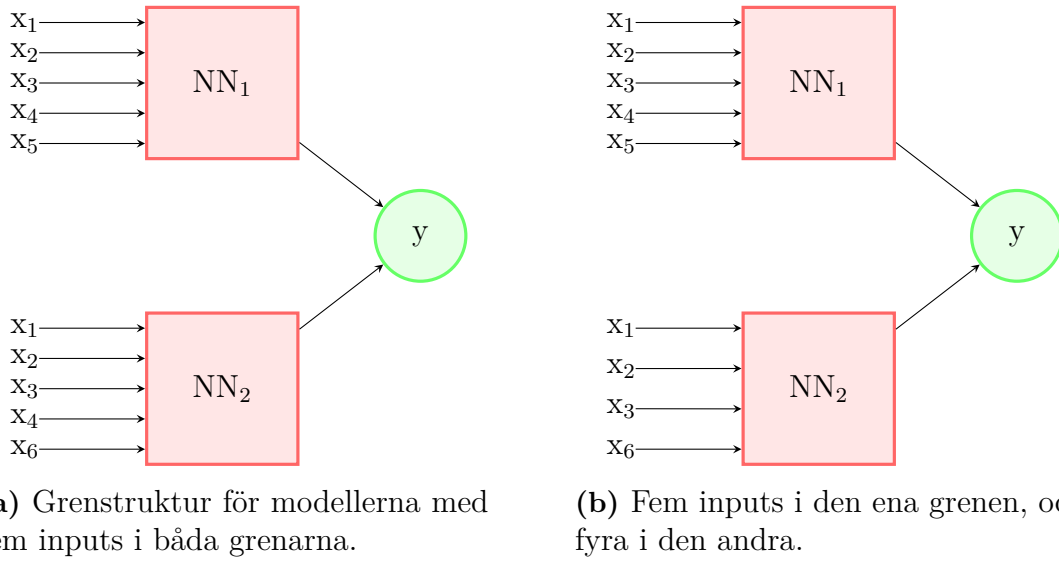
3.5.2.2 Andra reduktionen

Steg två i arbetet med grenmodeller var att reducera antalet inputparametrar i en av grenarna till fyra stycken, se figur 3.9b. Strukturen på $G_{1,i,1}$ var densamma för modellerna med högst och lägst MSE_i , vilket indikerade att parametrarna i $G_{1,i,2}$ hade större betydelse för valideringsfelet än de i $G_{1,i,1}$. Med hänsyn till detta faktum och den begränsade beräkningskraften, togs beslutet att ytterligare reduktion av parametrar skulle testas för strukturerna 1 och 2 i föregående avsnitt.

De nya förgreningarna gjordes genom att en inputparameter plockades bort i en gren, medan den andra grenen hölls intakt. Därefter gjordes samma sak med den motsatta grenen. De strukturer som saknade en av de sex inputparametrarna efter denna procedur togs bort. Från de två ursprungliga konfigurationerna skapades 15 nya, unika förgreningar. I tabell 3.3 redovisas MSE för konfigurationerna med lägst respektive högst MSE.

Tabell 3.3: Medelkvadratfel MSE_i för den bästa och sämsta grenstrukturen med fem inputparametrar i en gren och fyra i den andra samt standardavvikelser σ_i för dessa. Kombinationen med lägst MSE är markerad med fetstil. Resultat för resterande konfigurationer visas i tabell B.2.

i	$(G_{2,i,1}), (G_{2,i,2})$	Medelvärde MSE_i	σ_i
1	$(\mathbf{A}_{ne}, \mathbf{x}, \gamma_E, \hat{\mathbf{s}}, \mathbf{A}_{ti_0}), (\mathbf{A}_{ne}, \hat{\mathbf{s}}, \mathbf{A}_{ti_0}, \tau)$	$2.3121 \cdot 10^{-3}$	$5.4 \cdot 10^{-6}$
...
15	$(A_{ne}, x, \hat{\mathbf{s}}, A_{ti_0}, \tau), (A_{ne}, x, \gamma_E, A_{ti_0})$	$2.6372 \cdot 10^{-3}$	$8.3 \cdot 10^{-6}$



Figur 3.9: I figurerna visualiseras grenstrukturerna för de två första förgreningarna.

Den förgrening som hade lägst MSE för en grenstruktur med fem inputparametrar i ena grenen och fyra i den andra var den där $G_{2,1,2}$ var identisk med $G_{1,i,2}$ för den första grenstrukturen. Den andra grenen bestod av inputparametrarna A_{ne} , \hat{s} , A_{ti_0} och τ . Denna grenstruktur var den enda som var gemensam mellan de två strukturerna som används för reducering; det skulle kunna tolkas antingen som att x tagits bort från den bästa strukturen, eller att γ_E tagits bort från den näst bästa. Vidare gällde för alla andra kombinationer att $|\text{MSE}_i - \text{MSE}_{\text{bäst}}| > 2\sigma_i$. I tabell B.2 redovisas MSE för alla konfigurationer.

3.5.2.3 Sista reduktionen

Proceduren där en inputparameter åt gången togs bort upprepades, och grenstrukturen med lägst MSE_i från den andra reduktionen användes. Först testades kombinationer med fyra inputparametrar i båda grenarna; alltså togs en parameter bort från $G_{2,1,1}$. Kombinationerna med lägst respektive högst MSE_i visas i tabell 3.4. Därefter skapades en struktur där $G_{2,1,1}$ förblev intakt, medan en parameter togs bort från $G_{2,1,2}$. I tabell 3.5 redovisas kombinationerna med högst respektive lägst MSE_i givet dessa strukturer.

Tabell 3.4: Grenstrukturer och medelkvadratfel MSE_i för strukturerna med fyra inputparametrar i båda grenarna, samt standardavvikelser σ_i för dessa. Kombinationerna är sorterade efter MSE_i , och den bästa kombinationen är markerad med fetstil.

i	$(G_{3,i,1}), (G_{3,i,2})$	Medelvärde MSE_i	σ_i
1	$(\mathbf{A}_{ne}, \mathbf{x}, \gamma_E, \hat{\mathbf{s}}), (\mathbf{A}_{ne}, \hat{\mathbf{s}}, \mathbf{A}_{ti_0}, \tau)$	$2.4149 \cdot 10^{-3}$	$4.6 \cdot 10^{-6}$
2	$(x, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, \hat{s}, A_{ti_0}, \tau)$	$2.4453 \cdot 10^{-3}$	$6.8 \cdot 10^{-6}$
3	$(A_{ne}, \mathbf{x}, \gamma_E, A_{ti_0}), (A_{ne}, \hat{\mathbf{s}}, A_{ti_0}, \tau)$	$2.5386 \cdot 10^{-3}$	$3.4 \cdot 10^{-6}$

Tabell 3.5: Grenstruktur och medelkvadratfel MSE_i för strukturerna med fem inputparametrar i en gren och tre i den andra, samt standardavvikelser σ_i för dessa. Kombinationerna är sorterade efter MSE_i , och den bästa kombinationen är markerad med fetstil.

i	$(G_{4,i,1}), (G_{4,i,2})$	Medelvärde MSE_i	σ_i
1	$(\mathbf{A}_{ne}, \mathbf{x}, \gamma_E, \hat{\mathbf{s}}, \mathbf{A}_{ti_0}), (\mathbf{A}_{ti_0}, \hat{\mathbf{s}}, \tau)$	$2.3955 \cdot 10^{-3}$	$3.2 \cdot 10^{-6}$
2	$(A_{ne}, x, \gamma_E, \hat{s}, A_{ne}), (\hat{s}, A_{ti_0}, \tau)$	$2.4276 \cdot 10^{-3}$	$7.4 \cdot 10^{-6}$
3	$(A_{ne}, x, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, A_{ti_0}, \tau)$	$2.6898 \cdot 10^{-3}$	$1.9 \cdot 10^{-5}$

Den struktur som åstadkom lägst MSE_i var den där $G_{2,1,1}$ behöll sina inputparametrar, medan A_{ne} plockades bort från $G_{2,1,2}$. För alla andra strukturer gällde att $|MSE_i - MSE_{\text{bäst}}| > 2\sigma_i$.

Det gjordes ett försök att reducera antalet inputparametrar även i $G_{4,1,1}$ men valideringsfelet översteg då riktmärkesvärdet. Det ansågs därför inte vara möjligt att minska antalet parametrar i grenarna mer, utan den struktur som sedan delades upp i mindre delar innehöll $G_{4,1,1}$ och $G_{4,1,2}$.

3.5.2.4 Förgrening av de grovt indelade grenarna

Det slutgiltiga målet var att grenmodellen skulle ha en struktur där inputparametrarna på ett optimalt sätt delats upp, så att de mellanliggande neurala nätverken tog emot två inputparametrar var. De inputparametrar som nätverken tog emot kunde antingen vara de ursprungliga inputparametrarna från QuaLiKiz, eller endimensionella outputparametrar bestående av två komprimerade inputparametrar. Denna struktur eftertraktades eftersom den skulle göra det möjligt att visualisera input- och outputparametrar likt i figur 2.9.

Det tillvägagångssätt som valdes var att en gren i taget behandlades; till en början lämnades $G_{4,1,2}$ orörd. Steg ett blev därmed att dela upp $G_{4,1,2}$ i två, mindre grenar genom att låta en inputparameter bilda en egen gren. Denna metod itererades sedan till dess att $G_{4,1,2}$ delats upp i fyra grenar, tre av med en ensam inputparameter, och en med ett par. Därefter gjordes en uppdelning av $G_{4,1,1}$. Eftersom denna gren endast bestod av tre inputparametrar var endast en iteration nödvändig. I tabellerna nedan redovisas relevanta resultat från samtliga iterationer, som sedan gav upphov till den slutgiltiga grenmodellen. Fullständiga resultat finns i appendix B.

Tabell 3.6: Medelkvadratfel MSE_i för den bästa och den sämsta strukturen där en parameter brutits loss från grenen med fem inputparametrar, samt σ_i för dessa. Den bästa kombinationen är markerad med fetstil. Resultat för resterande konfigurationer visas i tabell B.5.

i	$((G_{5,i,11}), (G_{5,i,12})), (G_{5,i,2})$	Medelvärde MSE_i	σ_i
1	$((\mathbf{A}_{ne}, \mathbf{x}, \gamma_E, \hat{\mathbf{s}}), (\mathbf{A}_{ti_0})), (\mathbf{A}_{ti_0}, \hat{\mathbf{s}}, \tau)$	$2.4374 \cdot 10^{-3}$	$8.4 \cdot 10^{-6}$
...
5	$(A_{ne}, x, \hat{s}, A_{ti_0}), (\gamma_E), (\hat{s}, A_{ti_0}, \tau)$	$2.5750 \cdot 10^{-3}$	$4.2 \cdot 10^{-5}$

MSE_i blev lägst när A_{ti_0} bildade en egen gren. För de övriga konfigurationerna gällde att $|MSE_i - MSE_{\text{bäst}}| > 2\sigma_i$.

Tabell 3.7: Medelkvadratfel MSE_i för strukturerna där två parametrar brutits loss från grenen med fem inputparametrar, samt standardavvikelser σ_i för dessa.

i	$((G_{6,i,11}), (G_{6,i,12}), (G_{6,i,13})), (G_{6,i,2})$	Medelvärde MSE_i	σ_i
1	$((\mathbf{A}_{ne}, \gamma_E, \hat{s}), (\mathbf{x}), (\mathbf{A}_{ti_0}), (\mathbf{A}_{ti_0}, \hat{s}, \tau)$	$2.4283 \cdot 10^{-3}$	$6.1 \cdot 10^{-6}$
2	$((x, \gamma_E, \hat{s}), (A_{ne}), (A_{ti_0}), (\hat{s}, A_{ti_0}, \tau)$	$2.4290 \cdot 10^{-3}$	$4.3 \cdot 10^{-6}$
3	$((A_{ne}, x, \hat{s}), (\gamma_E), (A_{ti_0}), (\hat{s}, A_{ti_0}, \tau)$	$2.4394 \cdot 10^{-3}$	$6.1 \cdot 10^{-6}$
4	$((A_{ne}, \gamma_E, x), (\hat{s}), (A_{ti_0}), (\hat{s}, A_{ti_0}, \tau)$	$2.4814 \cdot 10^{-3}$	$5.4 \cdot 10^{-6}$

När ytterligare en gren bildades från den ursprungliga, stora grenen, var modellen med lägst MSE_i den då x brutits loss. Dock var skillnaden mellan $MSE_{\text{bäst}}$ och MSE_i liten ($< \sigma_i$) för två av de tre resterande modellerna, och därför togs beslutet att inte utesluta någon struktur inför nästa iteration.

Tabell 3.8: Medelkvadratfel MSE_i för strukturerna där tre parametrar brutits loss från grenen med fem inputparametrar, samt standardavvikelser σ_i för dessa. Resultat för resterande konfigurationer redovisas i tabell B.6.

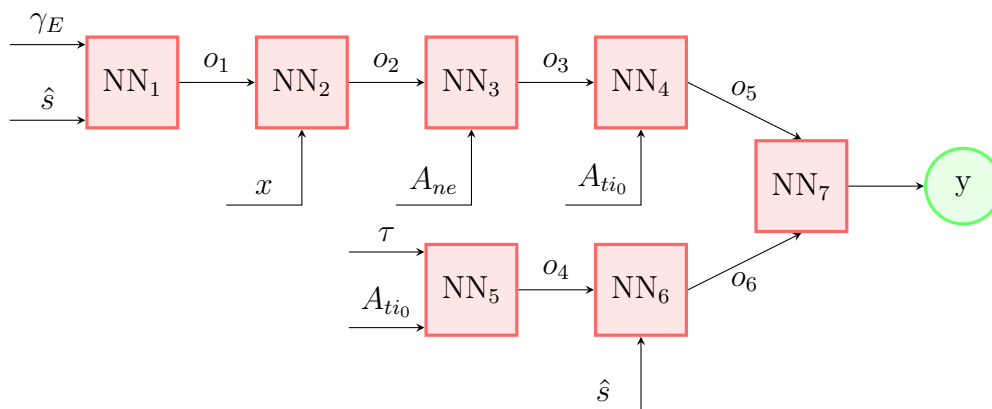
i	$((G_{7,i,11}), \dots, (G_{7,i,13})), (G_{7,i,2})$	Medelvärde MSE_i	σ_i
1	$((\mathbf{A}_{ne}, \hat{s}), (\gamma_E), (\mathbf{x}), (\mathbf{A}_{ti_0}), (\mathbf{A}_{ti_0}, \hat{s}, \tau)$	$2.4881 \cdot 10^{-3}$	$4.0 \cdot 10^{-6}$
2	$((\gamma_E, \hat{s}), (x), (A_{ne}), (A_{ti_0}), (\hat{s}, A_{ti_0}, \tau)$	$2.4937 \cdot 10^{-3}$	$4.9 \cdot 10^{-6}$
...
6	$((A_{ne}, x), (\hat{s}), (\gamma_E), (A_{ti_0}), (\hat{s}, A_{ti_0}, \tau)$	$2.5830 \cdot 10^{-3}$	$1.2 \cdot 10^{-6}$

I och med den slutliga uppdelningen av den stora grenen visade det sig att lägst MSE_i genererades då A_{ne} och \hat{s} fanns i samma gren. Denna konfiguration var tätt följd av konfigurationen med γ_E och \hat{s} i samma gren, och därför användes båda dessa strukturer för den sista förgreningen.

Tabell 3.9: Medelkvadratfel MSE_i för den bästa och sämsta modellen där en parameter brutits loss från grenen med tre inputparametrar, samt standardavvikelser σ_i för dessa. Resultat för resterande konfigurationer visas i tabell B.7.

i	$((G_{8,i,11}), \dots, (G_{8,i,14})), ((G_{8,i,21}), (G_{8,i,22}))$	Medelvärde MSE_i	σ_i
1	$((\gamma_E, \hat{s}), (\mathbf{x}), (\mathbf{A}_{ne}), (\mathbf{A}_{ti_0}), ((\mathbf{A}_{ti_0}, \tau), (\hat{s}))$	$2.4197 \cdot 10^{-3}$	$3.7 \cdot 10^{-6}$
...
6	$((\gamma_E, \hat{s}), (x), (A_{ne}), (A_{ti_0}), (\hat{s}, A_{ti_0}), (\tau)$	$2.4610 \cdot 10^{-3}$	$3.6 \cdot 10^{-6}$

Det visade sig att den grenmodell som genererade lägst MSE_i efter den sista förgreningen hade γ_E och \hat{s} i $G_{8,1,11}$, och A_{ti_0} och τ i $G_{8,1,12}$. För resterande modeller gällde att $|MSE_i - MSE_{\text{bäst}}| > 2\sigma_i$. Det noterades även att alla grenmodeller med denna uppdelning hade lägre MSE_i än den bästa modellen i tabell 3.8. I kommande delar av rapporten kommer denna modell helt enkelt benämnas "grenmodellen". De ursprungliga, grövre grenarna kommer att benämnas G_1 och G_2 . Grenmodellen visualiseras i figur 3.10.



Figur 3.10: Den slutgiltiga grenmodellens struktur. NN₁, NN₂, NN₃ och NN₄ representerar G_1 , medan NN₄ och NN₅ representerar G_2 .

4

Resultat och diskussion

I detta avsnitt redovisas och diskuteras resultaten och som påträffats under projektet.

4.1 Sammanfattning av delresultat

Under sektion 3.3 beskrivs det hur autokodare har använts för att se hur många dimensioner det gick att reducera datan till. Figur 3.3 visar att med en tvådimensionell latent representation av tillväxthastigheterna blev felet cirka $1.159 \cdot 10^{-3}$, vilket bedömdes vara lågt nog för att ge en god representation av datan med tillräcklig noggrannhet. Vidare redogör sektion 3.4 vilka inputparametrar som är mest betydelsefulla för ett neuralt nätverks förmåga att prediktera tillväxthastigheterna, samt hur många av de mest betydelsefulla inputparametrarna behövs för att prediktera tillväxthastigheterna med tillräcklig precision. Detta illustreras i figur 3.5 där sex inputparametrar anses ge ett tillräckligt lågt värde på felet. De sex inputparametrarna med störst påverkan på felet är: x , A_{ne} , τ , γ_E , \hat{s} , A_{ti_0} .

4.2 Diskussion av delresultat

4.2.1 Autokodare

Figur 3.3 visar att tillväxthastigheten hos instabiliteter i jonskala kan beskrivas väl av endast två variabler. I minimum fallet så minskar felet med ca 80 % om två noder används istället för en, medan felet endast minskar med ca 30 % när tre noder används istället för två. Däremot visar figur 3.3 att genomsnittet är märkbart sämre vilket förmodligen handlar om att många modeller har fastnat i lokala minimum. Då målet med autokodare var att ta reda på hur mycket datan kunde komprimeras utan att ha ett för stort fel så är med mer relevant att kolla på det lägsta felet istället för genomsnitt. Det finns även inget som tyder på att grenmodellen skulle uppvisa samma problem med att fastna i lokal minimum.

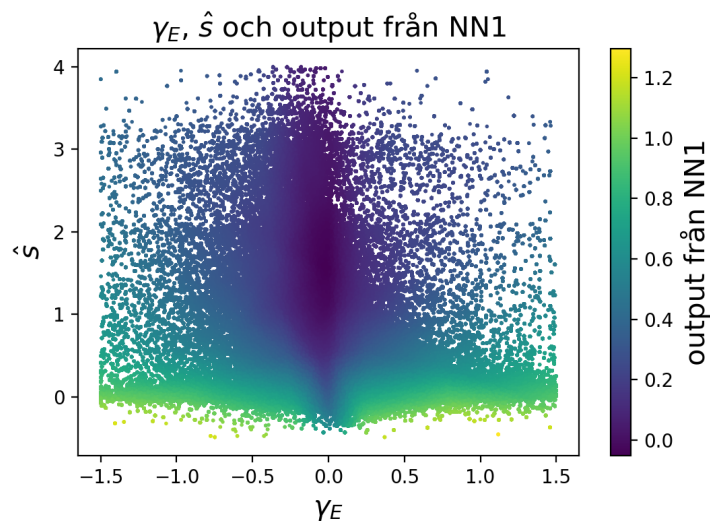
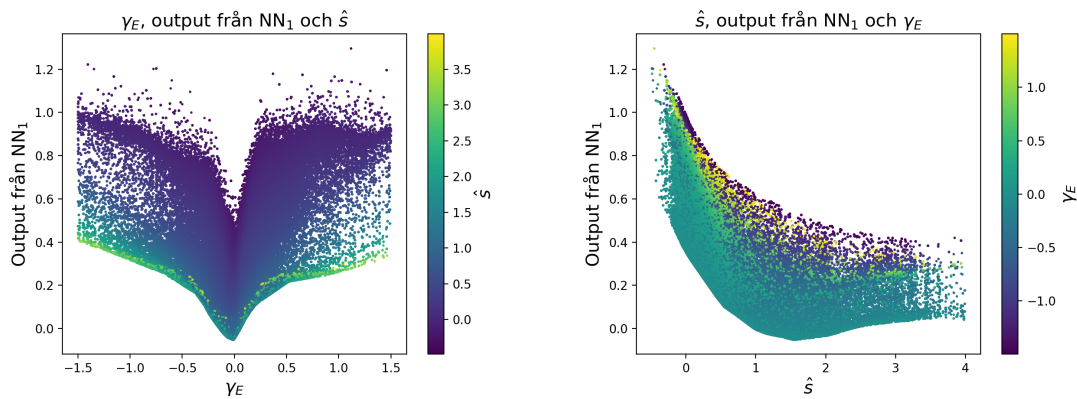
4.2.2 Reduktion av parameterrymd

För att utförligt kunna hitta samband mellan de olika inputparametrarna med grenmodellen behöver många olika kombinationer av inputparametrar testas att paras ihop på olika sett. Detta gör att fler inputparametrar leder till drastiskt många

fler möjliga kombinationer vilket är anledningen till att de mest essentiella inputparametrarna togs fram först. I figur 3.5 går det att se att MSE var ca 0.0017 för ett nätverk med samtliga inputparametrar. Utifrån detta och hur felet utvecklades genom att succesivt ta bort inputparametrar så argumenterades det för att det åtminstone behövdes sex stycken inputparametrar för att beskriva datan tillräckligt bra då felet låg på cirka 0.0023, vilket är en ökning på 35 %. Då syftet inte var att skapa en surrogatmodell som kan prediktera tillväxhastigheterna så bra som möjligt utan att göra en tolkningsbar modell så ansågs 35 % som en godtarbar ökning av felet. Även projektets begränsade tidsram användes som motivation av valet att endast använda sex inputparametrar.

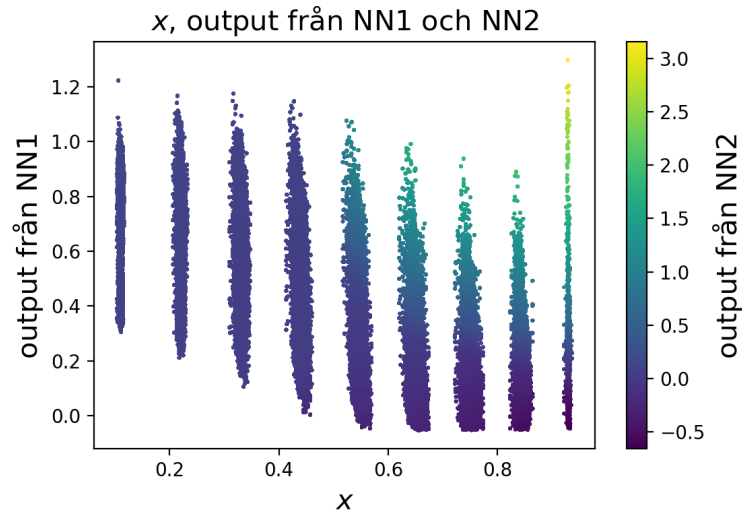
4.3 Resultat och diskussion av grenmodellen

Här visas resultat och diskussion av samtliga plottar som visar hur inputdata beräknar outputdata för alla neurala nätverk i Figur 3.10

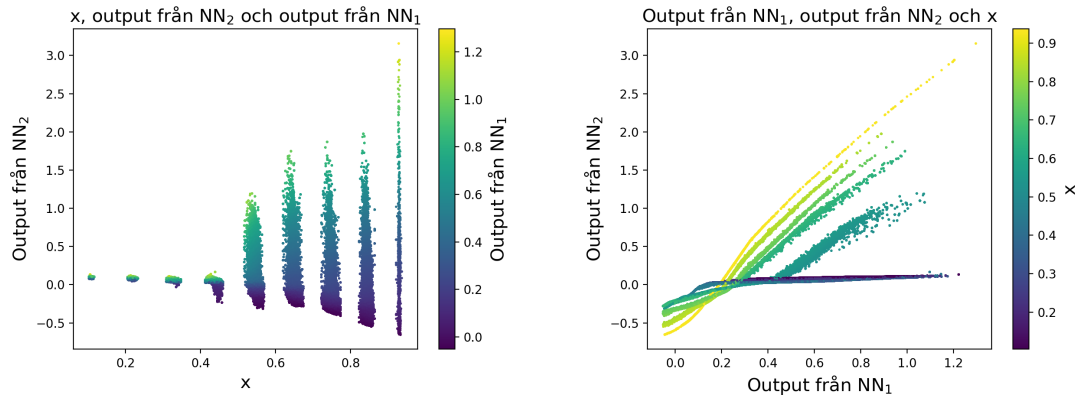
(a) γ_E , \hat{s} och output från NN₁.(b) γ_E , output från NN₁ och \hat{s} .(c) \hat{s} , output från NN₁ och γ_E .

Figur 4.1: Figurerna visar samband mellan inputparametrarna γ_E , \hat{s} och output från NN₁ i figur 3.10.

Figur 4.1 visar att både γ_E och \hat{s} signifikant påverkar output från NN₁, men deras effekter varierar beroende på deras värden. När \hat{s} är nära noll är output från NN₁ betydligt högre. För högre värden på \hat{s} minskar dock NN₁s outputvärden. γ_E har en mycket signifikant inverkan på output från NN₁, särskilt när γ_E är nära noll, vilket också leder till låga outputvärden från NN₁. γ_E verkar få en svagare effekt på output från NN₁ desto längre ifrån noll γ_E är. Det är även intressant att notera hur γ_E har symmetrisk påverkan på outputen från NN₁. Vid $\gamma_E = 0$ blir det två separata regimer på vardera sida av noll, vilket syns i Figur 4.1b. En regim i detta fall avser en distinkt uppdelning som uppstår från olika parametervärden.



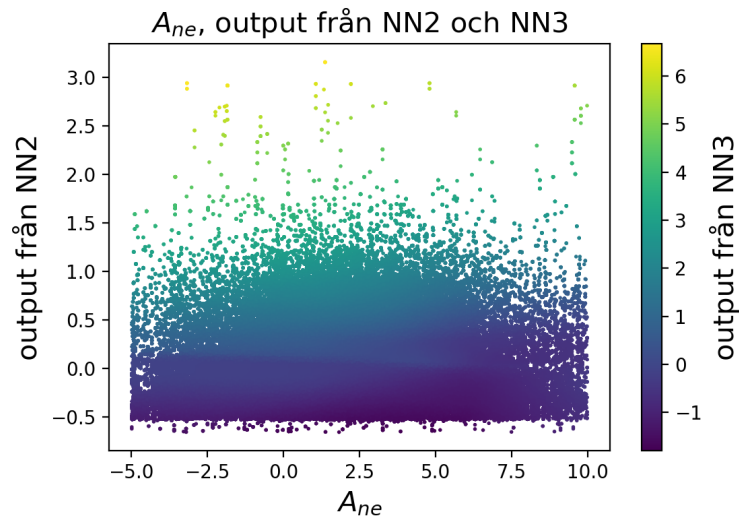
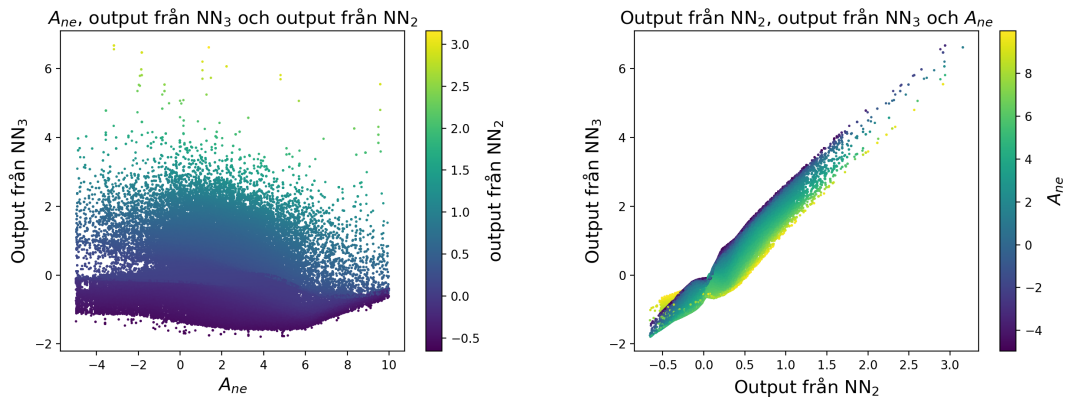
(a) x , output från NN_1 och output från NN_2



(b) x , output från NN_2 och output från NN_1 . (c) Output från NN_1 , output från NN_2 och x .

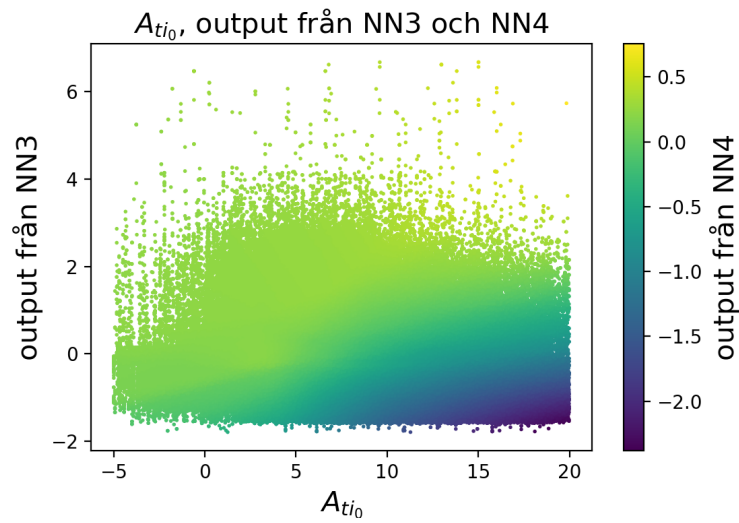
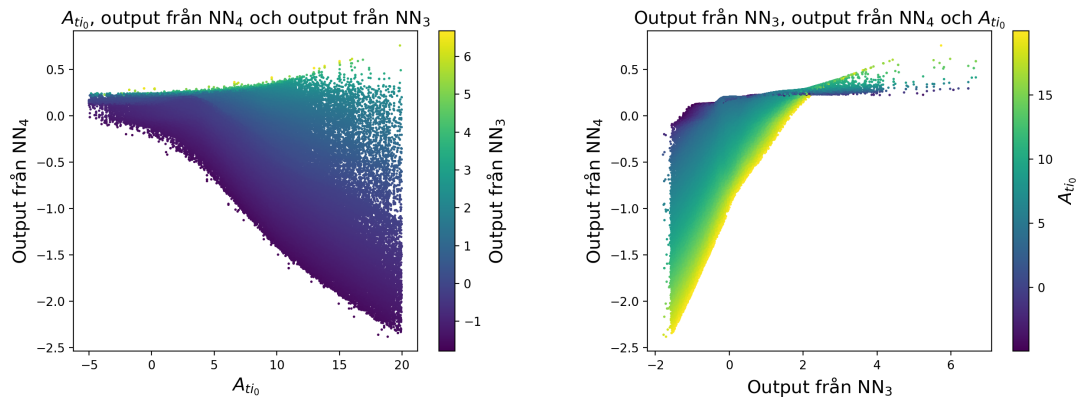
Figur 4.2: Figurerna visar samband mellan x , output från NN_1 och output från NN_2 .

Vidare observeras i figur 4.2 ett beteendeskifte för x -värden runt 0.4. Det går att göra en direkt fysikalisk koppling för detta samband då x representerar den radiella positionen i en tokamak. Plasmats instabiliteter varierar betydligt när de är nära tokamakens kärna, jämfört med när det är nära dess yttre gränser. Denna skillnad i beteende illustreras tydligt i Figur 4.2b och 4.2c. För $x < 0.4$ är outputvärdena från NN_2 noll eller nära noll, medan output från NN_2 ökar då $x > 0.4$. I med att output från NN_1 är nästintill irrelevant för $x < 0.4$ så innebär det att γ_E är nästintill irrelevant för $x < 0.4$. Samma argument går inte att göra för \hat{s} , då \hat{s} förekommer även i G_2 .

(a) A_{ne} , output från NN_2 och output från NN_3 .(b) A_{ne} , output från NN_3 och output från NN_2 .(c) A_{ne} , output från NN_2 och output från NN_3

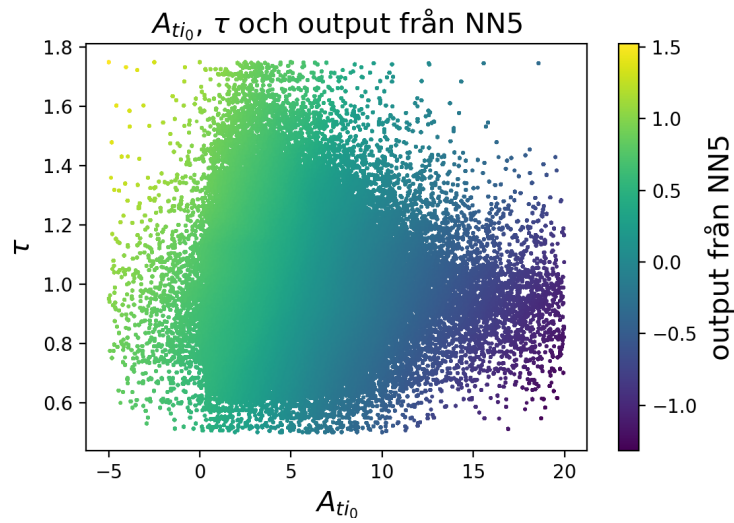
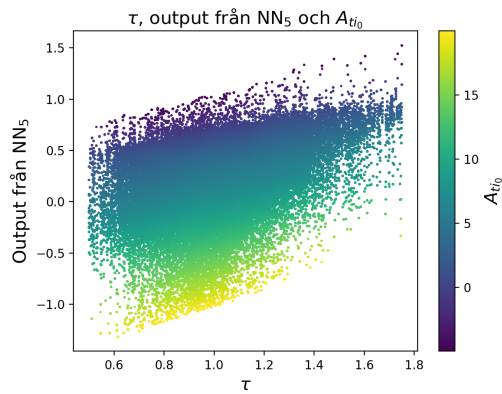
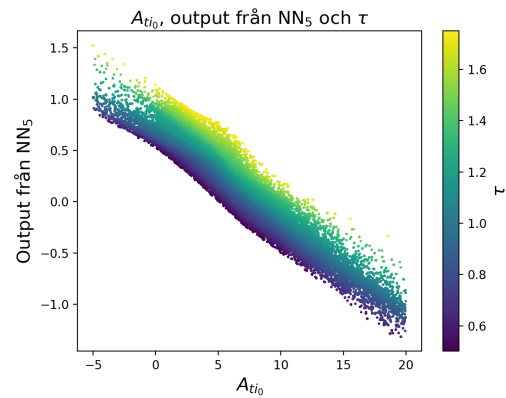
Figur 4.3: Figurerna visar samband mellan A_{ne} , output från NN_2 och output från NN_3 .

Figur 4.3 visar att A_{ne} har en låg påverkan på output från NN_3 , en slutsats som kan dras utifrån att outputns värde nästan är oberoende av A_{ne} -värdet. Detta indikerar att grenmodellen troligtvis skulle kunna förenklas vidare genom att inte ha med A_{ne} , eftersom påverkan av A_{ne} till stor del verkar kunna återskapas av andra inputparametrar.

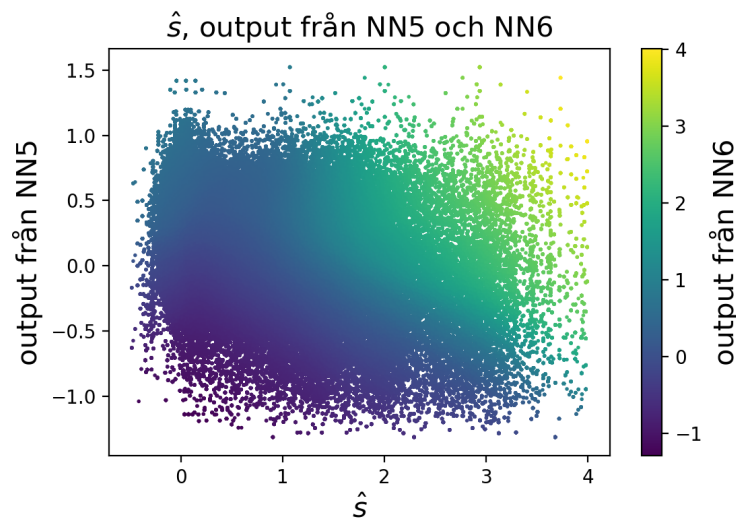
(a) A_{ti_0} , output från NN₃ och output från NN₄(b) A_{ti_0} , output från NN₄ och output från NN₃.(c) Output från NN₃, output från NN₄ och A_{ti_0} .

Figur 4.4: Figurerna visar samband mellan A_{ti_0} , output från NN₃ och output från NN₄.

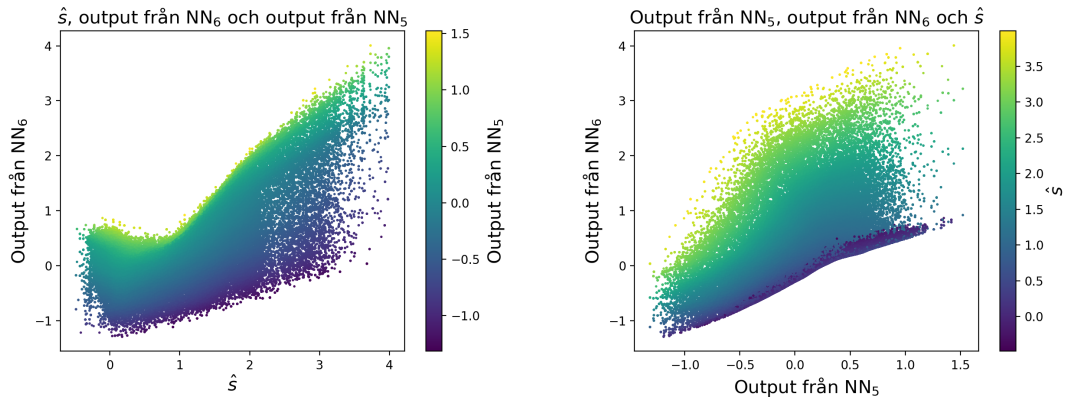
I figur 4.4a visas ett intressant samband där outputen från NN₃ som härstammar från tre olika input-parametrar ställs mot A_{ti_0} . Inputparametern A_{ti_0} är jontemperaturgradienten och är den viktigaste inputparametern för att prediktera tillväxthastigheter för jonskalor. I figuren syns det att A_{ti_0} får större påverkan på outputen från NN₄ då A_{ti_0} växer, vilket resulterar i lägre värden på outputen från NN₄. För de flesta värden verkar dock output från NN₃ ha en mer dominant effekt på vad output från NN₄ blir.

(a) A_{ti_0} , τ och output från NN₅(b) τ , output från NN₅ och A_{ti_0} (c) A_{ti_0} , output från NN₅ och τ **Figur 4.5:** Figureerna visar samband mellan A_{ti_0} , τ och output från NN₅.

Vidare i analysen av grenmodellen visas det i figur 4.5 att A_{ti_0} har betydligt större påverkan på output från NN₅ jämfört med τ , vilket är väntat då A_{ti_0} var den viktigaste inputen. Figuren visar hur output från NN₅ minskar när A_{ti_0} ökar.



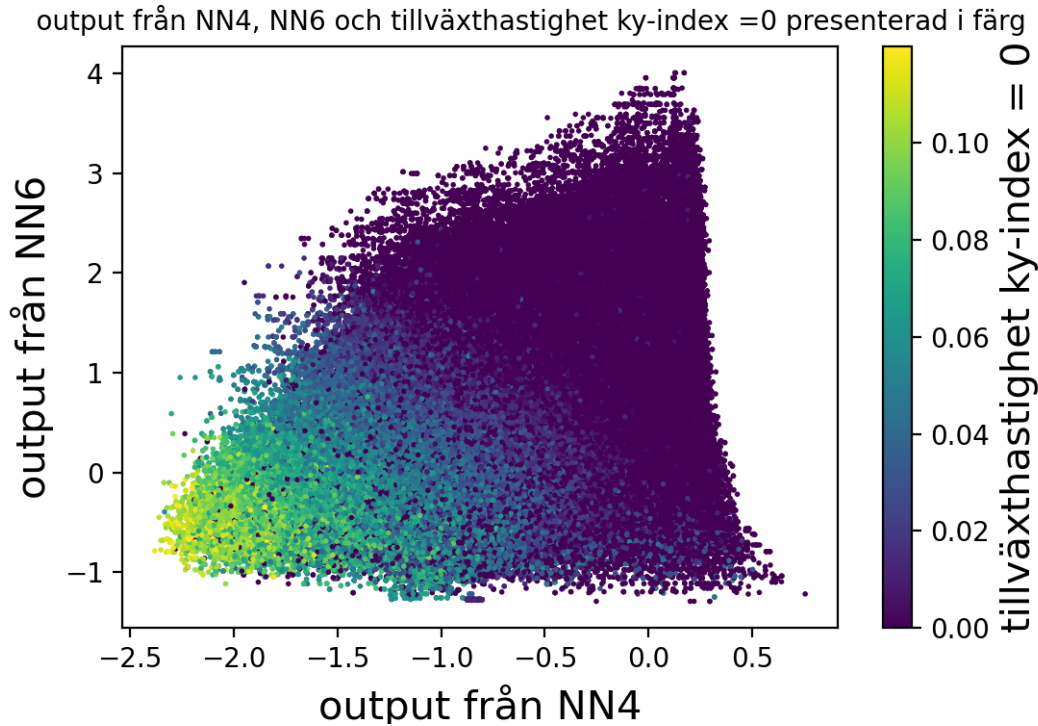
(a) \hat{s} , output från NN_5 och output från NN_6



(b) \hat{s} , output från NN_6 och output från NN_5 . (c) Output från NN_5 , output från NN_6 och \hat{s}

Figur 4.6: Figurerna visar samband mellan $A_{t_{i_0}}$, output från NN_3 och output från NN_4 .

Figur 4.6 visar att både \hat{s} och output från NN_5 har stor påverkan på outputen från NN_6 . Output från NN_6 verkar beräknas med tendenser av additionsliknande beräkningar av \hat{s} och output från NN_5 . Det innebär att högre värden på \hat{s} och output från NN_5 leder till högre värden på output från NN_6 , medan det motsatta gäller för lägre värden på \hat{s} och output från NN_5 .



Figur 4.7: 4.5 och 4.6 visar plottar för sista övre grenen och sista nedre grenen i figur 3.10, medan 4.7 visar den slutliga plotten där tillväxthastigheter predikteras från de sista grenarna. I bilden är k_y -index = 0. För resterande k_y -index se figur A.3.

Slutligen i grenmodellen så plottades det hur tillväxthastigheter för olika k_y -index förhåller sig till output från NN_4 och output från NN_6 i Figur 4.7. Där visas det vid vilka värden på output från NN_4 och NN_6 som leder till stabila respektive instabila tillväxthastigheter. En tillväxthastighet är stabil om värdet är noll, och instabil om det är över noll. I figur A.3 illustreras det hur variationer i tillväxthastigheter från NN_4 och NN_6 förhåller sig till olika k_y -index. Detta korrelerar med Figur 2.3, där tillväxthastigheterna ökar med ökande k_y -index upp till 4 och därefter minskar. I Figur 4.7 observeras att lägre outputvärden från NN_4 och NN_6 leder till ökad instabilitet, medan högre outputvärden från NN_4 och NN_6 resulterar i stabila tillväxthastigheter. För att öka förståelsen om vilka värden på inputparametrar som leder till instabila tillväxthastigheter så går det att gå baklänges i grenmodellen.

Genom en bakåtblickande analys i grenmodellen längs G_1 , alltså den övre grenen i Figur 3.10, blir det möjligt att analysera hur låga värden på output från NN_4 härstammar från höga värden på A_{vi_o} och låga värden på output från NN_3 . Det som avgör output från NN_3 är främst output från NN_2 , då A_{ne} inte har en särskilt signifikant påverkan. Vid låga värden på output från NN_3 gäller det att output från NN_2 också är lågt. Output från NN_2 härstammar från x och output från NN_1 , där låga värden på output från NN_2 uppnås när $x < 0.4$ eller output från NN_1 är nära noll. Output från NN_1 härstammar från \hat{s} och γ_E , där värden nära noll på output från NN_1 främst uppnås när γ_E är nära noll och \hat{s} är över cirka 0.5, vilket syns tydligast

i Figur 4.1b och 4.1c. Utifrån detta så är approximativt de värden på inputparametrar i G_1 som resulterar i högst tillväxthastigheter för instabiliteter då A_{ti_0} är högt, $x < 0.4$, när \hat{s} är litet men inte under cirka 0.5 och γ_E är nära noll.

En bakåtblickande analys i grenmodellen längs G_2 , alltså den lägre grenen i figur 3.10 så går det att observera att låga outputvärden från NN_6 uppstår då output från NN_5 och \hat{s} båda har låga värden. Output från NN_5 härstammar från τ och A_{ti_0} , där låga värden på output från NN_5 i stort sett enbart kommer från höga värden på A_{ti_0} , då τ inte har en nämnvärd påverkan. Utifrån detta så är approximativt de värden på inputparametrar i G_2 som leder till högst tillväxthastigheter för instabiliteter då A_{ti_0} är högt och \hat{s} är lågt.

\hat{s} och A_{ti_0} förekommer i både G_1 och G_2 . Höga värden på A_{ti_0} visar i både G_1 och G_2 ökande tillväxthastigheter för instabiliteter. Beträffande \hat{s} varierar dess effekt i de olika grenarna. I G_1 så innebär \hat{s} -värden under 0.5 lägre tillväxthastighet för instabiliteter, men för övriga \hat{s} är det relativt konstant. I G_2 å andra sidan så innebär lägre värden på \hat{s} högre tillväxthastighet för instabiliteter. Sammantaget är approximativt de värden på inputparametrar som leder högst tillväxthastigheter för instabiliteter då A_{ti_0} är högt; lågt värde på \hat{s} , men inte under cirka 0.5; γ_E noll eller nära noll och $x < 0.5$. A_{ne} och τ har i grenmodellen en låg påverkan på instabiliteters tillväxthastigheter och är därmed inte med i bedömningen av vilka inputparameter-värden som påverkar instabiliteters tillväxthastigheter.

4.4 Begränsningar i arbetet

En betydande begränsning under projektet var den beräkningskapacitet och den tid som fanns tillgänglig. Eftersom träning av olika neurala nätverksmodeller utfördes på egna datorer, varierade tiden det tog att genomföra beräkningarna, och i vissa fall var den exceptionellt lång. Eftersom körningarna ofta var tidskrävande, var det inte möjligt att genomföra dem tillräckligt många gånger för att uppnå statistiskt robusta resultat. På grund av detta finns viss osäkerhet om den bästa strukturen för grenmodellen faktiskt hittades. En skattning av σ för de tre bästa av tio modeller användes för att rättfärdiga vilka modeller som testades - och denna metod garanterar inte ett korrekt resultat. Om det hade funnits mer tid hade modellerna körts fler gånger, vilket hade genererat säkrare väntevärden och standardavvikelser. Det hade då varit möjligt att utföra hypotestest för att finna sannolikheten för att en viss modell faktiskt var bäst.

4.5 Framtida arbete

Syftet med detta arbete beskrevs under sektion 1.2 som att skapa en surrogatmodell av QuaLiKiz där det skulle gå att kunna hitta samband mellan input- och outputparametrarna. En vidareutveckling av detta syfte skulle vara att kunna analysera sambanden mellan inputparametrarna och tillväxthastigheterna ur ett mer omfattande fysikaliskt perspektiv. Detta med hopp om att en ökad förståelse av

inputparametrarna i QuaLiKiz kan leda till att det går att minska instabiliteterna inom plasmata i en tokamak.

För att komma dit skulle framtida utvecklingsvägar för grenmodellen kunna innebära att antingen implementera den modell som presenteras i detta kandidatarbete eller utforma en mer avancerad version som inkluderar fler inputparametrar och uppnår högre precision. Att använda grenmodeller kan inte bara förbättra precisionen hos andra surrogatmodeller för QuaLiKiz, utan också stärka tilliten till användningen av dessa modeller.

5

Slutsatser

Tillväxthastigheter för jonskalorna kunde effektivt komprimeras från nio dimensioner till två dimensioner utan att en signifikant mängd information gick förlorad, vilket observerades i de mest framgångsrika körningarna av autokodaren.

De inputparametrar som är mest väsentliga för att prediktera tillväxthastigheter för jonskalorna är följande sex (från mest viktig till minst viktig): A_{ti_0} - Normaliserad jontemperaturgradient, \hat{s} - Magnetisk skjuvning, γ_E - ExB skjuvningshastighet, τ - Jon-elektrontemperatur-relation, A_{ne} - Normaliserad elektrondensitetgradient och x - Normaliserad radiell position. Det visades även att det neurala nätverket kunde prediktera tillväxthastigheterna bättre än den linjära modellen, oavsett antal parametrar som användes.

Några fysikaliska samband som tydligt kunde hittas i grenmodellen var att γ_E , som motsvarar ExB skjuvningshastighet har symmetrisk påverkan. För x , alltså normaliserad radiell position så finns det olika fysikaliska beteenden beroende på om x är större eller mindre än 0.4, där $x > 0.4$ leder till högre output-värden medan $x < 0.4$ nästan enbart gav outputvärden som var nära noll. Detta är en representation av att plasmat har olika beteende nära kärnan av tokamaken och vid yttre kanten av tokamaken.

Uppskattningsvis genom analys av graferna från grenmodellen är de värden på inputparametrarna som leder till mest instabilitet då A_{ti_0} är högt, \hat{s} är lågt men inte under cirka 0.5, γ_E är noll eller nära noll och $x < 0.4$.

6

Sammanfattning

I detta arbete har en surrogatmodell av Qualikiz skapats, benämnd grenmodellen. Grenmodellens syfte var att visa hur inputparametrar från Qualikiz leder till tillväxthastigheter av instabiliteter genom en sammanhängande struktur av neurala nätverk. Varje separat neuralt nätverk inom grenmodellen tar in två inputs och producerar en output, vilket möjliggör visualisering av dessa relationer i form av plottar. I grenmodellen komprimerades inputs i varje steg av de neurala nätverken fram till det sista steget, där tillväxthastigheter predikterades. För att få en indikation på hur mycket grenmodellen kunde komprimeras samtidigt som den effektivt predikterar tillväxthastigheter, tillämpades autokodare på outputen, det vill säga tillväxthastigheterna. Autokodaren visade att outputen kunde reduceras från nio till två dimensioner utan signifikant informationsförlust. Eftersom inputparametrarna effektivt kunde förutsäga tillväxthastigheterna med neurala nätverk, innebar detta att tillväxthastigheterna kunde predikteras från komprimerade inputs i två dimensioner i grenmodellen. Att använda samtliga 15 inputs från Qualikiz i grenmodellen skulle innebära för mycket körningstid och komplexitet för tolkbarheten av sambanden hos grenmodellen. Därav bestämdes det vilka inputs som var mest betydelsefulla för att prediktera tillväxthastigheterna för jonskalorna, vilket både genererade värdefull insikt och angav vilka sex inputs som skulle användas i grenmodellen.

När grenmodellen var utförd möjliggjorde den spårbarhet från input till output, vilket är av stort värde för att hantera det så kallade svarta låda-problemet som är associerat med neurala nätverk. Ökad tolkbarhet av surrogatmodeller baserade på neurala nätverk av Qualikiz-modellen förbättrar förståelsen och gör det därmed både etiskt och ekonomiskt mer försvarbart att använda modellen i praktiska tillämpningar.

Framtida utvecklingsvägar för grenmodellen kan innebära att forskare antingen implementerar den modell som har utvecklats i detta kandidatarbete, eller utformar en annan version av grenmodellen som omfattar fler inputparametrar och uppnår högre precision. Vidare skulle användningen av grenmodeller inte bara kunna öka precisionen i andra surrogatmodeller av Qualikiz, utan även stärka tilliten till användningen av dessa modeller. Slutligen skulle detta kunna leda till att förståelsen ökar om hur det går att minska instabiliteter i plasma för magnetisk innesluten fusion och därmed lämna ett positivt avtryck i utvecklingen av fusion på jorden.

Referenser

- [1] Oak Ridge National Laboratory, "ITER 3D plasma equilibrium with ripple contours, Flickr," nov. 2009. [Online]. Tillgänglig: <https://www.flickr.com/photos/37940997@N05/4071616631>.
- [2] M. B. Irena Chatzis, "What is Fusion, and Why Is It So Difficult to Achieve?" *IAEA Bulletin*, vol. 62-2, ss. 4–5, maj 2021. [Online]. Tillgänglig: <https://www.iaea.org/bulletin/what-is-fusion-and-why-is-it-so-difficult-to-achieve>.
- [3] M. Greshko. "Scientists achieve a breakthrough in nuclear fusion. Here's what it means." (2022), [Online]. Tillgänglig: <https://www.nationalgeographic.com/science/article/scientists-achieve-breakthrough-nuclear-fusion> (Hämtad: 28 april 2024).
- [4] M. Kikuchi, *Frontiers in Fusion Research*. Naka, Ibaraki, Japan: Springer, 2011. [Online]. Tillgänglig: <https://doi.org/10.1007/978-1-84996-411-1>.
- [5] E. Fransson, "Turbulent transport in tokamak plasmas: linear-, quasi- and non-linear simulations," doktorsavhandling, Chalmers University of Technology, 2023. [Online]. Tillgänglig: https://research.chalmers.se/publication/535405/file/535405_Fulltext.pdf.
- [6] C. Llewellyn-Smith och D. Ward, "Fusion power," *European Review*, vol. 13, nr 3, ss. 337–359, 2005. DOI: 10.1017/S1062798705000499.
- [7] Iter Organization. "External Heating Systems." (2024), [Online]. Tillgänglig: <https://www.iter.org/mach/Heating> (Hämtad: 6 maj 2024).
- [8] A. Vlasov. "Tokamaks, Stellarators, Laser-based and Alternative Concepts: Report Offers Global Perspective on Nuclear Fusion Devices." (2023), [Online]. Tillgänglig: <https://www.iaea.org/newscenter/news/tokamaks-stellarators-laser-based-and-alternative-concepts-report-offers-global-perspective-on-nuclear-fusion-devices> (Hämtad: 10 april 2024).
- [9] T. H. Moulden, "An Introduction to Turbulence Phenomena," i *Handbook of Turbulence: Volume 1 Fundamentals and Applications*, W. Frost och T. H. Moulden, red. Boston, MA: Springer US, 1977, ss. 23–51. DOI: 10.1007/978-1-4684-2322-8_2.
- [10] C. Stephens, X. Garbet, J. Citrin, C. Bourdelle, K. van de Plassche och F. Jenko, "Quasilinear gyrokinetic theory: a derivation of QuaLiKiz," *Journal of Plasma Physics*, vol. 87, 2021, 3rd International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy. DOI: <https://doi.org/10.1017/S0022377821000763>.
- [11] S.-M. Udrescu och M. Tegmark, "AI Feynman: A physics-inspired method for symbolic regression," *Science Advances*, vol. 6, nr 16, eaay2631, 2020. DOI: 10.1126/sciadv.aay2631.

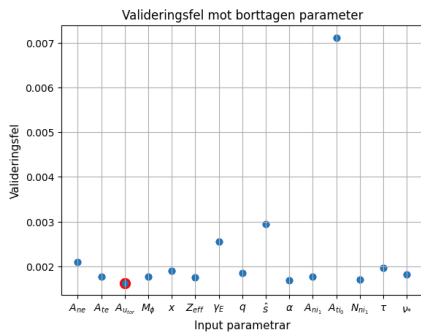
-
- [12] JET Team, "Fusion energy production from a deuterium-tritium plasma in the JET tokamak," *Nuclear Fusion*, vol. 32, nr 2, ss. 187–203, 1992. DOI: 10.1088/0029-5515/32/2/I01.
- [13] F. Fernflores, *Einstein's Mass-Energy Equation, Volume I*. New York, USA: Monumentum Press, 2017.
- [14] A. Gillgren, "Machine learning applications for predicting the pedestal in tokamak plasmas," licentiatavhandling, Chalmers University of Technology, 2023. [Online]. Tillgänglig: https://research.chalmers.se/publication/539794/file/539794_Fulltext.pdf.
- [15] The Editors of Encyclopaedia. "Nuclear fission and nuclear fusion: what you should know." (2022), [Online]. Tillgänglig: <https://www.orano.group/en/unpacking-nuclear/nuclear-fission-and-nuclear-fusion-what-you-should-know> (Hämtad: 21 april 2024).
- [16] B. Nouhi, N. Darabi, P. Sareh, H. Bayazidi, F. Darabi och S. Talatahari, "The fusion–fission optimization (FuFiO) algorithm," *Scientific Reports*, vol. 12, nr 12396, 2022. DOI: 10.1038/s41598-022-16498-4.
- [17] E. Fransson, "Simulation and assessment of particle transport in fusion plasmas," Licentiatuppsats, Institution for Physics, 2021. [Online]. Tillgänglig: https://research.chalmers.se/publication/535405/file/535405_Fulltext.pdf.
- [18] Z. Wang, "Current status of research on magnetic confinement fusion and superconducting tokamak devices," *Procedia Computer Science*, vol. 228, ss. 163–170, 2023, 3rd International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy. DOI: <https://doi.org/10.1016/j.procs.2023.11.020>.
- [19] J. Sánchez, "Nuclear fusion as a massive, clean, and inexhaustible energy source for the second half of the century: brief history, status, and perspective," *Energy Science & Engineering*, vol. 3, nr 6, ss. 165–176, okt. 2014. DOI: <https://doi.org/10.1002/ese3.43>.
- [20] Iter Organization. "What Is A Tokamak?" (2024), [Online]. Tillgänglig: <https://www.iter.org/mach/Tokamak> (Hämtad: 7 maj 2024).
- [21] S. Li, H. Jiang, Z. Ren och C. Xu, "Complex Boundary Value Problems of Nonlinear Differential Equations 2014," *Hindawi, Volume 2014*, jan. 2014. DOI: <https://doi.org/10.1155/2014/940965>. (Hämtad: 26 april 2024).
- [22] D. Gray et al., "Plasma exhaust and density control in tokamak fusion experiments with neutral beam or ICRF auxiliary heating," *Nuclear Fusion*, vol. 38, nr 11, s. 23, nov. 1998. DOI: 10.1088/0029-5515/38/11/301.
- [23] D. K. Cheng, *Field and Wave Electromagnetics 3. appl*. London, Storbritannien, Pearson, 2013.
- [24] M. Gordinier, J. Davis, F. Scott och K. Schultz, *Encyclopedia of Physical Science and Technology, 3. appl*. USA, Academic Press, 2001.
- [25] U. Rehman, A. Maqsood, M. Abbasi och E. Anjum, "Assessing the Efficacy of Two-Fluid Plasma Model and Magnetohydrodynamic Model in Simulating Resistive Drift Instability in Fusion Plasmas: A Comparative Study.," *IEEE Transactions on Plasma Science, Plasma Science, IEEE Transactions on, IEEE Trans. Plasma Sci*, vol. 52, nr 2, ss. 183–192, 2024. DOI: 10.1109/TPS.2024.3365710.

-
- [26] J. Seo et al., "Multimodal Prediction of Tearing Instabilities in a Tokamak," i *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023, ss. 1–8. DOI: 10.1109/IJCNN54540.2023.10191359.
- [27] H. de Blank, "Guiding Center Motion," *Fusion Science and Technology*, vol. 45, nr 2T, ss. 47–54, 2004. DOI: <https://doi.org/10.13182/FST04-A468>.
- [28] H. Y. Hiroaki Masuda Ko Higashitani, *Powder Technology Handbook*. CRC Press, 2006, kap. 2.1, s. 109.
- [29] J. P. Freidberg, *Plasma Physics and Fusion Energy: Transport*. Cambridge, Storbritannien, Cambridge University Press, 2007.
- [30] J. R. Duff, "Observation of Trapped-Electron Mode Microturbulence In Improved Confinement Reversed-Field Pinch Plasmas: Transport and Instabilities," doktorsavhandling, Wisconsin Plasma Physics Laboratory, Wisconsin, USA, 2017. [Online]. Tillgänglig: https://wippl.wisc.edu/pub_files/theses/thesisDuff133_2018.pdf.
- [31] J. Citrin et al., "Tractable flux-driven temperature, density, and rotation profile evolution with the quasilinear gyrokinetic transport model QuaLiKiz," *Plasma Physics and Controlled Fusion*, vol. 59, 2017. DOI: <https://doi.org/10.1088/1361-6587/aa8aeb>.
- [32] IBM. "What is machine learning?" (2023), [Online]. Tillgänglig: <https://www.ibm.com/topics/machine-learning> (Hämtad: 27 febr. 2024).
- [33] C. Forssén. "TIF385 Bayesian inference and machine learning: Neural networks." (2023), [Online]. Tillgänglig: <https://cforsen.gitlab.io/tif385-book/content/MachineLearning/NeuralNet.html> (Hämtad: 1 mars 2024).
- [34] IBM. "How do neural networks work?" (2023), [Online]. Tillgänglig: <https://www.ibm.com/topics/neural-networks> (Hämtad: 27 febr. 2024).
- [35] X. Glorot, A. Bordes och Y. Bengio, "Deep Sparse Rectifier Neural Networks," i *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, G. Gordon, D. Dunson och M. Dudík, red., ser. Proceedings of Machine Learning Research, vol. 15, Fort Lauderdale, FL, USA: PMLR, april 2011, ss. 315–323. [Online]. Tillgänglig: <https://proceedings.mlr.press/v15/glorot11a.html>.
- [36] H. Koyuncu, "Loss Function Selection in NN based Classifiers: Try-outs with a Novel Method," i *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2020, ss. 1–6. [Online]. Tillgänglig: <https://ieeexplore.ieee.org/document/9223208>.
- [37] I. Goodfellow, Y. Bengio och A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [38] J. B. Diederik P. Kingma, "Adam: A Method for Stochastic Optimization," i *3rd International Conference for Learning Representations*, San Diego, 2015. [Online]. Tillgänglig: <https://doi.org/10.48550/arXiv.1412.6980> (Hämtad: 19 april 2024).
- [39] P. Li, Y. Pei och J. Li, "A comprehensive survey on design and application of autoencoder in deep learning," *Applied Soft Computing*, vol. 138, s. 110176, 2023. DOI: <https://doi.org/10.1016/j.asoc.2023.110176>.
- [40] D. Bank, N. Koenigstein och R. Giryes, "Autoencoders," i *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*,

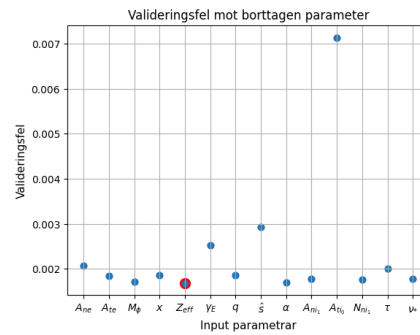
- L. Rokach, O. Maimon och E. Shmueli, red. Cham: Springer International Publishing, 2023, ss. 353–374. [Online]. Tillgänglig: https://doi.org/10.1007/978-3-031-24628-9_16.
- [41] A. Adadi och M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, ss. 52 138–52 160, 2018. DOI: 10.1109/ACCESS.2018.2870052.
- [42] J. Borsander, O. Lindberg, W. Rieck och M. Åqvist, "Surrogatmodell av QualiKiz för turbulenta instabiliteter i en tokamak," kandidatuppsats, Chalmers tekniska högskola, Göteborg, 2023. [Online]. Tillgänglig: <https://odr.chalmers.se/server/api/core/bitstreams/c9d30660-84f7-445d-bf02-eacba9eb4c89/content>.

A

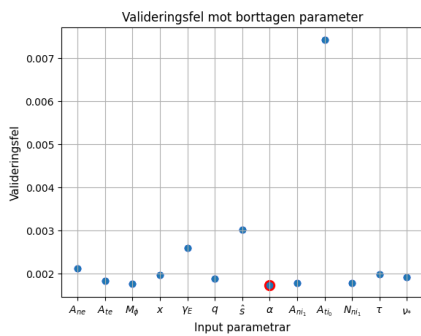
Grafer



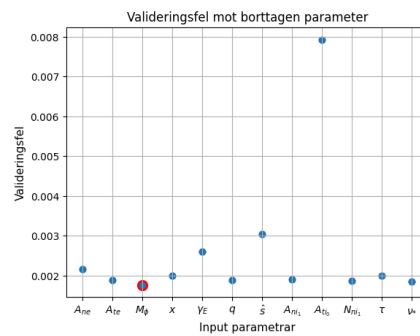
(a) 14 parametrar



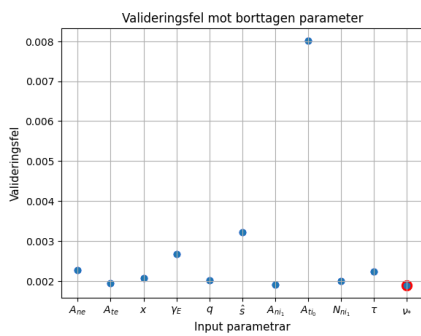
(b) 13 parametrar



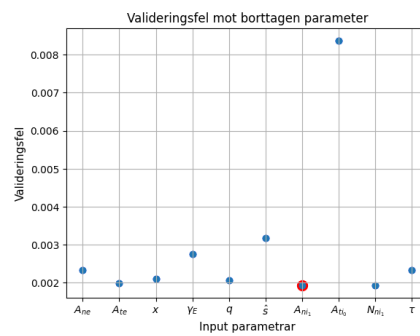
(c) 12 parametrar



(d) 11 parametrar

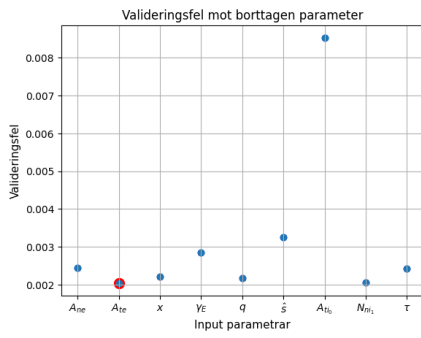


(e) 10 parametrar

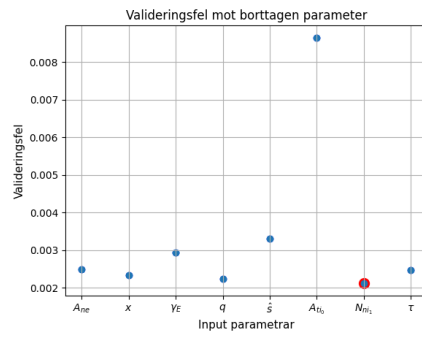


(f) 9 parametrar

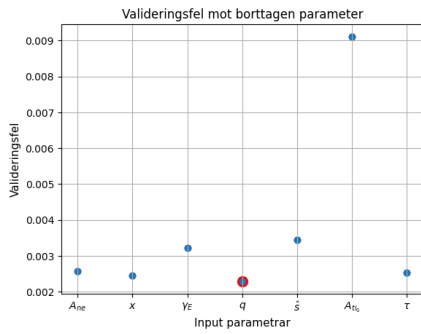
Figur A.1: Grafer som visar minskande antal parametrar använda för att återskapa outputsen



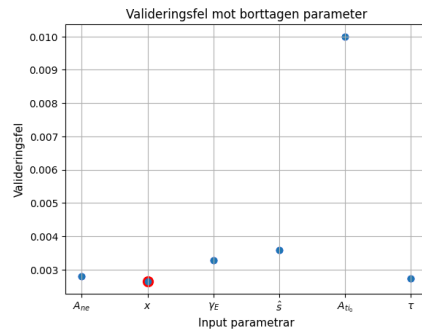
(g) 8 parametrar



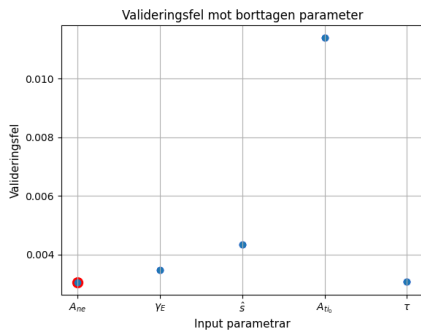
(h) 7 parametrar



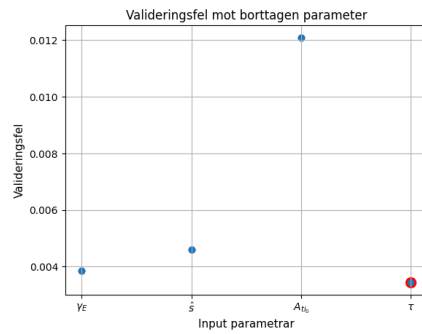
(i) 6 parametrar



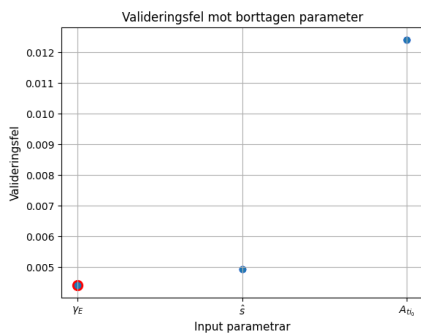
(j) 5 parametrar



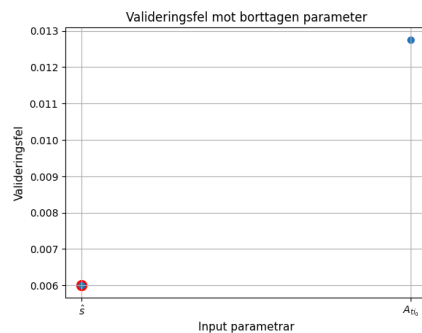
(k) 4 parametrar



(l) 3 parametrar

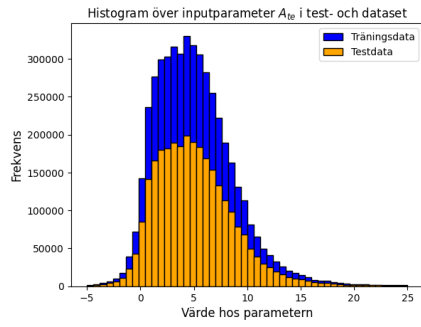


(m) 2 parametrar

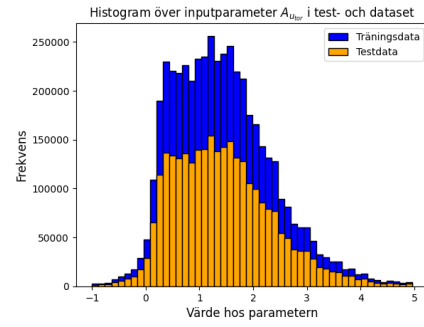


(n) 1 parameter

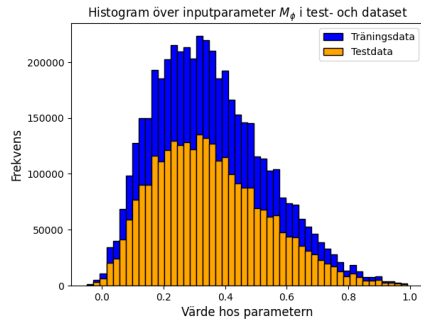
Figur A.1: Grafer som visar minskande antal parametrar använda för att återskapa outputs.



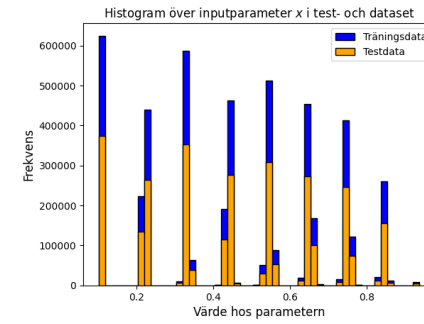
(a) A_{te}



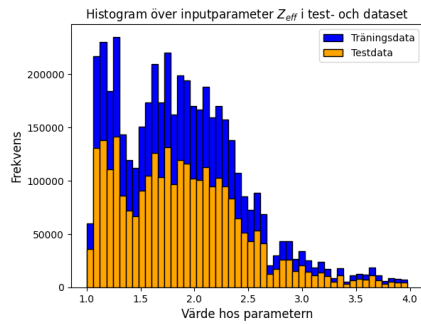
(b) $A_{u_{tor}}$



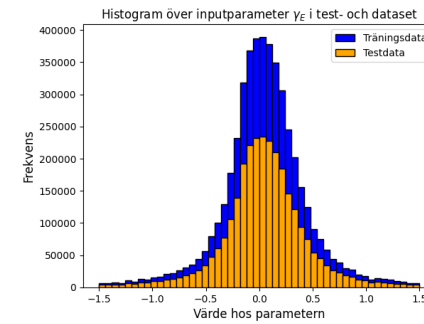
(c) M_ϕ



(d) x

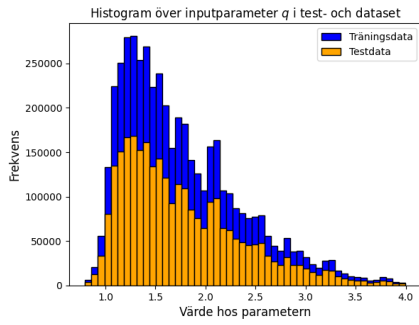


(e) Z_{eff}

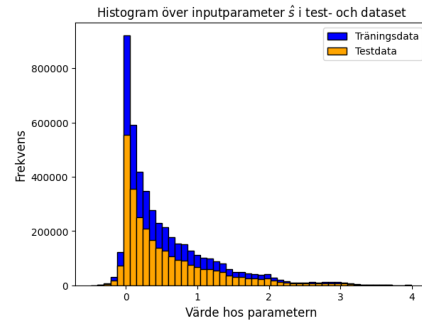


(f) γ_E

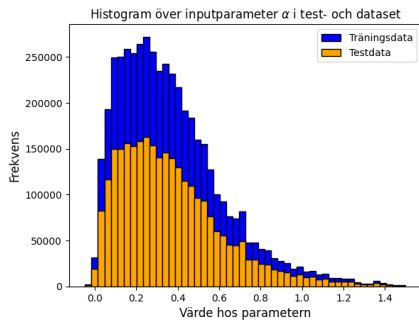
Figur A.2: Histogram över inputparametrernas värden i både träningsdata- och testdatasetet



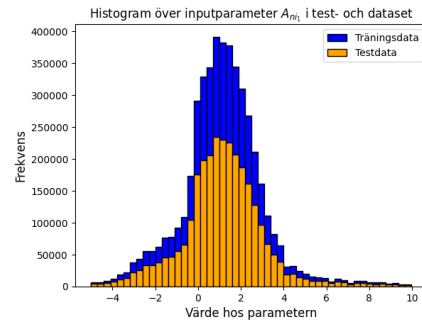
(g) q



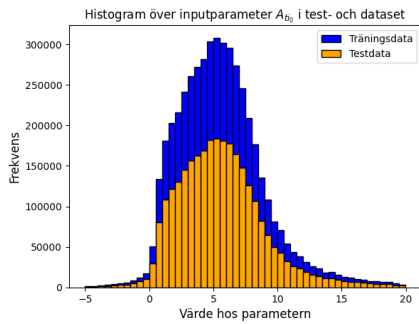
(h) \hat{s}



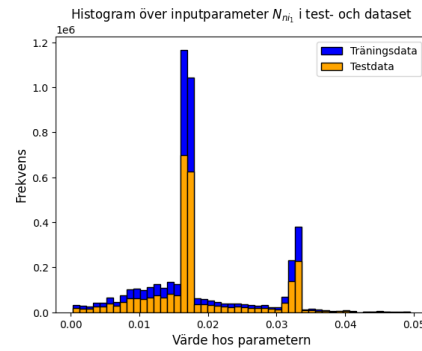
(i) α



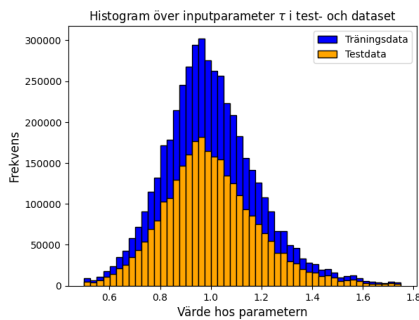
(j) A_{ni}



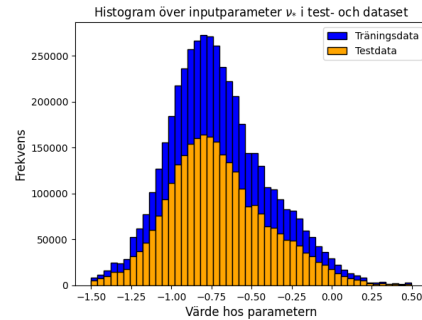
(k) A_{ti_0}



(l) N_{ni_1}

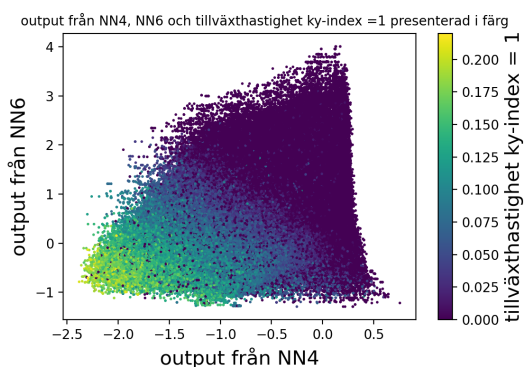


(m) τ

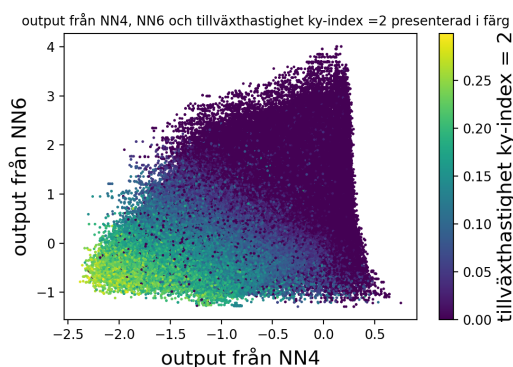


(n) ν_*

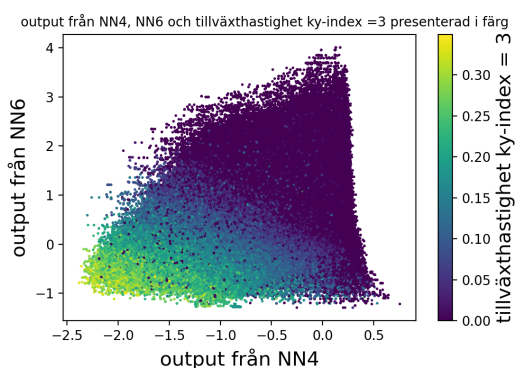
Figur A.2: Histogram över inputparametrernas värden i både träningsdata- och testdatasetet.



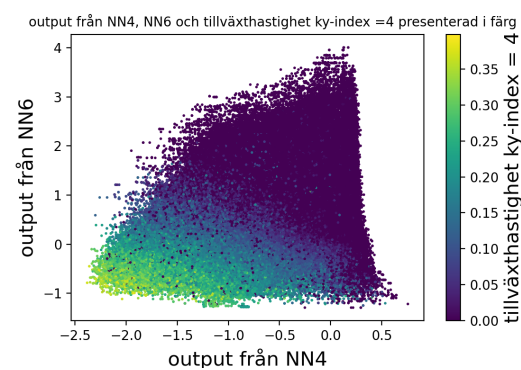
(a) ky-index = 1



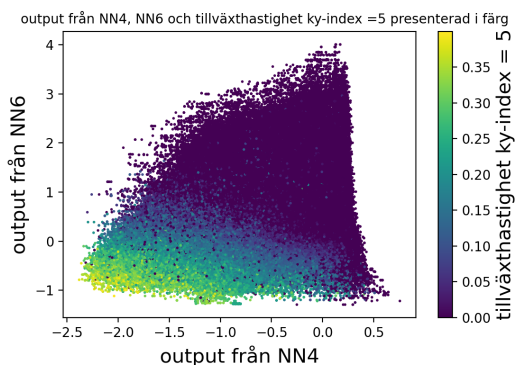
(b) ky-index = 2



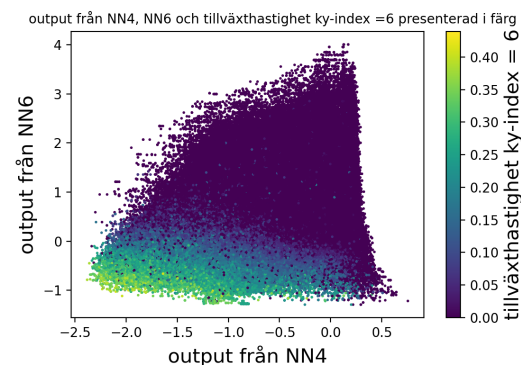
(c) ky-index = 3



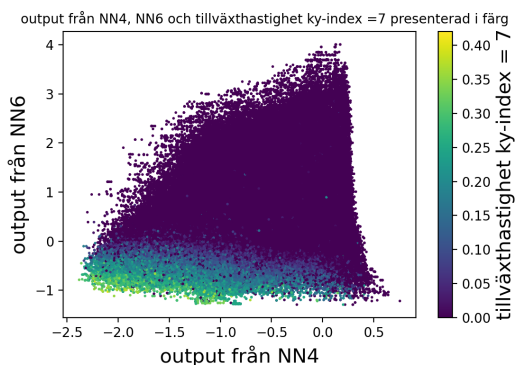
(d) ky-index = 4



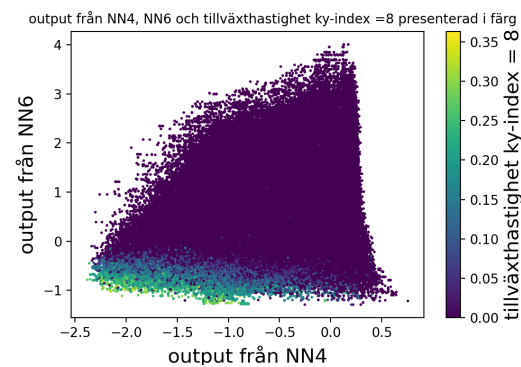
(e) ky-index = 5



(f) ky-index = 6



(g) ky-index = 7



(h) ky-index = 8

Figur A.3: Predikterade tillväxthastigheter för ky-index 1...8.

B

Tabeller

Tabell B.1: Medelkvadratfel MSE_i för de tre bästa modellerna med fem inputparametrar i båda grenarna samt standardavvikelser σ_i för dessa. Tabellen är sorterad efter MSE_i , där index 1 har lägst MSE_i .

i	$(G_{1,i,1}), (G_{1,i,2})$	Medelvärde MSE_i	σ_i
1	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, \mathbf{x}, \hat{s}, A_{ti_0}, \tau)$	0.002289374	$5.6911 \cdot 10^{-6}$
2	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, \gamma_E, \hat{s}, A_{ti_0}, \tau)$	0.002290242	$5.2818 \cdot 10^{-6}$
3	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, \tau), (A_{ne}, \mathbf{x}, \hat{s}, A_{ti_0}, \tau)$	0.002291681	$6.4074 \cdot 10^{-6}$
4	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, \tau), (A_{ne}, \gamma_E, \hat{s}, A_{ti_0}, \tau)$	0.002293302	$6.0155 \cdot 10^{-6}$
5	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, \tau), (\mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}, \tau)$	0.002315368	$4.9859 \cdot 10^{-6}$
6	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, \tau)$	0.002317406	$7.2568 \cdot 10^{-6}$
7	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (\mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}, \tau)$	0.002322046	$8.3359 \cdot 10^{-6}$
8	$(A_{ne}, \mathbf{x}, \hat{s}, A_{ti_0}, \tau), (\mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}, \tau)$	0.002326568	$6.8119 \cdot 10^{-6}$
9	$(A_{ne}, \gamma_E, \hat{s}, A_{ti_0}, \tau), (\mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}, \tau)$	0.002346610	$7.6464 \cdot 10^{-6}$
10	$(A_{ne}, \mathbf{x}, \gamma_E, A_{ti_0}, \tau), (A_{ne}, \mathbf{x}, \hat{s}, A_{ti_0}, \tau)$	0.002371496	$6.5564 \cdot 10^{-6}$
11	$(A_{ne}, \mathbf{x}, \gamma_E, A_{ti_0}, \tau), (\mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}, \tau)$	0.002393592	$5.4755 \cdot 10^{-6}$
12	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, \tau), (A_{ne}, \mathbf{x}, \gamma_E, A_{ti_0}, \tau)$	0.002401700	$2.1627 \cdot 10^{-5}$
13	$(A_{ne}, \mathbf{x}, \gamma_E, A_{ti_0}, \tau), (A_{ne}, \gamma_E, \hat{s}, A_{ti_0}, \tau)$	0.002427563	$2.3625 \cdot 10^{-6}$
14	$(A_{ne}, \mathbf{x}, \hat{s}, A_{ti_0}, \tau), (A_{ne}, \gamma_E, \hat{s}, A_{ti_0}, \tau)$	0.002432467	$5.5810 \cdot 10^{-6}$
15	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, \mathbf{x}, \gamma_E, A_{ti_0}, \tau)$	0.002495518	$9.5680 \cdot 10^{-7}$

Tabell B.2: Medelkvadratfel MSE_i för de tre bästa modellerna med fem inputparametrar i en gren och fyra i den andra samt standardavvikelser σ_i för dessa. Tabellen är sorterad efter MSE_i , där index 1 har lägst MSE_i .

i	Namn	Medelvärde MSE_i	σ_i
1	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, \hat{s}, A_{ti_0}, \tau)$	0.0023121	$5.3897 \cdot 10^{-6}$
2	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, \gamma_E, \hat{s}, \tau)$	0.0023435	$6.7731 \cdot 10^{-6}$
3	$(A_{ne}, \mathbf{x}, \hat{s}, A_{ti_0}, \tau), (\mathbf{x}, \gamma_E, \hat{s}, A_{ti_0})$	0.0023636	$6.8521 \cdot 10^{-6}$
4	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (\gamma_E, \hat{s}, A_{ti_0}, \tau)$	0.0023663	$6.0224 \cdot 10^{-6}$
5	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (\mathbf{x}, \hat{s}, A_{ti_0}, \tau)$	0.0023684	$5.9683 \cdot 10^{-6}$
6	$(A_{ne}, \mathbf{x}, \hat{s}, A_{ti_0}, \tau), (A_{ne}, \mathbf{x}, \gamma_E, \hat{s})$	0.0023828	$9.5843 \cdot 10^{-6}$
7	$(A_{ne}, \gamma_E, \hat{s}, A_{ti_0}, \tau), (A_{ne}, \mathbf{x}, \gamma_E, \hat{s})$	0.0023830	$5.6249 \cdot 10^{-6}$
8	$(A_{ne}, \gamma_E, \hat{s}, A_{ti_0}, \tau), (\mathbf{x}, \gamma_E, \hat{s}, A_{ti_0})$	0.0024021	$7.8739 \cdot 10^{-6}$
9	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, \mathbf{x}, \hat{s}, \tau)$	0.0024062	$1.0436 \cdot 10^{-5}$
10	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, \mathbf{x}, A_{ti_0}, \tau)$	0.0025039	$6.4619 \cdot 10^{-6}$
11	$(A_{ne}, \gamma_E, \hat{s}, A_{ti_0}, \tau), (A_{ne}, \mathbf{x}, \gamma_E, A_{ti_0})$	0.0025238	$9.4787 \cdot 10^{-6}$
12	$(A_{ne}, \mathbf{x}, \hat{s}, A_{ti_0}, \tau), (A_{ne}, \mathbf{x}, \hat{s}, A_{ti_0})$	0.0025454	$7.9349 \cdot 10^{-6}$
13	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, \gamma_E, A_{ti_0}, \tau)$	0.0025537	$6.3885 \cdot 10^{-6}$
14	$(A_{ne}, \gamma_E, \hat{s}, A_{ti_0}, \tau), (A_{ne}, \mathbf{x}, \hat{s}, A_{ti_0})$	0.0025661	$3.9378 \cdot 10^{-6}$
15	$(A_{ne}, \mathbf{x}, \hat{s}, A_{ti_0}, \tau), (A_{ne}, \mathbf{x}, \gamma_E, A_{ti_0})$	0.0026372	$8.2659 \cdot 10^{-6}$

Tabell B.3: Medelkvadratfel MSE_i för de tre bästa modellerna med fyra inputparametrar i samma gren samt standardavvikelser σ_i för dessa. Tabellen är sorterad efter MSE_i , där index 1 har lägst MSE_i .

i	Namn	Medelvärde MSE_i	σ_i
1	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}), (A_{ne}, \hat{s}, A_{ti_0}, \tau)$	0.0024149	$4.5999 \cdot 10^{-6}$
2	$(\mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, \hat{s}, A_{ti_0}, \tau)$	0.0024453	$6.8494 \cdot 10^{-6}$
3	$(A_{ne}, \mathbf{x}, \gamma_E, A_{ti_0}), (A_{ne}, \hat{s}, A_{ti_0}, \tau)$	0.0025386	$3.4276 \cdot 10^{-6}$

Tabell B.4: Grenstruktur och medelkvadratfel MSE_i för strukturerna med fem inputparametrar i en gren och tre i den andra, samt standardavvikelser σ_i för dessa. Kombinationerna är sorterade efter MSE_i , och den bästa kombinationen är markerad med fetstil.

i	$(G_{4,i,1}), (G_{4,i,2})$	Medelvärde MSE_i	σ_i
1	$(\mathbf{A}_{ne}, \mathbf{x}, \gamma_E, \hat{s}, \mathbf{A}_{ti_0}), (\mathbf{A}_{ti_0}, \hat{s}, \tau)$	0.0023955	$3.1591 \cdot 10^{-6}$
2	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ne}), (\hat{s}, A_{ti_0}, \tau)$	0.0024276	$7.3801 \cdot 10^{-6}$
3	$(A_{ne}, \mathbf{x}, \gamma_E, \hat{s}, A_{ti_0}), (A_{ne}, A_{ti_0}, \tau)$	0.0026898	$1.9465 \cdot 10^{-5}$

B. Tabeller

Tabell B.5: Medelkvadratfel MSE_i för de kombinationer där en parameter brutits loss från grenen med fem inputparametrar, samt σ_i för dessa. Tabellen är sorterad efter MSE_i , där index 1 har lägst MSE_i .

i	$(G_{4,i,1}), (G_{4,i,2})$	Medelvärde MSE_i	σ_i
1	$((\mathbf{A}_{ne}, \mathbf{x}, \gamma_E, \hat{\mathbf{s}}), (\mathbf{A}_{ti_0}), (\mathbf{A}_{ti_0}, \hat{\mathbf{s}}, \tau)$	0.0024374	$8.3607 \cdot 10^{-6}$
2	$(A_{ne}, \gamma_E, \hat{\mathbf{s}}, A_{ti_0}), (x), (\hat{\mathbf{s}}, A_{ti_0}, \tau)$	0.0024795	$1.7574 \cdot 10^{-6}$
3	$(A_{ne}, x, \gamma_E, A_{ti_0}), (\hat{\mathbf{s}}), (\hat{\mathbf{s}}, A_{ti_0}, \tau)$	0.0025302	$7.4556 \cdot 10^{-6}$
4	$(x, \gamma_E, \hat{\mathbf{s}}, A_{ti_0}), (A_{ne}), (\hat{\mathbf{s}}, A_{ti_0}, \tau)$	0.0025325	$1.1201 \cdot 10^{-5}$
5	$(A_{ne}, x, \hat{\mathbf{s}}, A_{ti_0}), (\gamma_E), (\hat{\mathbf{s}}, A_{ti_0}, \tau)$	0.0025750	$4.1824 \cdot 10^{-5}$

Tabell B.6: Medelkvadratfel MSE_i för strukturerna där tre parametrar brutits loss från grenen med fem inputparametrar, samt standardavvikelser σ_i för dessa. Tabellen är sorterad efter MSE_i , där index 1 har lägst MSE_i .

i	$((G_{6,i,11}), (G_{6,i,12}), (G_{6,i,13}), (G_{6,i,2}))$	Medelvärde MSE_i	σ_i
1	$((\mathbf{A}_{ne}, \hat{\mathbf{s}}), (\gamma_E), (\mathbf{x}), (\mathbf{A}_{ti_0}), (\mathbf{A}_{ti_0}, \hat{\mathbf{s}}, \tau)$	0.0024881	$3.9861 \cdot 10^{-6}$
2	$((\gamma_E, \hat{\mathbf{s}}), (x), (A_{ne}), (A_{ti_0}), (\hat{\mathbf{s}}, A_{ti_0}, \tau)$	0.0024937	$4.9382 \cdot 10^{-6}$
3	$((A_{ne}, \gamma_E), (x), \hat{\mathbf{s}}), (A_{ti_0}), (\hat{\mathbf{s}}, A_{ti_0}, \tau)$	0.0025309	$4.1010 \cdot 10^{-6}$
4	$((\hat{\mathbf{s}}, x), (\gamma_E), (A_{ne}), (A_{ti_0}), (\hat{\mathbf{s}}, A_{ti_0}, \tau)$	0.0025403	$8.0208 \cdot 10^{-6}$
5	$((\gamma_E, x), (\hat{\mathbf{s}}), (A_{ne}), (A_{ti_0}), (\hat{\mathbf{s}}, A_{ti_0}, \tau)$	0.0025438	$1.1844 \cdot 10^{-5}$
6	$((A_{ne}, x), (\hat{\mathbf{s}}), (\gamma_E), (A_{ti_0}), (\hat{\mathbf{s}}, A_{ti_0}, \tau)$	0.0025830	$1.2461 \cdot 10^{-6}$

Tabell B.7: Medelkvadratfel MSE_i för strukturerna där en parameter brutits loss från grenen med tre inputparametrar, samt standardavvikelser σ_i för dessa. Tabellen är sorterad efter MSE_i , där index 1 har lägst MSE_i .

i	$((G_{6,i,11}), (G_{6,i,12}), (G_{6,i,13}), (G_{6,i,2}))$	Medelvärde MSE_i	σ_i
1	$((\gamma_E, \hat{\mathbf{s}}), (\mathbf{x}), (\mathbf{A}_{ne}), (\mathbf{A}_{ti_0}), ((\mathbf{A}_{ti_0}, \tau), (\hat{\mathbf{s}}))$	0.0024197	$3.7399 \cdot 10^{-6}$
2	$((A_{ne}, \hat{\mathbf{s}}), (x), (\gamma_E), (A_{ti_0}), (A_{ti_0}, \tau), (\hat{\mathbf{s}}))$	0.0024360	$2.7205 \cdot 10^{-6}$
3	$((\gamma_E, \hat{\mathbf{s}}), (x), (A_{ne}), (A_{ti_0}), (\hat{\mathbf{s}}, \tau), (A_{ti_0}))$	0.0024435	$4.1899 \cdot 10^{-6}$
4	$((A_{ne}, \hat{\mathbf{s}}), (x), (\gamma_E), (A_{ti_0}), (\hat{\mathbf{s}}, \tau), (A_{ti_0}))$	0.0024524	$3.2704 \cdot 10^{-6}$
5	$((A_{ne}, \hat{\mathbf{s}}), (x), (\gamma_E), (A_{ti_0}), (\hat{\mathbf{s}}, A_{ti_0}), (\tau))$	0.0024573	$8.0234 \cdot 10^{-6}$
6	$((\gamma_E, \hat{\mathbf{s}}), (x), (A_{ne}), (A_{ti_0}), (\hat{\mathbf{s}}, A_{ti_0}), (\tau))$	0.0024610	$3.62095 \cdot 10^{-6}$

Institutionen för rymd-, geo- och miljövetenskap
Chalmers tekniska högskola
Göteborg, Sverige 2024
www.chalmers.se



CHALMERS