NTNU
Norwegian University of
Science and Technology

Empirical Prediction of Residuary Resistance of Fast Catamarans

Even Wollebæk Førrisdal

Marine Technology
Submission date: June 2018
Supervisor: Sverre Steen, IMT
Co-supervisor: Hans Jørgen Rambech, SINTEF Ocean

Norwegian University of Science and Technology
Department of Marine Technology

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MASTER THESIS IN MARINE TECHNOLOGY

SPRING 2018

FOR

Even Wollebæk Førrisdal

Empirical prediction of resistance of fast catamarans

In the early design stage, often before the hull lines are designed, there is a need for a quick resistance calculation method, that requires little input. Such methods are usually based on empirical data. SINTEF Ocean has an empirical resistance calculation methods for fast catamarans, named CatRes, based on regression on model tests performed at MARINTEK up to about 1997. They see a need to update this method with data from more recent model tests. One should also consider if the chosen input variables are sufficient, or if other parameters should be considered. In addition, a more complex and advanced type of empirical method might be applied, such as for instance Artifcial Neural Networks.

On this basis, the main objective of the master thesis is to develop an improved empirical resistance prediction method for fast catamarans, implemented in a computer tool. The computer tool shall be made so that it is suitable for implementation in a larger software system. The resistance prediction method shall be fully described in the thesis. It shall be tested and validated as far as the available data allows, and the validation process and results shall be thoroughly described in the thesis. It is also important that the range of validity and the limitations of the resistance prediction method have been defined and are stated in the thesis.

In the thesis the candidate shall present his personal contribution to the resolution of problem within the scope of the thesis work.

Theories and conclusions shall be based on mathematical derivations and/or logic reasoning identifying the various steps in the deduction.

The thesis work shall be based on the current state of knowledge in the field of study. The current state of knowledge shall be established through a thorough literature study, the results of this study shall be written into the thesis. The candidate should utilize the existing possibilities for obtaining relevant literature.

The thesis shall be organized in a rational manner to give a clear exposition of results, assessments, and conclusions. The text should be brief and to the point, with a clear language. Telegraphic language should be avoided.

The thesis shall contain the following elements: A text defining the scope, preface, list of contents, summary, main body of thesis, conclusions with recommendations for further work, list of symbols and acronyms, reference and (optional) appendices. All figures, tables and equations shall be numerated.

The supervisor may require that the candidate, in an early stage of the work, present a written plan for the completion of the work. The plan shall include a budget for the use of laboratory or other resources that will be charged to the department. Overruns shall be reported to the supervisor.

The original contribution of the candidate and material taken from other sources shall be clearly defined. Work from other sources shall be properly referenced using an acknowledged referencing system.

The thesis shall be submitted electronically (pdf) in DAIM:

- Signed by the candidate
- The text defining the scope (this text) (signed by the supervisor) included
- Computer code, input files, videos and other electronic appendages can be uploaded in a zip-file in DAIM. Any electronic appendages shall be listed in the main thesis.

The candidate will receive a printed copy of the thesis.

Supervisor	: Professor Sverre Steen
Co-supervisor	: Hans Jørgen Rambech (SINTEF Ocean)
Start	: 15.01.2018
Deadline	: 11.06.2018

Trondheim, 15.01.2018

Sverre Steen
Supervisor

Preface

This thesis concludes my time as a student at Department of Marine Technology at the Norwegian University of Science and Technology. The work has been carried out during the spring of 2018 and is the final requirement for the degree of *Master of Science (Sivilingeniør).*

The work builds on the literature review carried out by the author in the fall of 2017, where the main focus was on current methods in empirical resistance prediction for fast catamarans and machine learning. Now, the work is continued by applying machine learning with an emphasis on artificial neural networks, to resistance data from model tests of fast displacement catamarans. Working on these topics have been both interesting and challenging. This thesis accounts for one hundred percent of the grade in the course TMR4930.

I would like to thank supervisor Professor Sverre Steen for the guidance throughout the project and for the opportunity of working with these interesting topics. Additionally, I would like to thank co-supervisor Hans Jørgen Rambech for the help with resistance data, and for discussions along the way. I must also thank SINTEF Ocean for giving me access to all their resistance reports - training neural networks without data would have been impossible. Lastly, I would like to thank my fellow students from office A2.011 for valuable discussions, and for creating an encouraging and ambitious atmosphere.

Trondheim, 11.06.2018

Even W. Førrisdal

Even Wollebæk Førrisdal

Abstract

This thesis focuses on resistance of fast displacement catamarans, with the purpose of developing an empirical residuary resistance estimation method based on artificial neural networks. The study builds on work carried out by Rambech [1998], where resistance data from model trials carried out by MARINTEK from 1990 to 1997 was analysed and an empirical resistance prediction method was made. This method, named CatRES, produces conservative resistance estimations for newer designs, and SINTEF Ocean has proposed the task of including model trials carried out from 1997 to 2017, and develop an improved empirical resistance estimation method. The total data set consists of 2313 samples.

After doing an extensive parameter study of neural network design and training the input parameters in Table 1 gave the best performing network with model input parameters: Fn, B/T, $L/\nabla^{1/3}$, $S/\nabla^{2/3}$, $s2/Lwl$ and Sb/S. The parameter range of the training data, and therefore the validity range of the empirical model is presented in Table 4.1. The best performing neural network found through the study was trained for 996 epochs. Through validation analysis, carried out using an independent data from Molland et al. [1995], signs of significant overtraining was observed. Training for fewer epochs yielded better generalisation ability as the prediction error for the independent data set decreased, and 19 epochs proved to be the best number of training epochs. Prediction ability was assessed by calculating mean squared error (MSE) between predicted and correct results. MSE of 0.1135 was obtained for the SINTEF Ocean dataset, and an MSE of 0.2072 was obtained for the independent data set.

Table 1: Optimal neural network training parameters, found through parameter study.

Parameter	Best Alternative
Training function	*trainbr*
Node distribution in hidden layers	[10-11-11-10]
Dataset	Full available dataset
Best number of training epochs	19

Overtraining and overfitting were avoided using built-in tools in Matlab and analysis with the independent data set. The consequence of exceeding parameter validity range was analysed, where the model predicted the general behaviour, but fails to predict residuary resistance perfectly. A missing input analysis was also carried out, where setting missing input equal to SINTEF Ocean sample mean yields the best prediction ability.

Sammendrag

Denne oppgaven fokuserer på motstand hos hurtiggående katamaraner, der hensikten er å utvikle en empirisk mostandsmodell for å estimere restmostand ved hjelp av kunstige nevrale nettverk. Studiet bygger på arbeidet gjort av Rambech [1998], der modellforsøk gjort av MARINTEK fra 1990 til 1997 ble benyttet til å utvikle en empirisk motstandsmodell. Metoden, kalt CatRES, har begynt å gi konservative estimater for nyere katamaran design, og SINTEF Ocean har derfor foreslått oppgaven med å inkludere slepeforsøk gjort fra 1997 til 2017, for å utvikle en forbedret empirisk motstandsmodell. Det totale datagrunnlaget er på 2313 datapunkter.

Etter et grundig parameterstudie av nevralt nettverksdesign i Matlab, ble den optimale kombinasjonen av designparametre presentert i Tabell 2 funnet. Den optimale parameterkombinasjonen for input-verdier til nettverket er: Fn, B/T, $L/\nabla^{1/3}$, $S/\nabla^{2/3}$, $s2/Lwl$ og Sb/S. Parameterområdet, som også er det nevrale nettverkets validitetsområde, er presentert i Tabell 4.1 Det best presterende nettverket funnet gjennom parameterstudier ble trent i 996 epoker. Gjennom verifiseringsanalyser gjort med et uavhengig datasett fra Molland et al. [1995], ble tegn på overtrening hos nettverket observert. Trening i færre epoker ble gjennomført for å øke generaliseringsevnen til nettverket. Prediksjonsevnen for det uavhengige datasettet økte drastisk ved trening i færre epoker, og en bedre generaliseringsevne ble oppnådd. Trening i 19 epoker viste seg å gi best ytelse. Prediksjonsevnen ble validert ved mean squared error (MSE), som ga en MSE på 0.1135 for datasettet gitt av SINTEF Ocean og en MSE på 0.2072 for det uavhengige datasettet.

Table 2: Optimale treningsparametre funnet gjennom parameterstudiet.

Parameter	Beste Alternativ
Treningsfunksjon	*trainbr*
Node distribusjon i skjulte lag	[10-11-11-10]
Datasett	Fullt tilgjengelig datasett
Optimalt antall treningsepoker	19

Overtrening og over-tilpassing har blitt unngått ved bruk av innebygde verktøy i Matlab, og ved verifisering og validering med uavhengig datasett. Effekten av å gå utenfor nettverkets gyldighetsområde har blitt analysert, hvor modellen klarte å predikere generell oppførsel, men ikke klarte å gjengi perfekte resultater. En analyse på konsekvens av manglende input-verdier ble også gjennomført, hvor det å sette manglende verdier lik gjennomsnittlig verdi hos modellene fra datasettet hos SINTEF Ocean ga best resultater.

Abbreviations

ANN	**A**rtificial **N**eural **N**etworks
API	**A**pplication **P**rogramming **I**nterface
CFD	**C**omputational **F**luid **D**ynamics
CPU	**C**entral **P**rocessing **U**nit
EY	Former **E**rnst and **Y**oung
GUI	**G**raphical **U**ser **I**nterface
ITTC	**I**nternational **T**owig **T**ank **C**onference
MAP	**M**aximum **A** **P**osteriori
ML	**M**achine **L**earning
MSE	**M**ean **S**quare **E**rror
No.	**Num**ber
NPL	**N**ational **P**hysical **L**aboratory
POLY	**Poly**nomial
RAM	**R**andom **A**ccess **M**emory
SUM	**Sum**mation

List of Symbols

α and β	In ANN regularisation: regularisation parameters
β	Correction of form factor due to the presence of other hull
$\boldsymbol{e}$	Error vector
$\boldsymbol{H}^{\boldsymbol{MAP}}$	Hessian matrix
$\boldsymbol{J}$	Jacobian matrix
ΔC_F	Roughness correction
δ	Errors in Backpropagation
γ	Correction of wave resistance due to the presence of the other hull
$\hat{Y_i}$	Prediction of target value from proposed model
λ	Regularisation coefficient
μ	Levenberg's damping factor
∇	Volume displacement
ρ	Density of water
ρ_{air}	Density of air
τ	Wave resistance interference factor
$\mathbf{w}$	Network weight vector
φ	Fullness parameter MARINTEK
$\widetilde{E}(W)$	Regulised performance function
a	Output from transfer function in neuron
A_P	Projected area above the waterline, normal to the direction of motion
B	Hull width
b	Bias added to input into neuron
$B(z)$	Width of transom at different draughts
b_i	Regression parameter in Molland 1994 method
Bwl	Total breadth of waterline
bwl	Breadth single hull

C_M	Midship section coefficient
C_{AA}	Air resistance coefficient
C_{Air}	Air resistance coefficient for the superstructure of the vessel
C_A	Correlation factor coefficient
C_{BD}	Transom stern resistance coefficient
C_B	Block coefficient
C_F	Frictional coefficient as defined by ITTC'57 friction correlation line
C_{Ri}	Froude number dependent coefficient i in CatRES method
C_R or C_r	Residuary resistance coefficient
C_{Tsi}	Froude number dependent coefficient i in CatRES method
C_{Ts}^{emp}	C_T found from empirical model in CatRES
C_T	Total resistance coefficient
C_{VM}	Wave making resistance coefficient
$C_W(F_N)$	Wave pattern resistance coefficient, without the presence of other hulls
C_W^{emp}	C_W found from empirical model in CatRES
C_X	Resistance coefficient of force R_X
Cb	Block coefficient
Cm	Midship section coefficient
Cp	Prismatic coefficient
D	Training data set
$Dtap$	Tunnel height at AP
$Dtfp$	Tunnel height at FP
$E(W)$	Performance function
E_W	Sum of all network weights
F_{NT}	Froude number for transom
Fn	Froude number
$ftap$	Air gap tunnel at AP
$ftfp$	Air gap tunnel at FP
g	Gravitational acceleration
H	Roughness of hull
$h_{\boldsymbol{w}}(\boldsymbol{x})$	Hypothesis function
H_D	Maximum depth of transom, varies with the trim angle and sinkage
k	Form factor without presence of other hull
k_i	Empirically calculated Froude number dependent, coefficients
L	Hull length
L_{PP}	Length of perpendiculars
LCB	Longitudinal center of buoyancy
Loa	Length over all

Lpp	Length between perpendiculars
Lwl	Length waterline
M	Characteristic neural network architecture
m	A constant in Prohaska's method, which gives a liner behaviour of $\frac{F_N^n}{C_{Fm}}$
m	In neural networks: total number of network parameters
N	Number of points in MSE
n	Number of observations
Pe	Required power
R_{AA}	Air resistance
$R_{Transom}$	Hydrostatic resistance on transom
R_X	Resistance force X
Rn	Reynolds number
Rtm	Resistance of model in test
S	Surface a force is acting on (wetted area for forces from water, projected area for air forces)
$s1$	Distance between centerlines
$s2$ or s	Distance between hulls
S_B	Transom stern area
T	Draught
T_{AP}	Draught at aft perpendicular
T_{FP}	Draught at front perpendicular
$Trim$	Initial trim at zero speed
V	Relative velocity between ship and air
$V_{Viscous}$	Calculated viscous resistance
Vm	Model speed
Vs	Full scale ship speed
w	Weigh multiplied to input into neuron
Y_i	Target value
y_k	Output from feed-forward neural network
Z_E	Integral to evaluate α and β
z_j	Activation of given output from nonlinear activation function

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Getting a rough estimate of a ships resistance without doing high-cost model trials or complicated computational fluid dynamics (CFD) analysis, is desirable in the early phase of a ship design process. Even before the hull lines are designed, there is a need for quick resistance estimation methods which require little input. An empirical resistance model for fast catamarans named CatRES was constructed based on the database of model trials performed by MARINTEK up to about 1997. Several model trials have been carried out in the last twenty years, and SINTEF Ocean has experienced CatRes to give conservative resistance predictions. They have therefore expressed a desire to update CatRES, with the new resistance data, and look for possibilities to enhance the current method.

In his project thesis, the author carried out a literature study in the field of empirical resistance prediction models for catamarans. The different aspects of resistance on catamarans were presented together with established methods for resistance estimation. Two of these methods; CatRES [Rambech, 1998] and Molland et al. [1995] method was presented in detail, while the empirical and semi-empirical methods of Sahoo et al. [2007] and Xuan et al. [2001] were briefly discussed. CatRES is a fourth order polynomial where the slenderness-ratio $\frac{L}{\nabla^{1/3}}$ is used to estimate the resistance coefficient. The method was developed using regression on model trials carried out from 1990-1997 by MARINTEK, where the models are mainly car- and passenger ferries, and cargo catamarans with a few models being fishing vessels for line fishing. Many new designs have been tested in the period from 1997 to 2017, and several of these are of the same vessel types. By including recent model trials from ferries and some other vessels, 1231 trials are added to the original dataset of 1082 trials, creating a sample of 2313 data points. New designs appear to have lower slenderness-ratio and higher Lwl/Bwl-ratio than the older designs. Higher displacement is a characteristic for planing vessels, as the hull is lifted partly out of the water when travelling at higher speeds. When screening the models together with SINTEF Ocean, planing vessels was however filtered out, as displacement and semi-displacement catamarans were desired. CatRES and Molland et al. [1995] method are both polynomial fitted models, using the dimensionless parameters: $\frac{L}{B}$, $\frac{B}{T}$ $\frac{L}{\nabla^{1/3}}$ and $\frac{s}{L}$ multiplied to Froude number dependent coefficients, to calculate resistance. The Froude number dependent coefficients are calculated for discrete Froude numbers, making an interpolation between adjacent Froude numbers necessary when used for non-discrete Froude numbers.

Machine learning, with emphasis on artificial neural networks (ANNs), was proposed as a feasible way of developing an empirical resistance prediction method for fast catamarans in the project thesis. Machine learning has gained serious momentum in the past couple of years as the available computer power has increased and computer cost has decreased, and therefore it is applicable to several fields of engineering. In marine applications, ANNs have been used in performance and cavitation estimation for propellers [Shora et al., 2017], analysis of spread mooring configurations for floating production systems [de Pina et al., 2016] and performance prediction of hydrofoil supported catamarans [Najafi et al., 2018].

One of the strengths of machine learning and artificial neural networks is their ability to find patterns in data, without the need of physical relationships or ”logic” relationships. Both CatRES and Molland et al. [1995] method used regression for discrete Froude numbers and uses these coefficients to further estimate resistance. By looking at the regression lines for the discrete Froude numbers for CatRES, coefficients of determinations in the range $0.9630 - 0.9855$ are observed [Rambech, 1998, Appendix C.7]. These R-squared values close to 1 indicates a good fit for the discrete Froude numbers, but how will the method perform in between Froude numbers? Will the method lose its accuracy when adjacent coefficients are interpolated? CatRES is not evaluated in this thesis, but a qualified guess is: probably.

Both CatRES and Molland et al. [1995] method were developed before the year 2000 and catamaran designs appear to have changed slightly since then. CatRES produces conservative estimated when used on newer designs, indicating poor performance for catamaran designs diverging from the designs between 1990-1997. Creating a new empirical resistance estimation method, using data from both older and newer designs, is therefore desired in order to find a model which yields good estimation for all designs. The use of artificial neural networks on resistance data from catamarans appears to be absent in published literature, which is something the author finds strange. Artificial neural networks appear to be applicable, but maybe the data available to other researchers is too limited to obtain well trained neural networks. The focus of this thesis will be on developing an empirical resistance prediction method for fast catamarans, using machine learning trained on data from model trials carried out by SINTEF Ocean. The intention is to find a method with high generalisation ability, which is suitable for estimating the resistance of both old and new designs. In the next sections, the objectives and scope of work are presented.

1.2 Objectives and Scope

The objective of this master thesis is to build on the preliminary work carried out in the project thesis and develop an improved empirical resistance prediction method for fast catamarans. The method shall be implemented in a computer tool suitable for implementation in a larger software system.

In order for the resistance model to be accurate, several tasks had to be carried out such as; extensive pre-processing of data from model trials, data validation, parameter study, training of the different neural networks, performance comparison, network validity confirmation and exportation of finished model. Lots of time have been put into digitisation of the resistance data, as most of the resistance data are found in reports in PDF-format, and therefore numbers had to be manually written into electronic format. Then comes the questions; which vessel parameters should be included in the empirical model? Which vessel parameters has the most impact on the performance of the neural networks? What is the validity range of the empirical model and how does it behave outside its validity range? These questions found the basis of this thesis, which have been broken down into the following objectives:

- The procedure of how to predict resistance for fast catamarans from model tests will be outlined.
- An overview of the established empirical resistance prediction models will be presented, representing the current state of knowledge in the field.
- Artificial neural network theory is presented.
- A catamaran parameter study is carried out, based on the available model resistance data from SINTEF Ocean. Ranges of the different model and ship parameters are presented.
- A parameter study of the neural network input-parameters in Matlab is carried out, and the best combination of parameters for this case is found.

- Results from artificial neural network training and validation will be presented, and a most feasible solution is proposed.
- The range of validity and limitations of the proposed method are tested, presented and discussed.

1.3 Thesis Outline

The rest of the thesis is structured as follows.

- Chapter 2 presents a review of theory related to the topics in this thesis. Ship resistance with an emphasis on resistance of catamarans, a review of current empirical resistance prediction methods and theory of machine learning with emphasis on artificial neural networks are presented. Some of the material is based on the literature study carried out by the author last fall, but the text has been modified and topics have been added.
- Chapter 3 contains results from this study. The data is presented, with available model data and its ranges, before a parameter study of input parameters when using Neural Network Toolbox in Matlab for designing artificial neural networks, is carried out. Then the optimal parameters are combined into training an optimal neural network. Lastly, this network is evaluated, by assessing the generalisation ability and parameter studies.
- Chapter 4 presents a discussion of the results, and how the choices made in this study affects the validity, accuracy and usage of this empirical resistance prediction method.
- Chapter 5 contains the final conclusion on the work that has been carried out, with a summary of remarks and limitations found through this study.
- Chapter 6 is the last chapter, presenting recommendations for further work with this method in order to develop and evaluate its abilities even further.

Chapter 2

Theory

In this chapter theory concerning resistance of catamarans, established empirical resistance prediction methods and machine learning, with emphasis on artificial neural networks, will be presented. Some of the material is based on the literature study carried out by the author in the project report last fall, but the text has been modified to fit this thesis. Theory of neural network training has been extended, pitfalls have been added and the workflow of designing neural networks in Matlab has been added as well.

2.1 Resistance

The different resistance components for any object travelling through water can be broken down into forces acting on the object. The consensus of calculating vessel hull resistance today is based on Froude scaling, which is represented with dimensionless coefficients as in Equation 2.1. And the total resistance coefficient for a ship is calculated as shown in Equation 2.2 [Fathi et al., 2012, page 56].

$$C_X = \frac{R_X}{\frac{1}{2} \cdot \rho \cdot S \cdot V^2} \tag{2.1}$$

$$C_T = C_R + (1 + k_0)(C_F + \Delta C_F) + C_{AA} + C_{BD} + C_A \tag{2.2}$$

C_X: resistance coefficient of force R_X.

R_X: resistance force X.

ρ: fluid density.

S: surface the force is acting on (wetted surface for forces from water, projected area for air forces).

V: relative velocity between object and fluid.

C_T: total resistance coefficient.

C_R: residuary resistance coefficient.

$(1 + k_0)$: form factor.

C_F: frictional resistance coefficient.

ΔC_F: hull roughness allowance.

C_{AA}: air resistance coefficient.

C_{BD}: base drag coefficient.

C_A: resistance correlation coefficient (correlating the result against full-scale trials).

An illustration of the different resistance components can be seen in Figure 2.1

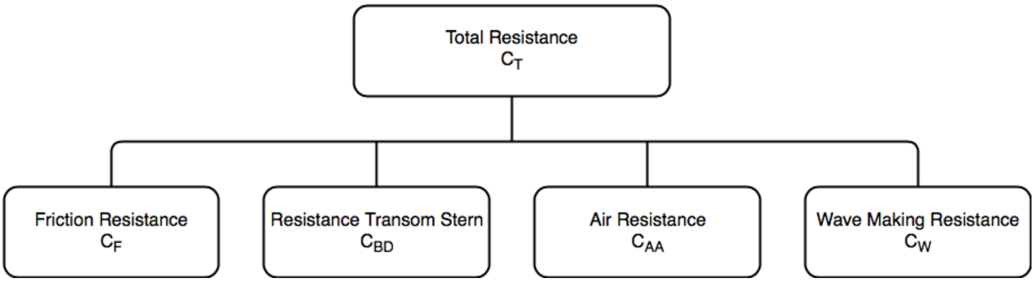

Figure 2.1: Illustration of catamaran resistance components.

The total resistance coefficient for a model scale catamaran is expressed in Equation 2.3 [Steen and Minsaas, 2013, equation 9.1]. The interaction between hulls is an important factor for catamarans, which is taken into account by the factors: γ and β.

$$C_{Tm} = \gamma \cdot C_W(F_N) + C_{Fm}(1 + k \cdot \beta) + C_{AAm} + C_{BDm} \tag{2.3}$$

C_{Tm}: total resistance coefficient for the model.

$\gamma \cdot C_W(F_N) = C_R$: residuary resistance coefficient.

γ: correction of wave resistance due to the presence of the other hull.

k: form factor without the presence of another hull.

β: correction of form factor due to the presence of other hulls.

C_{Fm}: frictional coefficient for the model.

C_{AAm}: air resistance coefficient for the model.

C_{BDm}: transom stern resistance coefficient for the model.

The residuary resistance coefficient, presented in Equation 2.2 and 2.3, is the difference between the measured total resistance and the calculated viscous resistance for a ship. It is assumed to be identical for full scale and model scale vessel at the same Froude number, and therefore it can be found by doing model trials of desired designs at desired speeds. The coefficient can be found as stated in Equation 2.4.

$$C_R = C_T - C_{Viscous} = C_{Tm} - \left[C_{Fm}(1 + k \cdot \beta) + C_{AAm} + C_{BDm}\right] \tag{2.4}$$

2.1.1 Frictional Resistance

The frictional force is the sum of all tangential shear forces acting on the hull surface due to the hull moving through water. ITTC [1987] recommends estimating the friction resistance using the ITTC'57 friction correlation line which can be expressed by Equation 2.5.

$$C_F = \frac{0.075}{(log(Rn) - 2)^2} \tag{2.5}$$

Equation 2.5 is valid for smooth surfaces, and because a ship hull has roughness an additional friction coefficient must be added. Roughness will for practical applications be most significant at high-speeds, and because catamarans typically operate at high speeds, this must be accounted for. An empirical expression for calculating roughness allowance is presented in Equation 2.6 [Steen, 2014, section 1.2.2].

$$\Delta C_F = \left[110 \cdot (H \cdot V)^{0.21} - 403\right] \cdot C_F^2 \tag{2.6}$$

H: roughness of the hull (typically 50-150 μm for newly docked vessels).

Equation 2.5 is describing the frictional resistance coefficient-model-ship correlation line, so in order to describe the friction coefficient for a specific hull it needs to be multiplied with a form factor to find the correct viscous resistance component. The hull of a model is assumed to be "smooth" and therefore the roughness allowance, presented in Equation 2.6, is neglected.

$$C_{VM} = (1 + k) \cdot C_F \tag{2.7}$$

To make this expression for wave making resistance, and residuary resistance, valid for a multi-hulled vessel, a parameter β is included to correct the form factor for the presence of other hulls [Insel and Molland, 1992].

$$C_{VM} = (1 + k \cdot \beta) \cdot C_F \tag{2.8}$$

2.1.2 Determining the Form Factor k

The form factor k in Equations 2.3, 2.7 and 2.8 can be found using different methods. In the following sub-sections the Prohaska's method for high Froude numbers and an empirical method are presented.

Prohaska's Method for High-Speed Towing Tests

In the Prohaska's method for low-speed monohulls, the model is towed at low speed and the form factor is found by linear regression of the resistance data [Steen, 2014, chapter 1.2.1]. However, it has been shown that the wave resistance reaches its maximum at a Froude number of $F_N = 0.5$, while most catamarans operate at $F_N > 0.5$. By increasing F_N further the wave resistance decreases, and at very high F_N the total resistance is due to dynamic lift, frictional resistance and air resistance only.

The form factor can be found by the original Prohaska's method, but there is a major problem regarding low speed towing test due to the transom stern resistance - which will be discussed later. The resistance due to wetted stern at low speeds will be dominating at $F_N < 0.2$ and must therefore be subtracted if the method is used. As an alternative, the form factor can be found by doing high-speed towing tests, where the wave resistance becomes small. The Prohaska' method can be applied to high-speeds by plotting:

$$\frac{C_{Tm} - C_{AA}}{C_{Fm}} = (1 + k) + m \cdot \frac{C_W(F_N)}{C_{Fm}} = (1 + k) + m \cdot \frac{F_N^n}{C_{Fm}} \tag{2.9}$$

Then n can be changed until a linear relation is obtained between $\frac{C_{Tm}-C_{AA}}{C_{Fm}}$ and $\frac{F_N^n}{C_{Fm}}$. The form factor k is given by the intersection with the y-axis, and now the wave or pressure resistance is expressed as:

$$C_W = m \cdot F_N^n \tag{2.10}$$

m: a constant in which gives a linear behaviour of $\frac{F_N^n}{C_{Fm}}$.

The linear tendency is normally obtained when $4 < n < 6$ for $F_N < 0.5$, while $n = -1.25$ for $F_N > 1.0$. An example of this high F_N Prohaska's method can be seen in Figure 2.2.

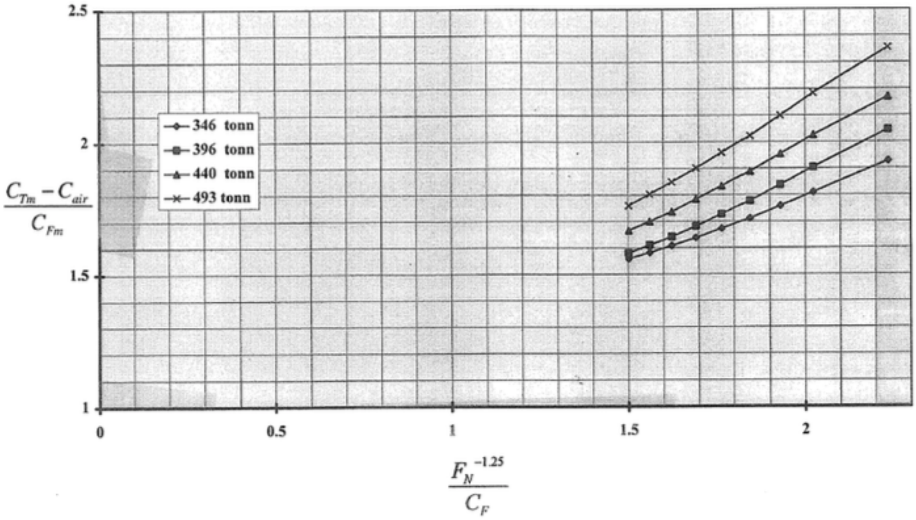

Figure 2.2: Form factor of a high-speed catamaran determined from high-speed towing tests. [Steen and Minsaas, 2013, figure 9.1].

Empirical Methods

The form factor k can also be found through empirical methods, which are based on model test and curve fit to data. SINTEF Ocean are using the formula presented in Equation 2.11. This expression is based on experimental data for conventional displacement hulls, together with some analytically based expressions [Fathi et al., 2012, Chapter 4.1]. The model is fit to give slightly lower form factor than the experimental data, making it an empirical model which to a large extent excludes the viscous pressure resistance from the form factor. Therefore, the model presented in Equations 2.11 and 2.12 will normally give a lower form factor than obtained from experiments.

$$k = 0.6\varphi + 145\varphi^{3.5} \tag{2.11}$$

$$\varphi = \frac{C_B}{L_{WL}}\sqrt{(T_{AP} + T_{FP}) \cdot B} \tag{2.12}$$

C_B: block coefficient.

L_{WL}: length of waterline.

T_{AP}: draught at aft perpendicular.

T_{FP}: draught at front perpendicular.

B: breadth.

2.1.3 Resistance due to a Transom Stern

For high-speeds, which is the typical operating condition for high-speed-catamarans, the water behind the stern separates along the edge between the transom stern and the hull in such a way that the entire transom is dry. In this case, the hydrostatic pressure is lost and the hull is experiencing a force due to the lower pressure behind the hull. This may be a significant contribution for high-speed catamarans.

For a completely dry stern this loss of hydrostatic force will be felt as a resistance, and can be expressed as:

$$R_{Transom} = \rho g \int_0^{H_D} z \cdot B(z) dz \tag{2.13}$$

g: gravitational acceleration.

H_D: maximum depth of transom, which varies with the trim angle and sinkage.

$B(z)$: width of transom at different draughts.

When the stern is not entirely dry a base drag is introduced. This force is due to still water behind the stern being dragged behind the vessel when it travels at low speeds. SINTEF Ocean have found an empirical expression for estimating the base drag, which is presented in Equation 2.14 [Fathi et al., 2012, Chapter 4.1].

$$C_{BD} = \frac{R_{Transom}}{\frac{1}{2}\rho S V^2} = C_{Transom} \approx 0.029 \cdot \frac{(\frac{S_B}{S})^{3/2}}{\sqrt{C_F}} \tag{2.14}$$

S_B: transom stern area.

S: the wetted surface of the hull in front of the transom.

The transom becomes dry approximately when $F_N > 0.2$, however, this value is depending strongly on the depth of the transom. A Froude number for the transom is a better way of defining the limit of dry transom [Robards and Doctors, 2003]:

$$F_{NT} = \frac{V}{\sqrt{g \cdot H_D}} \tag{2.15}$$

Robards and Doctors [2003] showed through experiments, that the transom is dry when $F_{NT} > 2.5$.

2.1.4 Air Resistance

Air resistance is highly dependent on the relative speed between the vessel and the wind or still air in the area where it travels. This resistance component can be significantly high for high-speed catamarans due to the second order dependence on relative speed, seen in Equation 2.16. The air resistance coefficient is presented in Equation 2.17.

$$R_{AA} = \frac{1}{2} \cdot V_0^2 \cdot A_P \cdot C_{air} \tag{2.16}$$

$$C_{AA} = \frac{R_{AA}}{\frac{1}{2}\rho S V_0^2} = \frac{\rho_{Air}}{\rho} \cdot C_{Air} \cdot \frac{A_P}{S} \tag{2.17}$$

A_P: projected area above the waterline, normal to the direction of motion.

V_0: relative velocity between ship and air.

ρ_{air}: density of air.

C_{Air}: air resistance coefficient for the superstructure of the vessel.

Figure 3.29 in Molland et al. [2011] shows that the air resistance coefficient for fast catamaran ferries normally is in the range of $0.50 < C_{Air} < 0.88$.

2.1.5 Resistance Correlation Coefficient

In order to correlate the model scale results against full-scale trials, a correlation coefficient is introduced. This usually varies for different towing facilities. The procedure at SINTEF Ocean towing facility is to apply $C_A = -0.228 \cdot 10^{-3}$ for single-screw ships, and $-0.2 \cdot 10^{-3} \leq C_A \leq -0.23 \cdot 10^{-3}$ for twin-screw ships [Fathi et al., 2012, page 59].

2.2 Empirical Resistance Prediction Methods

Over the years several empirical models for estimating resistance on displacement catamarans have been developed. Following is a presentation of established methods, such as the CatRES method developed by SINTEF Ocean (MARINTEK) and other acknowledged resistance prediction schemes.

2.2.1 CatRES Method

CatRES method is an empirical method based on data from models tested in the MARINTEK towing facility from 1990 to 1998. The models are mostly of passenger catamarans with speeds of 30 to 40 knots and lengths of approximately 40 meters in full scale. In this model, the main parameter is the length-displacement ratio (or slenderness parameter): $L/\nabla^{1/3}$. The total resistance is calculated through Equation 2.18, where the total resistance coefficient C_{Ts} are dependent on the applied method. All methods presented below have resistance coefficients which are calculated for five discrete values of the Froude number; Fn=0.6, 0.7, 0.8, 0.9, 1.0. For each of these Froude numbers, the resistance coefficient is calculated from an empirical expression. If the method is applied to a design where the Froude number is not exactly the same as the Fn indicated above, interpolation between resistance coefficients for the two adjacent Froude numbers must be carried out.

In the following subsections, the different methods for calculating C_{Ts} are presented. All expressions in this section are taken from the ShipX manual where CatRES is a plug-in [Fathi et al., 2012, Chapter 4.10]. This model is known for producing conservative resistance calculations as the model data, which the resistance model is based upon, are of older designs. The resistance model will therefore be less accurate when used on newer designs.

$$R_{Ts} = C_{Ts} \cdot \frac{\rho_s}{2} \cdot V_s^2 \cdot S_s \tag{2.18}$$

Method 1: Correlation on C_{Ts}

In this method the total resistance coefficient can be expressed as:

$$C_{Ts} = C_{Ts}^{\text{emp}} + C_{AAs} \tag{2.19}$$

Where C_{AAs} is the air resistance coefficient, and C_{Ts}^{emp} is found from the fifth order polynomial:

$$C_{Ts}^{\text{emp}} = C_{Ts0} + C_{Ts1}\Big(\frac{L}{\nabla^{1/3}}\Big) + C_{Ts2}\Big(\frac{L}{\nabla^{1/3}}\Big)^2 + C_{Ts3}\Big(\frac{L}{\nabla^{1/3}}\Big)^3 + C_{Ts4}\Big(\frac{L}{\nabla^{1/3}}\Big)^4 + C_{Ts5}\Big(\frac{L}{\nabla^{1/3}}\Big)^5 \tag{2.20}$$

The coefficients C_{Ts0} to C_{Ts5} are Froudes number dependent.

Method 2: Correlation on C_R

In this method the total resistance coefficient can be expressed as:

$$C_{Ts} = C_R^{\text{emp}} + (C_{Fs} + \Delta C_F) + C_A + C_{AAs} \tag{2.21}$$

Where C_R^{emp} is found from the fifth order polynomial:

$$C_R^{\text{emp}} = C_{R0} + C_{R1}\Big(\frac{L}{\nabla^{1/3}}\Big) + C_{R2}\Big(\frac{L}{\nabla^{1/3}}\Big)^2 + C_{R3}\Big(\frac{L}{\nabla^{1/3}}\Big)^3 + C_{R4}\Big(\frac{L}{\nabla^{1/3}}\Big)^4 + C_{R5}\Big(\frac{L}{\nabla^{1/3}}\Big)^5 \tag{2.22}$$

C_{R0} to C_{R5} are Froudes number dependent coefficients.

∇: volume of displacement.

s: distance between hulls.

Method 3: Correlation on C_R With Correction

In this method the effect of draught, beam and wetted area is included in the calculation of the total resistance coefficient. Equation 2.21 is corrected to include the mentioned parameters:

$$C_{Ts} = C_R^{\text{emp}}\Big(k_1 + k_2\frac{S}{\nabla^{2/3}}\Big) \cdot \Big(k_3 + k_4\frac{B_{\text{demi}}}{T}\Big) \cdot \Big(k_5 + k_6\frac{s}{L}\Big) + (C_{Fs} + \Delta C_F) + C_A + C_{AAs} \tag{2.23}$$

k_1 to k_6 are empirically calculated coefficients, which are Froudes number independent.

B_{demi}: beam of the demi hulls.

C_R^{emp}: from Equation 2.22.

Method 4: Correlation on C_W

In this method, the correlation between the residuary resistance with viscous pressure resistance subtracted, C_W, is done. Here the viscous pressure resistance is expressed by the form factor k, which is found through a similar way as presented in Section 2.1.2. the factor m in Equation 2.9 is assumed to be equal to 1, which yields:

$$\frac{C_{Tm} - C_{AAm}}{C_{Fm}} = (1 + k) + \frac{Fn^{-n}}{C_{Fm}} \tag{2.24}$$

n is found through least-squares-fit of the experimental data in order to find a linear relation in Equation 2.24, and then k can be found. The total resistance coefficient can then be expressed as:

$$C_{Ts} = C_W^{\text{emp}} + (C_{Fs} + \Delta C_F) \cdot (1 + k) + C_{AAs} \tag{2.25}$$

Where C_W^{emp} is the Froudes number dependent, empirically calculated wave resistance coefficient found from:

$$C_W^{\text{emp}} = C_{W0} + C_{W1}\big(\frac{L}{\nabla^{1/3}}\big) + C_{W2}\big(\frac{L}{\nabla^{1/3}}\big)^2 + C_{W3}\big(\frac{L}{\nabla^{1/3}}\big)^3 + C_{W4}\big(\frac{L}{\nabla^{1/3}}\big)^4 + C_{W5}\big(\frac{L}{\nabla^{1/3}}\big)^5 \tag{2.26}$$

Again, the coefficients: C_{W0} to C_{W5} are Froudes number dependent coefficients.

Method 5: Correlation on C_W With Correction

Again the effect of draught, beam and wetted area are taken into account, which corrects Equation 2.25 into:

$$C_{Ts} = C_W^{\text{emp}}\big(k_1 + k_2\frac{S}{\nabla^{2/3}}\big) \cdot \big(k_3 + k_4\frac{B_{\text{demi}}}{T}\big) \cdot \big(k_5 + k_6\frac{s}{L}\big) + (C_{Fs} + \Delta C_F) \cdot (1 + k) + C_{AAs} \tag{2.27}$$

Coefficients k_1 to k_6 are Froudes number independent, and empirically calculated. They are however, not equal to the ones presented in method 2. C_W^{emp} is calculated using Equation 2.26.

2.2.2 Molland et al. Method (1994)

Molland et al. [1995] continued the work done by Insel and Molland [1992], by including resistance model data for ten more models into their original database of four models. The authors proposed that the total resistance of a catamaran could be expressed as:

$$C_{\text{T CAT}} = (1 + \beta k)C_F + \tau C_W \tag{2.28}$$

Where C_W can be calculated by the equation:

$$\begin{aligned} C_W = b_0 &+ b_1\frac{L}{B} + b_2\frac{B}{T} + b_3\frac{L}{\nabla^{1/3}} + b_4\frac{s}{L} + b_5\frac{L}{B}\frac{B}{T} + b_6\frac{L}{B}\frac{L}{\nabla^{1/3}} + b_7\frac{B}{T}\frac{L}{\nabla^{1/3}} + b_8\frac{L}{B}\frac{s}{L} + b_9\frac{B}{T}\frac{s}{L} \\ &+ b_{10}\frac{L}{\nabla^{1/3}}\frac{s}{L} + b_{11}\frac{L}{B}\frac{B}{T}\frac{L}{\nabla^{1/3}} + b_{12}\frac{L}{B}\frac{B}{T}\frac{s}{L} + b13\frac{L}{B}\frac{L}{\nabla^{1/3}}\frac{s}{L} + b_{14}\frac{B}{T}\frac{L}{\nabla^{1/3}}\frac{s}{L} + b_{15}\frac{L}{B}\frac{B}{T}\frac{L}{\nabla^{1/3}}\frac{s}{L} \end{aligned} \tag{2.29}$$

Where the coefficients: b_0 to b_{15} are found from regression [Sahoo et al., 2007]. The form factor in Equation 2.28 is also calculated through an empirical expression which is dependent on several hull form characteristics, as presented in Equation 2.30. Figure 2.3

shows how the method predicts C_W for round bilge catamarans.

$$(1+\beta k) = f\big(\frac{B}{T}; \frac{L}{\nabla^{1/3}}; \frac{s}{L}; (1+k)\big) \tag{2.30}$$

T: characteristic draught.

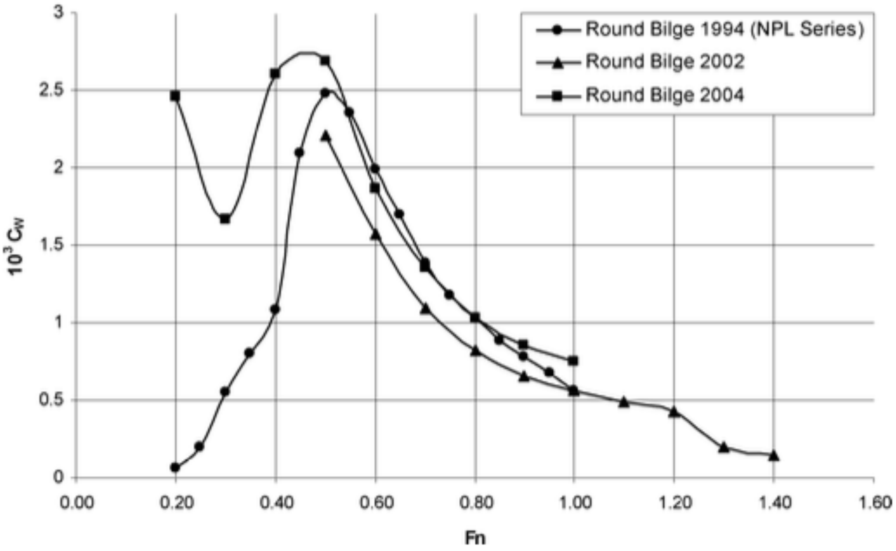

Figure 2.3: Predicted wave resistance coefficients for round bilge catamarans using Molland et al. [1995] method, [Sahoo et al., 2007, Figure 7].

2.2.3 Other Empirical Methods

There are more empirical methods for resistance estimations of catamarans than the two presented earlier in this section. Sahoo et al. [2007] presents several other methods in their paper, which is an overview of empirical, and semi-empirical methods for predicting catamaran resistance. Two of the presented methods by Xuan et al. [2001], and Subramanian and Joy [2004] are based on the software SHIPFLOW. SHIPFLOW divides the domain around the object into three areas which can be solved by either: potential flow method, boundary layer method or Navier-Stokes method; which makes it a combined computational fluid dynamics and 3D potential theory program [Sahoo et al., 2007]. By doing regression analysis on the obtained results from SHIPFLOW, Subramanian and Joy [2004], and Xuan et al. [2001] have developed their empirical models. The methods are not available online and are therefore difficult to evaluate, but according to Sahoo et al. [2007], the results are accurate.

2.3 Regression Models

Regression is based on developing models to predict future outcomes by finding logical patterns in known data. This could be looking at resistance data from catamaran model tests, wanting to predict the resistance for a new design. In order to find a model, which describes the relationship between desired parameters in a satisfactory manner, different concepts of regression models can be utilised. Regression models are often divided into three groups: white, grey and black box models. Where a white box model is an analytic mathematical model describing the physical problem, a black box model is when there is no physical relationship between the phenomena and the mathematical model describing the phenomena, and a grey box model has empirical coefficients combined with expressions based on physical considerations.

When describing the resistance of a fast displacement catamaran, there is no obvious linear relationship between resistance and a given variable. The problem is complex, being dependent on parameters such as hull spacing, length/beam rations, hull separation, velocity, stern characteristics, wetted area ratios and trim. When the physical phenomena in question are depending on a significant number of parameters, which are difficult to relate by using physical relationships, black box models are more favourable. An ANN algorithm is free to find patterns unrelated to physical parameters, which makes it able to find more accurate resistance prediction models. Artificial neural networks and deep learning algorithms will, however, demand large amounts of training data and computer power, in order to construct an algorithm which predicts resistance accurately.

In the following sections, two different types of regression models are presented; models based on polynomial curve fitting and machine learning.

2.3.1 Models Based on Polynomial Curve Fitting

Polynomial curve fitting is based on fitting a n$^{\mathrm{tn}}$ order polynomial to a given data set. The order of the polynomial is varied in order to obtain the lowest possible error between proposed model and actual data. This error can be expressed as the mean squared error (MSE), which can be seen in Equation 2.31.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2 \tag{2.31}$$

N: number of points.

$\hat{Y}_i$: predicted target value.

Y_i: target value.

Figure 2.4, shows a data set and four proposed polynomials to describe the data. Table 2.1 shows the mean squared error between the sample and the polynomials, as presented in Equation 2.31, indicating which one are better suited to describe the sample.

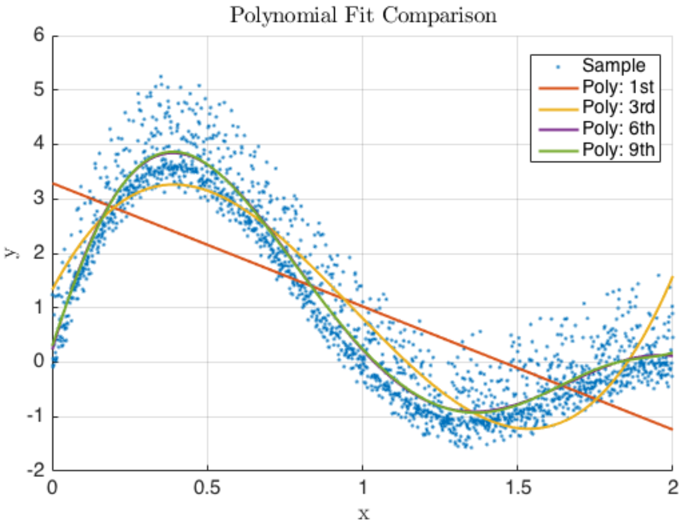

Figure 2.4: Polynomials of various order fitted to an example data sample. MSE presented in Table 2.1.

Table 2.1: Mean squared error for polynomial fit to sample data.

Polynomial	MSE
Poly 1^{st}	1.3134
Poly 3^{rd}	0.4566
Poly 6^{th}	0.2249
Poly 9^{th}	0.2246

By looking at the polynomials describing the sample in Figure 2.4, and the characteristic values for mean squared errors, one can see that the higher order polynomials are best describing the sample in this example.

2.3.2 Polynomial Curve Fitting used in Empirical Resistance Models for Catamarans

As presented in Section 2.2, empirical models for predicting resistance of fast displacement catamarans are commonly based on polynomial curve fitting to data from model trials. The CatRES method presented in Section 2.2.1, is a fifth order polynomial fit, based on the variable $\frac{L}{\nabla^{1/3}}$. In the project thesis by Hans Jørgen Rambech [1998] the original polynomial fitted resistance model was a fourth order polynomial, which can be seen in Figure 2.5. The fourth order polynomial seems to describe the total resistance ($C_{\mathrm{TS}} - C_{\mathrm{AAS}}$) in a satisfying way.

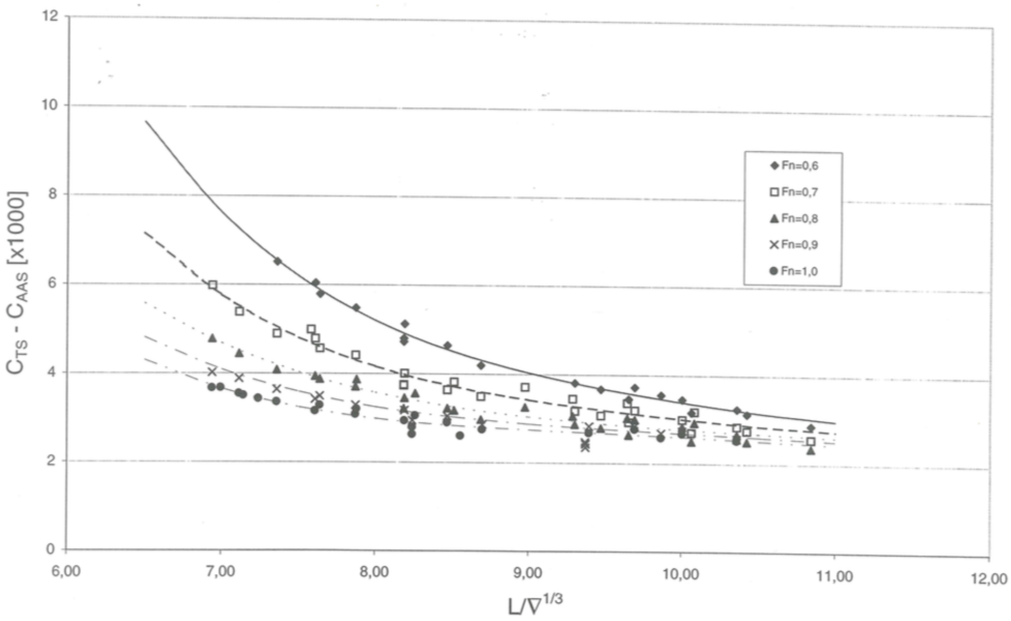

Figure 2.5: Regression on $C_{TS} - C_{AAS}[x1000]$ with fourth order polynomials from [Rambech, 1998, figure 6.2].

Other empirical resistance prediction models are based on other types of polynomials. The model proposed by Molland et al. [1995], presented in Section 2.2.2, was based on the combination of the parameters: $(\frac{L}{B})$, $(\frac{B}{T})$, $(\frac{L}{\nabla^{1/3}})$ and $(\frac{s}{L})$.

When using polynomial curve fitting the relationships between output: resistance, and input: dimensions, speed or other parameters are important. Figure 2.5, indicates a smooth descending relationship between the parameter $\frac{L}{\nabla^{1/3}}$ and the resistance, for a constant Froude number. But this method does not show how other dimensional changes such as; transom area, propulsion system or hull separation will affect the resistance. As mentioned earlier, one of the advantages of artificial neural networks or deep learning algorithms is that they can take in large amounts of inputs and find out how they affect the output.

2.3.3 Machine Learning

Machine learning is defined as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty [Murphy, 2012, chapter 1.1]. By using probability theory the machine learning methods are detecting which methods will predict the most accurate results, based on some past data, and how the tested models are performing in relation to each other. The difference from other disciplines within computer science is that the programs are designed to ”learn” to perform a task without being explicitly programmed to perform that task. Machine learning is based on the ”learning by doing” principle, where the program is given large amounts of training data and is let to find the best possible algorithm to describe the phenomena. There are mainly two different learning approaches in machine learning:

> **Supervised learning**: is when some input data and the related output data is fed into the algorithm. The algorithm will then train the program to generate output data which is consistent to the given output data.
>
> **Unsupervised learning**: is when there is no output data provided. The algorithm will then try to discover hidden structures in the data.

Supervised learning is the most common approach in problems concerning classification and regression. This is because the desired output from a certain combination of inputs is known for the training set. Unsupervised learning is, on the other hand, better for clustering, when data such as strings, images and sounds should be grouped. Unsupervised learning does not require labelled data for training, which limits the resources required for pre-processing data.

When creating and training a machine learning algorithm, the goal is to make a function f which accurately predicts an output y from a set of input parameters $\mathbf{x}$. Mathematically this can be described as in Equation 2.32.

$$y = f(\mathbf{x}) \tag{2.32}$$

When training the algorithm using supervised learning, y and $\mathbf{x}$ are known, and an optimal function f is desired.

2.4 Artificial Neural Networks

Artificial neural networks are a famous sub-group of machine learning. Inspired by an animal brain, the method tries to emulate how the neurons are communicating in order to solve problems. As in a biological brain, the network will consist of a large number of neurons interconnected in a great network. When the network is given a sensor input, the first layer in the network ”fires”, triggering actions in the next connected layers with neurons, which ultimately gives an output. Next, the main building blocks in an ANN are presented.

2.4.1 The Neuron in Artificial Neural Networks

The main building block in an artificial neural network is the neuron. A simple model of an input-neuron can be seen in Figure 2.6.

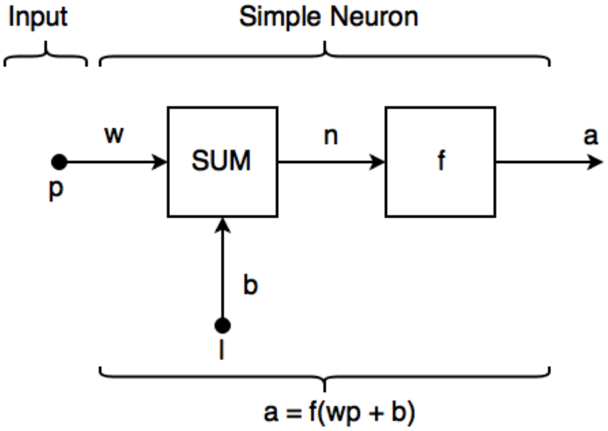

Figure 2.6: Graphical and mathematical formulation of a simple neuron [Beale et al., 2017].

The neuron in Figure 2.6 is receiving one or more inputs p, which is multiplied with weights w (producing the product wp). Then a bias b is added to the product in the SUM-function. Next the input, weight and bias are put through the transfer function f, creating the output a. The process is described in Equation 2.33.

$$a = f(wp + b) \tag{2.33}$$

In this neuron, the parameters w and b needs to be assigned characteristic values. This is the basis of the training process for the neural network, which will be described later.

The transfer function, denoted as f in Equation 2.33, needs to be chosen for the neuron. A simple transfer function is the linear transfer function, presented in Equation 2.34.

$$f(x) = x \tag{2.34}$$

This will give an infinite output range, and as the output a often is desired to be within a unit range: $-1 \leq a \leq 1$, this transfer function is not optimal. Other functions are therefore used, such as the hyperbolic tangent function: $tanh(x)$, and different types of Sigmoid functions. These functions are also differentiable, which is an important feature for the transfer functions [Nielsen, 2015]. In the next section, the assembly of multiple neurons into one network is presented and how these layers can be put together to form an advanced structure.

2.4.2 Neurons Combined in Layers Making Advanced Structures

By combining multiple neurons as shown in Figure 2.6, a structure called a layer of neurons can be constructed. By using multiple neurons, a layer with n neurons can be constructed having different biases (b) and weigh parameters (w). During training, these biases and weigh functions are found, so the network gives the desired outputs. By expanding the structure using multiple layers, an advanced structure can be created. If one layer, as presented in Figure 2.7, is drawn as a circle - the total architecture of a relatively simple neural network can be visualized as in Figure 2.8. When designing neural networks, as the one presented in Figure 2.8, it is desirable to design the structure for a specific task. For example, the number of input layers in Figure 2.8 should be equal to the number of input parameters. Then one layer will describe one input variable. The number of interior layers or hidden layers should be varied in order to find which structure is showing the best ability to give correct outputs. And at last, the number of output layers should be equal to the number of possible outcomes.

2.4.3 Performance of an Artificial Neural Network

As mentioned in the last section, the mean squared error is often used as a performance indicator often called cost function, for the network [Beale et al., 2017, Page 3-17]. The MSE

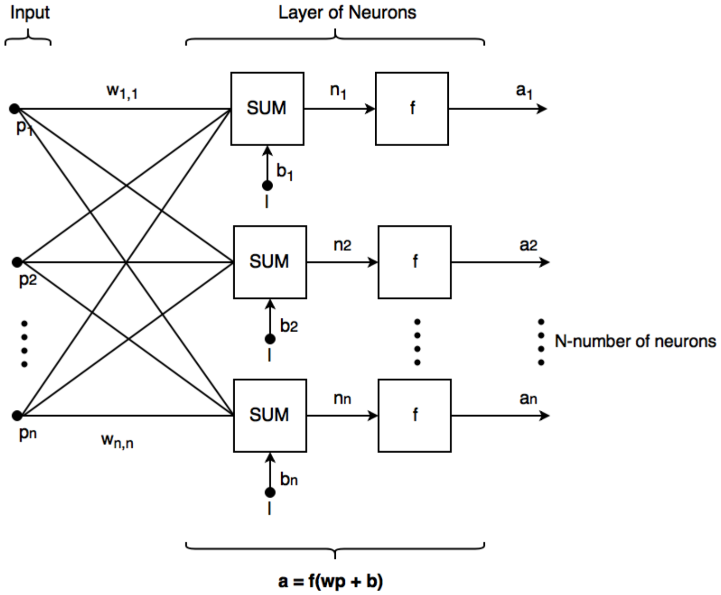

Figure 2.7: Graphical and mathematical formulation of a layer made out of neurons [Beale et al., 2017].

is calculated in the same way as presented in Equation 2.31, in Section 2.3.1. Having a low MSE is desirable for an algorithm, as this indicates a high level of consistency between the actual value and the predicted value. The MSE is also used as the benchmark-parameter for ANNs in Matlab [Beale et al., 2017, Page 4-46].

When training a neural network, the training and validation MSE are important performance parameters. When the network is trained, training MSE decreases as the network learns to reproduce the correct output from the given input. Next, the steps in training a neural network are presented, while concepts of over- and under training are discussed later in this paper.

2.4.4 Training a Neural Network

When training neural networks or a machine learning algorithm in general, the objective is to find an algorithm which fits the training data in a best possible way. Training means getting input parameters and systematically changing the biases and weight parameters in each neuron in order for the final output to coincide with the results from the training data. When the neural network structure is complex, containing several layers, possibly in parallel, the number of biases and weight parameters to train can be significantly high.

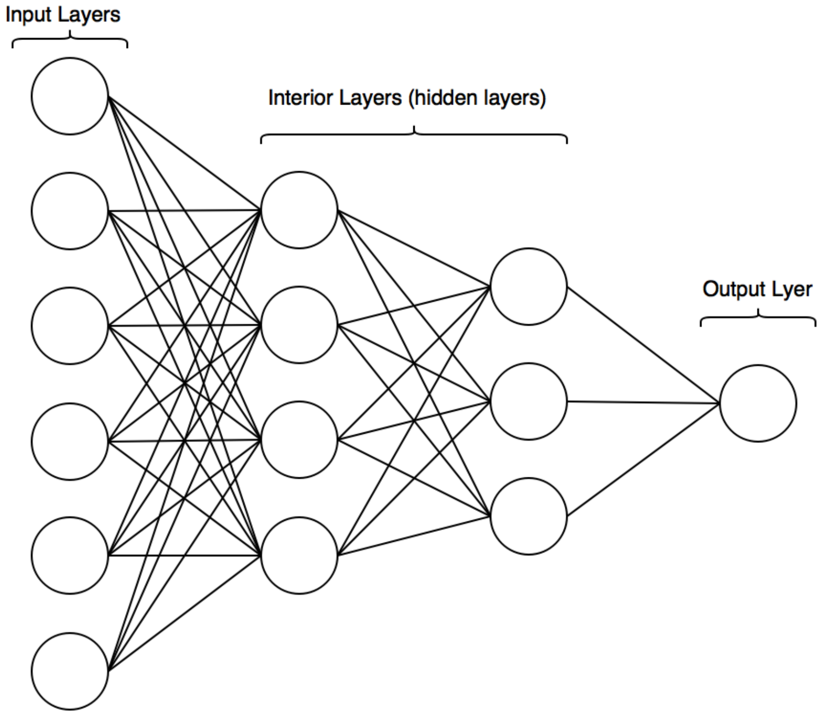

Figure 2.8: Visualisation of an artificial neural network structure. Multiple layers (circles) are put together.

In order to train all parameters, significant amounts of training data and time are also required. The general procedure of training a machine learning algorithm is presented in Figure 2.9 [Abu-Mostafa et al., 2012, Figure 1.2]. By assuming there is an optimal algorithm f which is perfectly describing the observed phenomena, the training is trying to find this algorithm. The training is, therefore, a large number of hypothesis tests where the algorithm decides which one of the proposed candidate formulas in hypothesis set H, which most accurately describes the phenomena. In order to check this, the learning algorithm tries the proposed model on the training examples and compares the output to the given results. In the end, an optimal model g is proposed as the best-fit formula for describing the data.

A common way of training and validation of a neural network is to divide the available dataset into three groups: a training set, a validation set and a test set. The training set is typically the largest data set, being the data which is used in training the algorithm. After one round of training with the training set, called an epoch, the algorithm is tested using the test data. So the MSE for both training and testing is calculated for each epoch. An

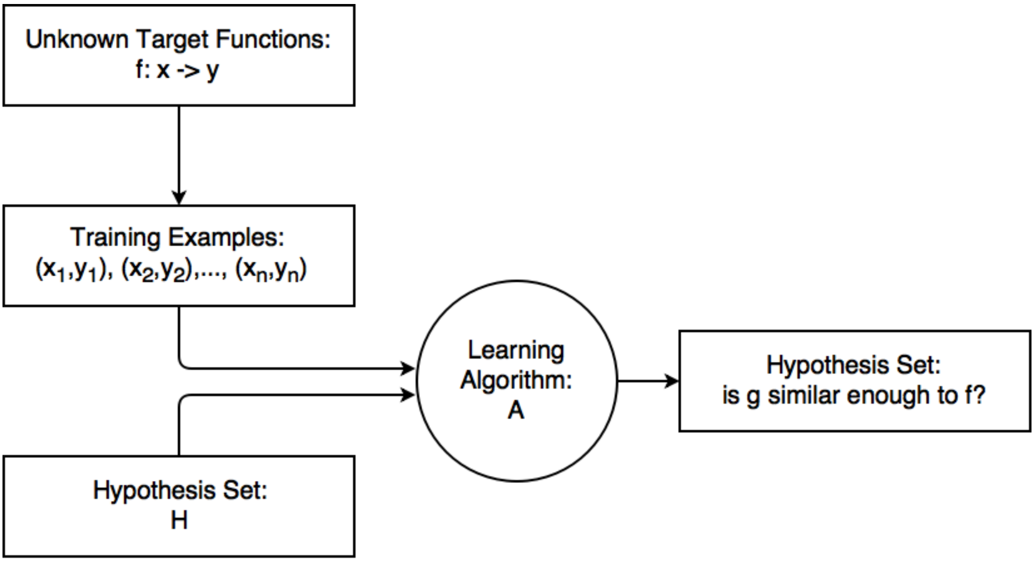
Unknown Target Functions:
f: x -> y
Training Examples:
$(x_1,y_1), (x_2,y_2),..., (x_n,y_n)$
Learning Algorithm:
A
Hypothesis Set:
is g similar enough to f?
Hypothesis Set:
H

Figure 2.9: Basic steps of training a machine learning algorithm [Abu-Mostafa et al., 2012, Figure 1.2].

algorithm can show fantastic performance with low training MSE, but poor performance with high test MSE. This is called overtraining and will be presented in the next section.

Training a machine learning algorithm can be highly timeconsuming, so one of the main focus area for researchers in artificial intelligence developing training schemes. In the podcast *Talking Machines*, the expert on intelligent systems Ryan Adams mentions backwards propagation as a training method superior to other training schemes [Adams and Gorman, 2015].

Early Stopping, Regularisation and Backpropagation

When training neural networks one strives to find the best-suited network, trained for an optimal number of epochs. Three central topics in training neural network are early stopping, regularisation and backpropagation, which will be presented below.

Early Stopping

Early stopping is a way to avoid over-trained networks, by controlling the effective complexity of the network. The training of non-linear network models corresponds to an iterative reduction of the error function, for example, the mean squared error. For many optimisation algorithms, the error is a non-increasing function of the iteration index [Bishop, 2006]. The blue training performance curve in Figure 2.11, is such a non-increasing function. However, the error measured with respect to independent data, is indeed not non-increasing. It often decreases at first, but increases as the networks start to over-train. This can be seen for the red test performance curve in Figure 2.11. Both Bishop [2006] and Tzafestas et al. [1996] recommends using early stopping which is the principle of breaking of the training at the point where the error of the test data set is at its lowest.

Regularisation

Regularisation is another way of preventing over-fitting. The number of input- and output parameters to and from a neural network is generally determined by the available data set and its dimensions. The number of hidden layers and neurons is, on the other hand, a free parameter. When applying regularisation in training, a penalty term is introduced to the performance function (MSE). The penalty increases together with the complexity. The simplest *reguliser* is the quadratic one, known as the *weight decay*, shown in Equation 2.35 [Bishop, 2006].

$$\widetilde{E}(W) = E(W) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \tag{2.35}$$

The effective model complexity is then determined by the regularised performance function ($\widetilde{E}(W)$), and the regularisation coefficient λ [Bishop, 2006]. The performance of the network will increase (poorer performance) for networks with large weight vectors $\mathbf{w}$.

Another regularisation scheme is **Bayesian regularisation backpropagation**. In this method the performance or loss function is determined as in Equation 2.36, and after the loss function is introduced the resulting loss function is as in Equation 2.37 [Foresee and Hagan, 1997]. Where the term E_W is the sum of all network weights shown in Equation 2.38.

$$E_D = \sum_{i=1}^{n}(t_i - a_i)^2 \tag{2.36}$$

$$\widetilde{E} = \beta E_D + \alpha E_W \tag{2.37}$$

$$E_W = \sum_{i,j} ||w_{ij}||^2 \tag{2.38}$$

The key to optimal training is finding the best regularisation parameters β and α. MacKay [1992] presents this situation as multiple ”Occam's razor” problems. An ”Occam's razor” problem-solving principle is such that when presented with competing answers to a problem, one should select the answer based on fewest assumptions. In science, the principle is used as a heuristic guide [Gauch, 2003]. The parameters β and α are known as hyperparameters which are affecting the loss function. MacKay [1992] has shown that the error decreases when $\alpha << \beta$. If $\alpha >> \beta$ the training algorithm will emphasise weight reduction at the expense of network errors, which produce smoother network response [Foresee and Hagan, 1997].

After each training epoch, the posterior distribution of the weights in the neural network can be updated according to Bayes' rule, as in Equation 2.39.

$$P(w|D,\alpha,\beta,M) = \frac{P(D|w,\beta,M)\cdot P(w|\alpha,M)}{P(D|\alpha,\beta,M)} \rightarrow Posterior = \frac{Likelihood \cdot Prior}{Evidence} \tag{2.39}$$

w: weights.

D: training set with input-target pairs.

M: characteristic neural network architecture.

In Equation 2.39 the prior distribution of weights is defined as in Equation 2.40.

$$P(w|\alpha,M) = (\frac{\alpha}{2\pi})^{m/2} \cdot exp(-\frac{\alpha}{2}w'w) \tag{2.40}$$

In this Bayesian framework the optimal weights should be found by maximising the posterior probability: $P(w|D,\alpha,\beta,M)$ of w. This is equal to minimizing the regularized loss function in Equation 2.37. According to MacKay [1992] one can write the joint posterior density as in Equation 2.41, where the term $P(D|\alpha,\beta,M)$ can be expressed as in Equation 2.42.

$$P(\alpha,\beta|D,M) = \frac{P(D|\alpha,\beta,M)\cdot P(\alpha,\beta|M)}{P(D|M)} \tag{2.41}$$

$$P(D|\alpha,\beta,M) = \frac{P(D|\mathbf{w},\beta,M)\cdot P(\mathbf{w}|\alpha,M)}{P(\mathbf{w}|D,\alpha,\beta,M)} = \frac{Z_E(\alpha,\beta)}{(\frac{\pi}{\beta})^{n/2}(\frac{\pi}{\alpha})^{m/2}} \tag{2.42}$$

Z_E: integral to evaluate α and β.

n: number of observations.

m: total number of network parameters.

MacKay [1992] have shown that the following expression is valid:

$$Z_E(\alpha,\beta) \propto |\boldsymbol{H^{MAP}}|^{-1/2} \cdot exp(-F(w^{MAP})) \tag{2.43}$$

Where $\boldsymbol{H^{MAP}}$ is the Hessian matrix of the objective function, and MAP = maximum a posteriori. The Hessian matrix can be approximated as:

$$\boldsymbol{H^{MAP}} = \boldsymbol{J'J} \tag{2.44}$$

And where the $\boldsymbol{J}$ is the Jacobian matrix which contains the first derivatives of the network errors, with respect to the network parameters. This is why having differentiable transfer functions is important, as stated before. By finding the Jacobian matrix of the network errors, the weights can be updated as presented in Equation 2.45.

$$w^{l+1} = w^{l} - \left[\boldsymbol{J}^{T}\boldsymbol{J} + \mu\boldsymbol{I}\right]^{-1} \cdot \boldsymbol{J}^{T}\boldsymbol{e} \tag{2.45}$$

In Equation 2.45 w^{l+1} is the proposed weights in next iteration, w^{l} is the weights in the current iteration, μ is the Levenberg's damping factor and $\boldsymbol{e}$ is the error vector.

Backpropagation

The gradient of the cost function is important in the optimisation problem as indicated in the previous section. In neural network training schemes, the gradient descent and evaluation of ∇E is needed - therefore having an efficient way of evaluating the gradient is important. The error backpropagation is one of these ways.

For a standard feed-forward neural network, the output y_k are a linear combination of the input x_i and weights. A general activation can be expressed as in Equation 2.46.

$$y_k = \sum_{i} w_{ki} \cdot x_i \tag{2.46}$$

Then the sum in Equation 2.46 is transformed by a nonlinear activation function $h(\cdot)$ to give the activation z_j of unit j:

$$z_j = h(a_j) \tag{2.47}$$

For a multi-layer neural network, the output from the first layer is the input to the next and so on. For each training sample the unit values are calculated to create the forward-propagation-information-flow from input to output. This propagation leads to the hopthesis function $h_{\boldsymbol{w}}(\boldsymbol{x})$ which is used to calculate the cost function in Expression 2.48 [Bishop, 2006].

$$E_n(\boldsymbol{w}) = \frac{1}{2}\left[h_{\boldsymbol{w}}(\boldsymbol{x}^{n}) - y^{n}\right]^{2} \tag{2.48}$$

So by taking the gradient of the cost function, one can obtain Expression 2.49.

$$\frac{\partial E_n}{\partial w_{ij}} = \left[h_{\boldsymbol{w}}(\boldsymbol{x}^{n}) - y^{n}\right] x_i n \tag{2.49}$$

Which by the chain rule can be expressed on general form as:

$$\frac{\partial E_n}{\partial w_{ij}} = \frac{\partial E_n}{\partial a_j}\frac{\partial a_j}{\partial w_{ij}} \tag{2.50}$$

For convenience the notation in Equation 2.51 is often used.

$$\delta_j \equiv \frac{\partial E_n}{\partial a_j} \tag{2.51}$$

The δ in Equation 2.51 is often referred to as a *errors*, and when it is introduced to Equation 2.50, the Expression can be written as:

$$\frac{\partial E_n}{\partial w_{ij}} = \delta_j z_i \tag{2.52}$$

Which is due to the fact that $\frac{\partial a_j}{\partial w_{ij}} = z_i$ when differentiating Equation 2.46. Equation 2.52 indicates that the derivatives can be found by multiplying the unit output δ by the unit input z. For the output layer, this yields:

$$\delta_k = \boldsymbol{x}^k - y^n \tag{2.53}$$

An finally for any unit in the network passing information to another unit indexed k, yields:

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k \tag{2.54}$$

Expression 2.54 is known as the *bacpropagation* formula, where the value δ for a particular hidden unit can be obtained by propagating the δ's backwards from units higher up the network.

Bishop [2006] proposes the following backpropagation procedure:

1. Let some input vector $\boldsymbol{x}_n$ propagate through the network and use Expressions 2.46 and 2.47 to find the activation of all hidden units and output units.

2. Calculate output δ_k's from Equation 2.53.

3. Backpropagate the δ's using Expression 2.54 to obtain δ_j for each hidden unit in the network.

4. Use Expression 2.52 to evaluate the required derivatives.

2.4.5 Pitfalls when Designing Neural Networks

There are several pitfalls when designing artificial neural networks. Below is a presentation of a selection of these, focusing on over- and under-fitting, and over- and undertraining.

Over- and Underfitting

Overfitting is defined as the phenomena when fitting the observed data with an increasing number of layers, no longer indicates that we will get a decent out-of-sample error, and may lead to the opposite effect [Abu-Mostafa et al., 2012, Chapter 4]. This is observed for overly complex network layer structures. Figure 2.10 visualises an example where four polynomials of different orders, are fitted to a set of points. The bottom right image is an example of an overfitted function. The polynomial predicts the ten blue points perfectly, but in between points errors are introduced. Especially between the two rightmost-, and the two leftmost points, significant errors are observed.

Underfitting, on the other hand, is when an insufficiently simple network, with few hidden layers and few neurons, is chosen to represent the data. The selected model does not have enough neurons with biases and weight functions to describe the training data properly, and low performance (high MSE) is observed. The top two images in Figure 2.10 shows examples of under fitted polynomials.

For the data set in Figure 2.10, the bottom left polynomial of order three, is the one best representing the trend in the data. However, for practical applications, the optimal complexity is hard to find through visual observations and needs to be found through trial and failure. By analysing the performance of both training data and the test data, overfitted networks shows poor performance for the data set not used in training. Therefore careful analysis has to be carried out to ensure optimal network structure.

Over- and Undertraining

Analysing the algorithm performance for both training and test data is essential to reveal over- and undertraining. Tzafestas et al. [1996] describes overtraining as the phenomena where the MSE of the test set starts to increase, while the network still improves its performance in learning the training data. This is explained as the network ”memorising” the training data but fails to find an accurate description of the phenomena. Figure 2.11 shows the learning curves of a neural network with 40 neurons distributed as 10-20-10 in three hidden layers. At epoch 130 the test performance is at its lowest with an MSE of 0.0393, but after this point, the error increases drastically even though the net training performance continues to decrease. Tzafestas et al. [1996] describes an overtrained network as not useful, because it cannot recognise unknown patterns, and thus have a small generalisation ability. They recommend stopping the training at the lowest point in the test performance curve marked with a black vertical line in Figure 2.11, where it is known to be a good balance between the approximation accuracy and the generalisation ability.

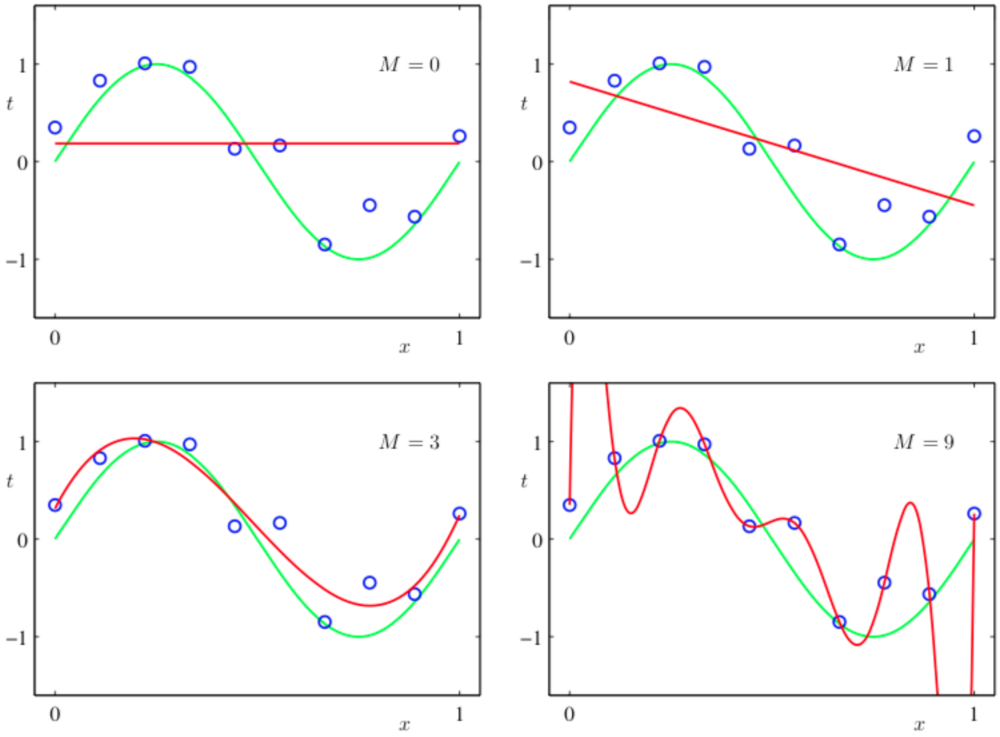

Figure 2.10: Plots of polynomials (red) having various orders M, against data points (blue) created by a sine function (green) plus some error [Bishop, 2006, figure 1.4].

2.4.6 Work Flow when Designing, Training and Validating Neural Networks with Neural Network Toolbox in Matlab

When designing, training and validating neural networks in Matlab, several functions, parameters and methods need to be chosen. Mathworks INC. [2018c] recommends the following procedure:

1. Import pre-processed training data.
2. Divide the data sample into three subsets: training set, validation set and test set.
3. Set desired training algorithm, or training function, and characteristic parameters.
4. Train network.
5. Post-process the results, compare network performance and determine degree of generalisation. Find best number of epochs.

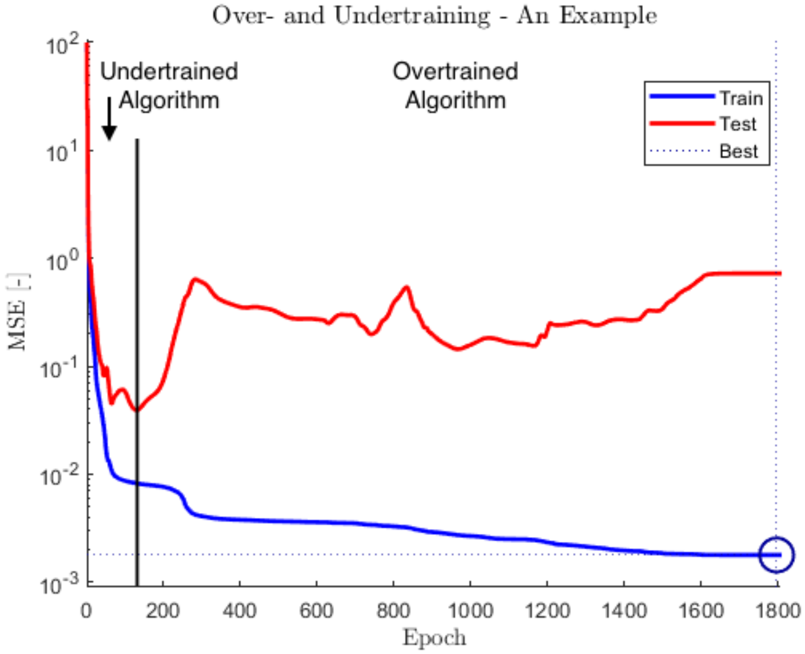

Figure 2.11: Training of a neural network for the full data set, training function: *trainbr* and three hidden layers [10 20 10] - trained for 1809 epochs.

The first step in neural network training is to import training data. The format of the input parameters can be as a vector with one parameter or as a matrix with multiple parameters. The output is a vector with the matching outputs to the parameters in the input matrix.

When training a neural network in Matlab the available data sample is divided into three subsets. The training set is used for computing the gradient and update weights and biases during training. When the network is trained for one epoch, the validation set is used for validation of the model. The error between predicted values and actual values in the validation set is used for determining the degree of over- and underfitting of the network. These results are used for training during next epoch. The last set is the test set. This test has the same function as the validation set, but the algorithm does not take the error between the predicted values- and the correct values of the test set into account during next epoch of training. Comparing training- and test performance is important to detect overfitting and choose the optimal number of epochs for training the neural network. There is no clear consensus on how the training/test-ratio of data should be to obtain optimal results, other than it should be between 90/10 and 75/25. The training data set should be as large as possible to ensure a generalised model, but the test sample should be large enough to reveal over-trained networks. Wanjawa and Muchemi [2014] proposes to use a ratio of 80/20, which will be done in this thesis.

In the next step, the training algorithm and characteristic parameters are set. There are several training algorithms available based on different schemes in the Neural Network Toolbox, some of them are presented in Table 2.2 [Mathworks INC., 2018a].

Table 2.2: Neural network training functions in the Neural Network Toolbox.

Algorithm	Description
trainlm	Levenberg-Marquardt method
trainrp	Resilent Backpropagation
trainscg	Scaled Conjugate Gradient
traincgb	Conjugate Gradient with Powell/Beale Restarts
traincgf	Fletcher-Powell Conjugate Gradient
traincgp	Polak-Ribiére Conjugate Gradient
trainoss	One Step Secant
traingdx	Variable Learning Rate Backpropagation
trainbr	Bayesian Regularisation Backpropagation

For data samples with noisy data the *trainbr* algorithm, based on the Bayesian Regularisation Backpropagation method presented in Section 2.4.4, is known to perform well. Next the characteristic parameters of the chosen training function needs to be set. The default values when using *trainbr* are presented in Table 2.3 [Mathworks INC., 2018b].

Table 2.3: Default values for the *trainbr* neural network training function.

Parameter	Default Value	Description
net.trainParam.epochs	1000	Maximum number of epochs to train
net.trainParam.goal	0	Performance goal
net.trainParam.mu	0.005	Marquardt adjustment parameter
net.trainParam.mu_dec	0.1	Decrease factor for mu
net.trainParam.mu_inc	10	Increase factor for mu
net.trainParam.mu_max	10^{10}	Maximum value for mu
net.trainParam.max_fail	0	Maximum validation failures
net.trainParam.min_grad	10^{-7}	Minimum performance gradient
net.trainParam.show	25	Epochs between displays (NaN for no displays)
net.trainParam.showCommandLine	false	Generate command-line output
net.trainParam.showWindow	true	Show training GUI
net.trainParam.time	inf	Maximum time to train in seconds

Finding the optimal values for the parameters in Table 2.3 should be done using trial and error method. In order to ”force” the network to train for a certain number of epochs, the net.trainParam.mu value should be set artificially low in order for the training not to finish due to early stopping as described earlier. When the data set is divided into subsets, training function is chosen and training parameters are set, the desired network structure needs to be chosen. The number of hidden layers and neurons in the hidden layers are specified as a vector of length m, with n_i neurons in each layer on the form: $[n_1, n_2, ..., n_{m-1}, n_m]$. One must also choose performance function, such as the mean squared error presented in Equation 2.31.

The next step is to train the network and monitor the progress. The training process is done with regards to the method in the chosen training function presented in Table 2.2.

The last step is to post-process the results. Performance is recorded for both training and test datasets during training, and the results should be examined to reveal over- and underfitting and over- and undertraining as described in Section 2.4.5. Then the network should be trained again for the optimal number of epochs before it can be exported as a Matlab function. If one wants to compare different network structures, steps three, four and five are typically set in a loop where parameters are changed before a new iteration.

Chapter 3

Results

In this chapter, the available data-set is presented together with a parameter study of neural network training parameters. The performance of tested networks with different parameter-sets is presented as well before fine-tuning of a most feasible neural network is carried out.

3.1 Data Presentation and Parameter Range

A substantial and time-consuming phase in this data-driven research project is the data pre-processing and preparation phase. The aim has been to digitise the data and make preparations for the machine learning algorithm to work as efficiently as possible when training. A neural network will not be more accurate than its training data-set, so ensuring correct data have been crucial to ensuring satisfactory quality of the empirical model. A data sample can have multiple problems which reduces its quality and hence reduce the quality of the method. Famili et al. [1997] presents the following common problems for real-world data samples:

- Corrupt and noisy data.
- Irrelevant data in the data sample.
- Missing attributes and small amounts of data.
- Fractured data: incompatible data.

Famili et al. [1997] proposes several methods for pre-processing data such as; data filtering, data ordering, data visualisation and data elimination. In the next section available parameters in the resistance data sample are presented.

3.1.1 Available Model Data

The data available per model can be divided into two sets: one set with dimensional characteristics per waterline, and one set with resistance data for a certain model tested at a certain waterline. Tables 3.1 and 3.2 presents the metadata for the two data-sets. Some parameters in Table 3.1 are seldom measured for each model, and are marked as *Not available for most models* in the table. The parameters in Table 3.2 are measured as the models are tested for different speed during the resistance tests. Some parameters are hard to record during the trials, such as wetted surface, and are therefore seldom recorded. These parameters, marked as: *Not available per resistance test* in Table 3.2, are on the other hand recorded for each waterline as presented in Table 3.1.

Table 3.1: List of available model and ship parameters per waterline.

Parameter	Abbreviation	Unit	Comment
Length overall	Loa	m	-
Length waterline	Lwl	m	-
Length between perpendiculars	Lpp	m	-
Total breadth	B	m	-
Total breadth waterline	Bwl	m	-
Breadth single hull	bwl	m	-
Distance between centerlines	s1	m	-
Distance between hulls (min)	s2 or s	m	-
Draught aft perpendicular (AP)	Tap	m	-
Draught front perpendicular (FP)	Tfp	m	-
Initial trim at zero speed	Trim	degrees	-
Tunnel height at AP	Dtap	m	Not available for most models
Tunnel height at FP	Dtfp	m	Not available for most models
Air gap tunnel at AP	ftap	m	Not available for most models
Air gap tunnel at FP	ftfp	m	Not available for most models
Volume of displacement	∇	m^3	-
Midship section coefficient	Cm	-	-
Prismatic coefficient	Cp	-	-
Block coefficient	Cb	-	-
Longitudinal centre of buoyancy	LCB	m	-
Wetted surface	S	m^2	-
Wetted surface of transom stern	Sb	m^2	-
Transverse projected area above WL	Atm or Atb	m^2	Not available for most models
Temp. tank or seawater	°C	-	
Hull roughness	H	m	-

Not all parameters presented in Tables 3.1 and 3.2 are used in the analysis, but all are

Table 3.2: List of available resistance data recorded during tests. Data is available per trial per design condition (waterline).

Parameter	Abbreviation	Unit	Comment
Model speed	Vm	m/s	-
Froude number	Fn	-	-
Resistance of model in test	Rtm	N	-
Residuary resistance coefficient	Cr	-	-
Full scale ship speed	Vs	m/s	-
Length of waterline in full scale	Lwl	m	Not available per resistance test
Wetted surface of full scale ship	S	m^2	Not available per resistance test
Required power	Pe	kW	-
Sinkage AP	AP	m	Not available for all models
Sinkage FP	FP	m	Not available all models
Running trim	Trim	degrees	Not available all models

imported into the electronic format. This is because data storage capacity and available memory can handle all data available in the resistance reports by SINTEF Ocean.

The most important feature of the data-set is its ability to represent the desired kind of high-speed displacement catamarans. If the neural network should be able to predict residuary resistance coefficient of all kinds of catamarans, all types of catamarans should be represented with the same amount of data. However, a network trained on data from different types of catamarans would not be able to predict accurately for all types, but rather give a rough estimate. The available model data given by SINTEF Ocean can be divided into two categories; old models which were used to develop CatRES and new models tested in the towing facility from 1997-2017. The models tested in the period 1990-1997 consists of 1082 model tests of mostly passenger-, car ferries and cargo catamarans with a few models being fishing vessels for line fishing. Most models designed with water-jet propulsion system are tested for Froude number in the range: $0.2 - 0.9$, while some of the latest tested designs are tested for Froude numbers between $0.5 - 1.5$. Models designed with propeller propulsion system are tested in the range $0.2 - 1.5$ as well. The data-set with new models consist of 1231 samples and are mostly ferries and some other vessels. Models designed with propeller propulsion system are consequently tested for Froude numbers in the range: $0.7 - 1.4$, while designs with water-jet are tested in the range: $0.2 - 1.4$.

Catamaran design has evolved from the first tests and hull characteristics have therefore changed as well. The data sample from 1990-1997 consists of catamarans with lengths between perpendiculars of 10m-96m, where most are about 40 meters. In the new data sample, catamarans have Lpps between 19m-130m, where most are about 40 meters. The design characteristics of the old models will be compared to the newer ones later in this thesis, but for now, it is assumed that SINTEF Ocean has tested a representative selection of passenger- and cargo catamaran ferries, together with some other catamarans with similar characteristics. It is also assumed that the data-sets represent these vessels in a good manner.

3.1.2 Normalised Model Data

LeCun et al. [1998, chapter 4.3] have found normalisation of the inputs to a neural network, to be a performance booster. The convergence is usually faster if the average of each input variable over the training set is close to zero. The input layer in the neural network will, however, map the input parameters to the desired format, as described in Section 2.4.1. Normalised input parameters are not new in marine applications, and different parameter rations have been used to compare designs for a long time. Ships vary greatly with design parameters such as dimensional size and speed, so finding dimensionless parameters which affect resistance have been important to compare designs. When choosing which normalised parameters to include in the training of neural networks, other considerations have to be made as well. The parameters should be as ”simple” as possible, as the method is to be used in an early phase of the design process - before detailed design is carried out. Secondly, the parameters should be representative of the properties of the ship.

Dimensional coefficients such as the block coefficient and slenderness ratio are known to have an impact on the resistance of catamarans. Hull separation and area of wetted stern are also important for fast catamarans. The stern will be partly- or completely dry for catamarans travelling at high Froude numbers as discussed in Section 2.1.3, and loss of hydrostatic pressure at higher speeds yields a significant contribution to catamaran resistance. The length/breadth-ratio is known to be an important parameter for ship resistance and capabilities at sea [Amdahl et al., 2013], so this parameter should be included in the neural networks. Table 3.3 shows which parameters that are included in CatRES and the empirical resistance model by Molland et al. [1995]. The last column indicates the parameters which are included as options for training the neural networks. A short-list with different parameter combination will be presented and tested later in this report.

3.1.3 Parameter Range in Data Sample

The model data provided by SINTEF Ocean have been divided into four groups: Data from models included in the data-set which CatRES was built upon and new models, which again is divided into models with jet propulsion and propeller propulsion. Table 3.4 shows the range and mean value of the different parameters in the four data-sets. Plots

Table 3.3: Vessel parameters included in CatRes [Rambech, 1998], Molland et al. [1995] method and the current work carried out in this thesis.

Parameter	CatRes (Rambech [1998])	Molland et al. [1995] Method	Current Work
Lwl/Bwl	X	X	X
B/T	X	X	X
$L/\nabla^{1/3}$	X	X	X
$S/\nabla^{2/3}$	X	-	X
$s2/Lwl$	X	X	X
Sb/S	-	-	X
Boolean (Jet/Prop)	-	-	X

showing parameter range for the models are presented in Appendix A.1. The Froude number ranges presented in Table 3.4, indicates some of the new catamaran design being tested at low speeds. This is for one model only, which have been tested for the wide Fn range: 0.092 – 1.110. Keeping outliers like this vessel in the data sample gives an illusion of wide parameter range, but because there are few samples with such low Froude numbers, the training cases in this range are low, and hence accuracy will be poor as well. All samples are however kept in the training sample, in order to have the largest training foundation as possible.

Table 3.4: Parameter range in SINTEF Ocean data sample.

Parameter		New: Jet	New: Propeller	Old: Jet	Old: Propeller
Fn	Range:	[0.0920-1.3260]	[0.6360-1.3800]	[0.1940-1.4360]	[0.1770-1.5540]
	Mean:	0.7118	0.9659	0.6009	0.7693
Lwl/Bwl	Range:	[2.3800-14.8310]	[4.7280-6.6130]	[3.0100-3.9400]	[1.9800-7.4074]
	Mean:	8.3281	5.1281	3.7109	2.7940
B/T	Range:	[1.2815-12.0650]	[1.9322-3.3813]	[1.1700-2.9700]	[1.1538-2.6037]
	Mean:	3.0389	3.0602	1.7862	1.6574
$L/\nabla^{1/3}$	Range:	[5.4187-9.7201]	[6.4253-8.0024]	[7.5700-10.8400]	[5.3100-9.2400]
	Mean:	6.8992	6.7285	9.4097	7.1257
$S/\nabla^{2/3}$	Range:	[6.6137-12.2942]	[10.1493-10.9816]	[9.1000-11.5400]	[7.8200-12.4000]
	Mean:	10.6331	10.4815	10.6023	9.9036
Sb/S	Range:	[0.0035-0.0216]	[0.0091-0.0185]	[0-0.0200]	[0-0.0106]
	Mean:	0.0128	0.0144	0.0011	0.0015
$s2/Lwl$	Range:	[0.0969-0.3152]	[0.1855-0.2696]	[0.1339-0.2401]	[0-0.3782]
	Mean:	0.2061	0.2154	0.2027	0.2106
$Cr \cdot 10^3$	Range:	[0.4620-9.7160]	[0.9890-3.8130]	[0.1210-5.3810]	[0.8630-18.3810]
	Mean:	2.1981	1.7125	2.3204	4.6835
Samples:		978	253	751	331

As mentioned earlier, Famili et al. [1997] recommends data filtering, data ordering, data visualisation and data elimination as methods of data pre-processing. For the data in Table 3.4 data visualisation have been utilised to find anomalies and errors in the data.

Outliers have been examined and the values were compared to the ones listed in the resistance reports. All extreme values were found to be mistakes in calculations and typos, and none remained unsolved. The total number of data points in the sample is 2313.

An issue experienced when working on data pre-processing was missing values. When importing values to Matlab, the missing values are set to zero, before training and validation starts. Especially for the older models, parameters such as the wetted area of stern were seldom tabulated. Figure 3.1 shows how the ratio Sb/S is absent for older models of both jet- and propeller type. This will introduce errors in the neural networks and be an obstacle for the training.

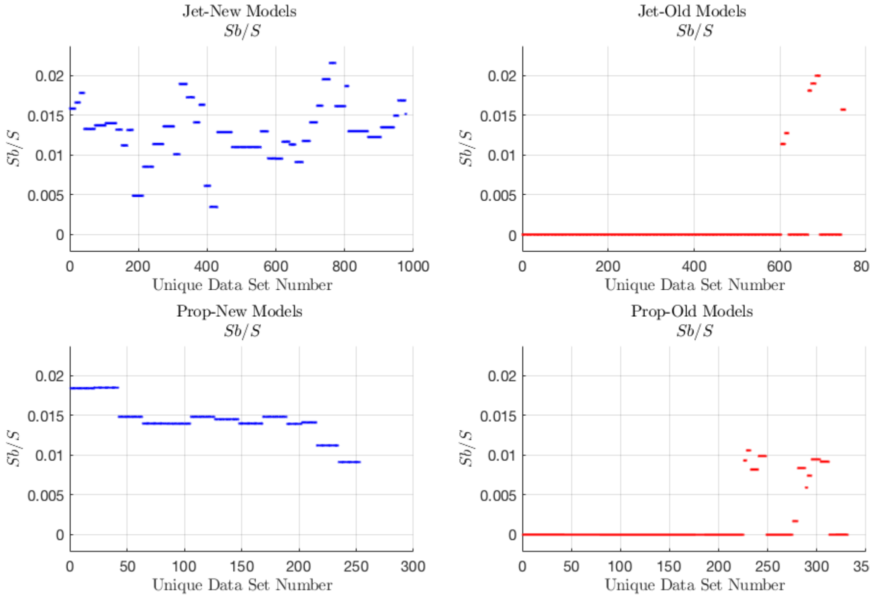

Figure 3.1: Parameter range of Sb/S in data-sets. Missing values in the sample with "old" models (right) are set to zero.

Famili et al. [1997] mentions data elimination as a possible way of pre-processing data-sets. However, excluding data samples with missing values means excluding large amounts of data from training- and validation data-sets. Large data-sets are required when training neural networks, and training with half the data samples may lead to poor performance. Another consideration is how the validity range of the neural network is for a reduced data-set. When limiting the data sample down to fewer models, the parameter-range is limited as well, leading to an empirical model with a narrower field of application. Table 3.5 compares the validity range of the total data sample and the data sample with new model data only.

By taking away the old data, the data sample is reduced with 46.78%, which has an impact on the data range. The relative difference for the Sb/S-ratio is however artificially high as several of the data points are set to zero, as discussed earlier. The largest relative difference for parameter range is seen for the slenderness-ratio $L/\nabla^{1/3}$ with a reduction of 22.22%. The mean value is reduced by 11.18%, and by looking at the parameter ranges in Table 3.4, we can see that the older models have a tendency of higher slenderness-ration than the new ones. The effect of removing data from the old models will be analysed later in this thesis.

Table 3.5: Comparing parameter range for full data sample and sample with new models only. Relative reduction presented in the right column $\left[\frac{New-Full}{Full} \cdot 100\%\right]$.

Parameter		New Models	Full Set	Relative Reduction
Fn	Range:	[0.0920-1.3800]	[0.0920-1.5540]	-11.91%
	Mean:	0.7640	0.711	7.33%
Lwl/Bwl	Range:	[2.3800-14.8310]	[1.9800-14.8310]	-16.05%
	Mean:	7.6705	5.6870	34.877%
B/T	Range:	[1.2815-12.0650]	[1.1538-12.0650]	-1.17%
	Mean:	3.0433	2.4368	24.89%
$L/\nabla^{1/3}$	Range:	[5.4187-9.7201]	[5.3100-10.8400]	-22.22%
	Mean:	6.8641	7.7280	-11.18%
$S/\nabla^{2/3}$	Range:	[8.6137-12.2942]	[7.8200-12.4000]	-19.64%
	Mean:	10.6020	10.5021	0.95%
Sb/S	Range:	[0.0035-0.0216]	[0-0.0216]	-16.14%
	Mean:	0.0132	0.00764	73.31%
$s2/Lwl$	Range:	[0.0969-0.3152]	[0-0.3782]	-42.28%
	Mean:	0.2080	0.2067	0.66%
$Cr \cdot 10^3$	Range:	[0.4620-9.7160]	[0.1210-18.3810]	-49.32%
	Mean:	2.0983	2.5404	-17.40%
Samples:		1231	2313	-46.78%

3.1.4 Parameter-set Short List

When choosing which parameters to include in the network training data-sets, inspiration was taken from the parameters used in CatRES and Molland et al. [1995] method, together with the resistance theory presented in Section 2.1. This lead to the parameters in Table 3.3. When assembling the parameters into a short list of test sets, inspiration was again taken from CatRES and Molland et al. [1995] method. The six proposed test sets are presented in Table 3.6. Parameter-set one is the parameters used in CatRES, while set two is CatRES including the boolean value (Jet/Prop): *false* - catamaran with propeller propulsion and *true* - catamaran with jet propulsion. Parameter-set three is same as parameter-set one, where the wetted-stern-wetted-area-ratio Sb/S is included. The fourth set are the parameters in set one, with both boolean value and Sb/S-ration included. The fifth set is a typical neural network training set, where all available parameters are included. This is to see whether the network shows better performance for as much data as possible. The last parameter-set is inspired by the Molland et al. [1995] method, containing the same parameters. The performance of the networks trained on the parameter-sets in Table 3.6 will be presented later in this thesis.

Table 3.6: Parameter combinations in the parameter-set-short-list.

Parameter	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
Boolean Jet/Prop	-	X	-	X	X	-
Fn	X	X	X	X	X	X
Lwl/Bwl	-	-	-	-	X	X
B/T	X	X	X	X	X	X
$L/\nabla^{1/3}$	X	X	X	X	X	X
$S/\nabla^{2/3}$	X	X	X	X	X	X
$s2/Lwl$	X	X	X	X	X	X
Sb/S	-	-	X	X	X	-

3.2 Neural Network Parameter Study

The neural network design has a major impact on the performance of the model, and on what kinds of complexity, the network can handle. There is a lot of literature on neural network performance and how to determine overtraining and overfitting, but there are few guidelines on how to structure the actual design parameters such as number of hidden layers and neurons in hidden layers. The general consensus is to use trial-and-failure on varied sets of networks, with different model parameter-sets. By taking the pitfalls from Section 2.4.5 into account, an optimal neural network is found.

Networks were trained using a desktop computer with the specifications presented in Table 3.7. This is not a high-performance computer, but it handled training the networks in a satisfactory manner. As presented in Section 2.4.6, several parameters need to be set and a training function must be chosen when training neural networks. In the next subsections parameter studies of: training functions, parameter-sets from the shortlist in Table 3.6, network structure and data-sets will be carried out in order to find the best combination,

Table 3.7: Hardware and software used when training neural networks.

Hardware	
CPU Model	Intel Core i5-4440
CPU Speed	3.10 GHz
Memory (RAM)	8 GB
Software	
Operating System	Windows 10 Pro (v. 1709)
Matlab and Neural Network Toolbox v.	R2018a

and hence the best performing neural network.

For the study carried out in the next subsections the default training values, presented in Table 2.3 have been used, unless stated otherwise. Different network structures have been tested in this parameter study in order to find the best overall performing parameters. The networks used are presented in Table 3.8 and are used for all tests, unless stated otherwise. One should note that the performance is plotted with a logarithmic y-scale for easier to compare similar results.

Table 3.8: First test network layouts. Various network sizes and hidden layers are tested.

Net Number	No. Hidden Layers	Node Distribution	Total Number of Nodes
1	1	15	15
2	1	20	20
3	1	30	30
4	1	40	40
5	1	50	50
6	1	60	60
7	1	100	100
8	2	20-20	40
9	2	30-30	60
10	2	20-30	50
11	2	30-20	50
12	3	15-15-15	45
13	3	20-20-20	60
14	3	10-15-20	45
15	3	20-15-10	45
16	3	20-30-20	70
17	3	30-15-30	75
18	4	10-10-10-10	40
19	4	15-15-15-15	60
20	4	20-20-20-20	80
21	4	20-30-30-20	100
22	4	30-40-40-30	140
23	4	10-40-50-30	130

3.2.1 Comparing Training Functions

First, the training functions presented in Table 2.2 are compared in order to find which one should be used together with the networks presented in Table 3.8. Figure 3.2 shows the performance of the different training functions. *Trainbr* and *trainlm* yields the best performance (lowest mean square error) for both the training data-set and the test data-sets. Training time is however significantly higher for the *trainbr* function, which can be seen in Figure B.1 in Appendix B.1. Because accuracy is one of the most important properties of an empirical method, the *trainbr* training function is used for the other test in this section. Even though the elapsed time is 10^2 times higher than the other functions, it is also 10^2 times more accurate. *Trainbr* shows a trend of being more accurate than *trainlm*, and is therefore preferred of the two.

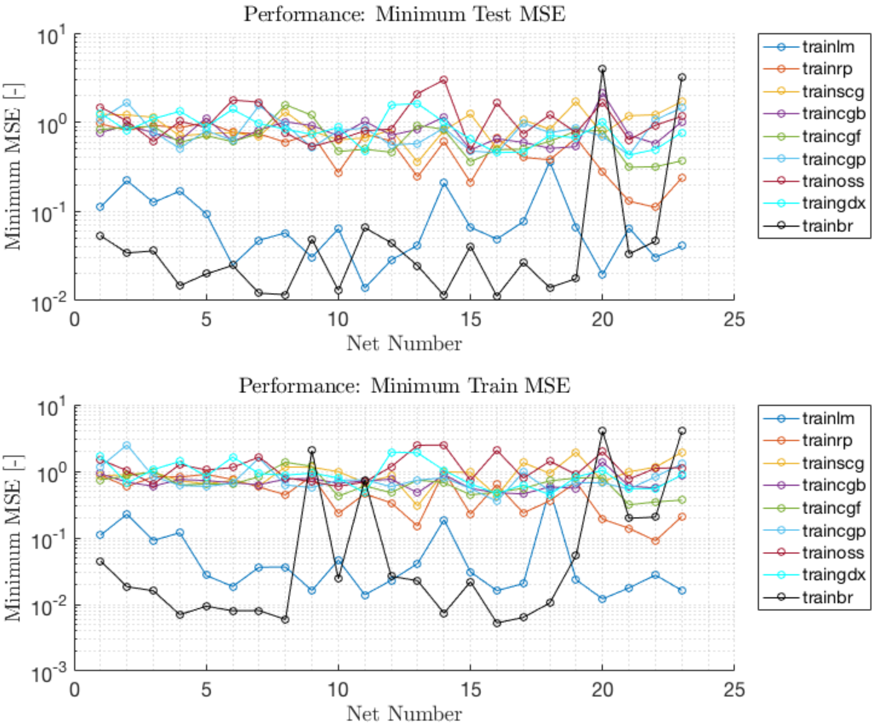

Figure 3.2: Results when testing training functions from Table 2.2 on networks in Table 3.8.

3.2.2 Comparing Parameter-sets and Network Structure

Next, the parameter-sets presented in Table 3.6 are compared when training and testing the networks in Table 3.8. Here the default training values are used and the networks are tested on the full data-set. The results are presented in Figure 3.3. The performance of the networks trained on the different parameter-sets shows similar behaviour for networks one to eight. As the complexity of the networks increases the test performance increases (mean squared error decreases). For network eight and upwards there is a clear individual difference for the network performance, which can be seen in the lower plot in Figure 3.3. The test performance has the same behaviour for the most complex networks from net 19 to 23. The best test performance is observed for network number 16 with parameter-set three. Plots for elapsed time in the training can be seen in Appendix B.1, Figure B.2.

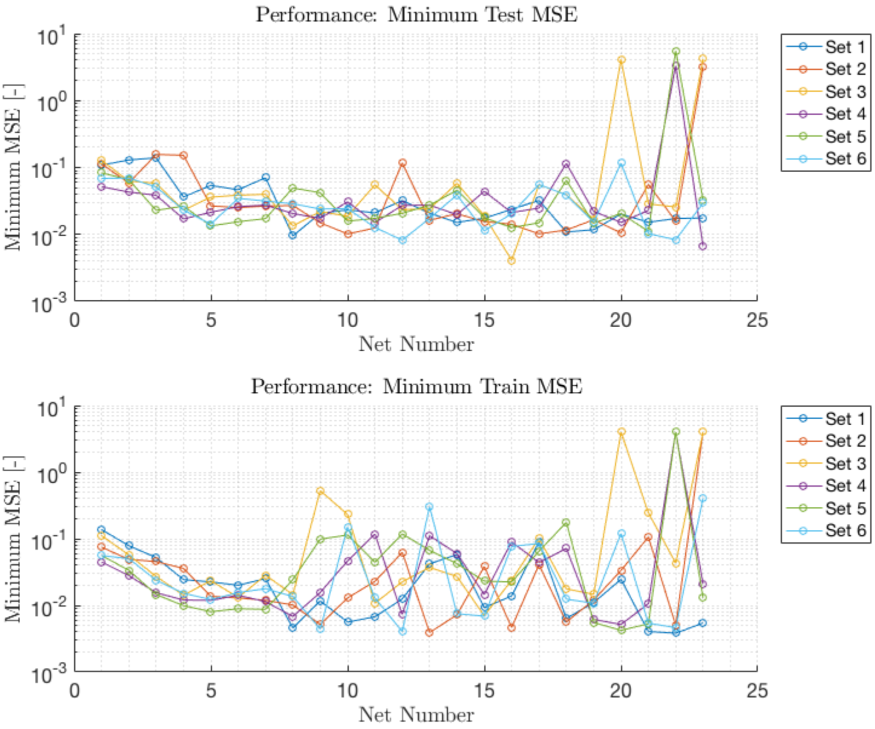

Figure 3.3: Results when testing parameter-sets from Table 3.6 on networks in Table 3.8.

3.2.3 Comparing Data-Sets: Full vs. New Models Only

As discussed in Section 3.1.3, inserting zeros for the missing values will lead to poor neural network performance as the training algorithm tries to train networks on the missing values. Figures 3.4 and 3.5 compares the performance of the networks in Table 3.8 and the data-sets in Table 3.6, when the networks are trained on the total data-set and the data-set with new models only. The best overall performance is observed for the trained networks with the new data only, with a performance 10-times better than the full data-set. The elapsed training time is presented in Figures B.3 and B.4 in Appendix B.1. One can see the elapsed time being similar for the two data-sets, for the lower network numbers. For network 19 and above, the results diverge and no particular tendency can be seen. Best performance characteristics are presented in Table 3.9.

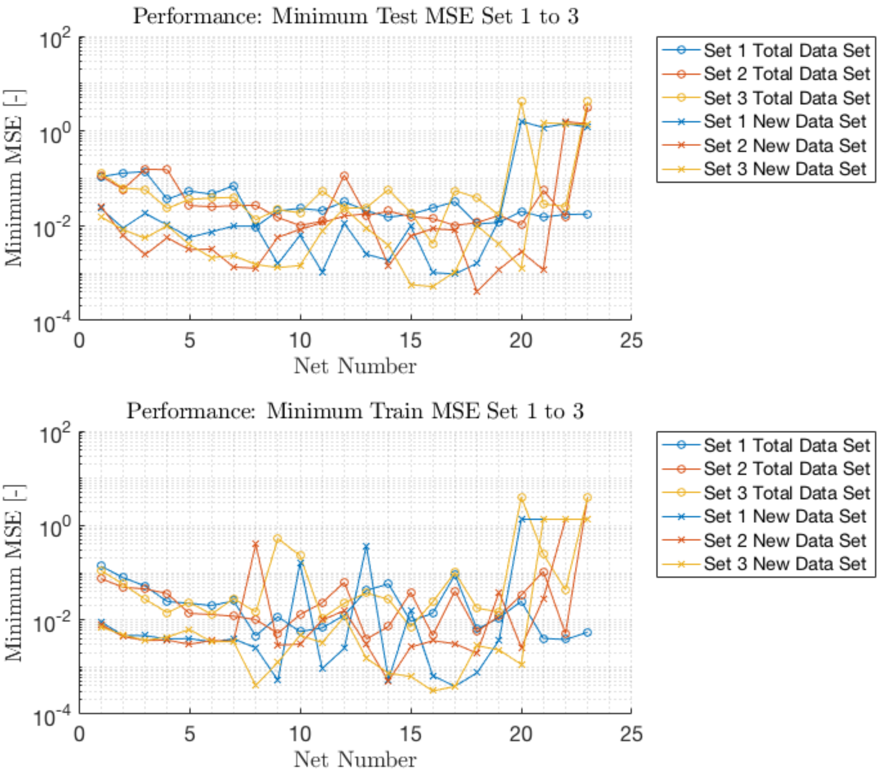

Figure 3.4: Results when comparing total data-set to new models only on networks in Table 3.8, part 1.

The best combination of parameter-set and neural network characteristics differs between

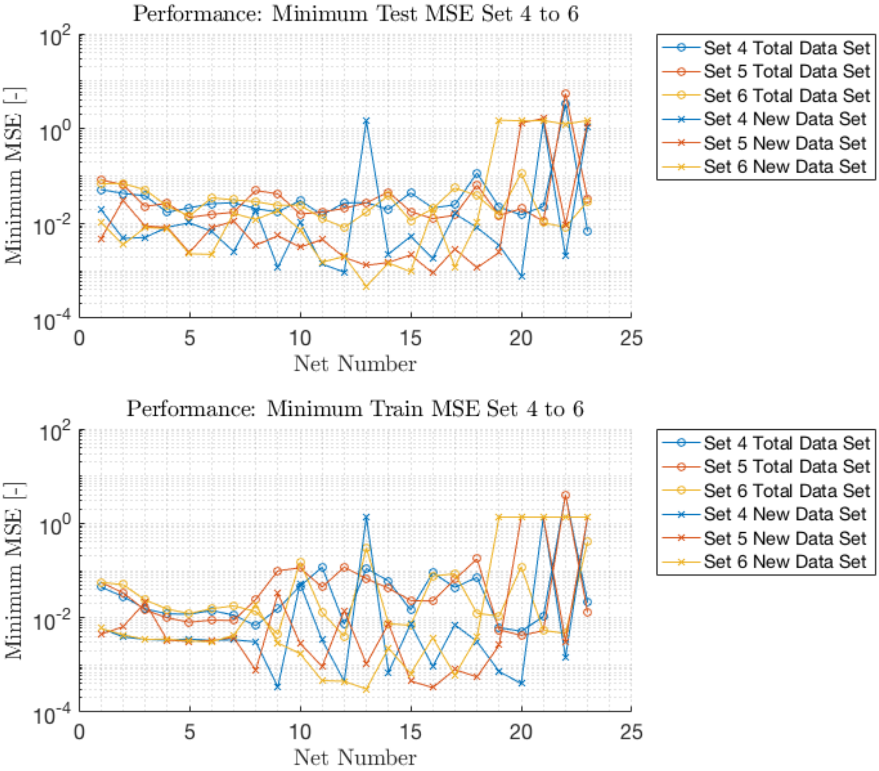

Figure 3.5: Results when comparing total data-set to new models only on networks in Table 3.8, part 2.

the two data-sets. For the total data-set, parameter-set three in combination with network number 16, gives the best test performance of $4.121 \cdot 10^{-3}$. For the data-set with new models only, parameter-set two in combination with network number 18 gives the best test performance of $4.082 \cdot 10^{-4}$. Hence the performance of the network trained on the new model data only, perform ten times better than the network trained on the whole data-set. This could be due to a more homogeneous data sample with narrower parameter spans, as indicated in Table 3.5

Which of the data-sets to use in further analysis, needs to be considered carefully. One needs to compare the loss of training data and loss of parameter validity range to the performance of the neural network. After discussing this issue with co-supervisor Hans Jørgen Rambech, the validity range was found to be most important, and therefore the full data-set together with parameter-set three is used in further analysis.

Table 3.9: Best performance characteristics for networks trained on full data-set and data-set with new models only.

Data-set	Optimal parameter-set	Optimal Network	Test MSE	Train MSE	Epoch
Total	Set 3	[20-30-20]	$4.121 \cdot 10^{-3}$	$3.109 \cdot 10^{-4}$	680
New Models	Set 2	[10-10-10-10]	$4.082 \cdot 10^{-4}$	$1.950 \cdot 10^{-3}$	656

3.3 Finding the Optimal Network

Through the parameter study carried out in the previous sections, the following knowledge is gained, and the following parameters are used in the optimisation tests. In Section 3.2.1 the training function *trainbr* gave the best performance of the ones available. Therefore this function is used for the optimised network as well. It was shown that networks 16 and 18 from Table 3.8 gave the best performance for training data-set. One can also observe that the symmetrical networks appeared to perform best, and hence the focus will be on symmetrical networks when finding the best one. Both larger three layer- and smaller four layered neural networks is tested in this section. The node distributions for the tested networks can be seen in Table 3.10.

The parameter-sets presented in Table 3.6, performed differently for the full data-set and for the data-set containing new model data only. After the discussion in Section 3.2.3 parameter-set three was chosen to be used together with the full data-set, in order to get the desired validity range for the empirical model.

It is important to remember that training neural networks is a generic training process, which produces different results for each training session. This is because initial weights and biases are set randomly, and therefore the training sessions have different starting points. However, setting customised initial parameters introduces a risk of missing an optimal network because wrong initial parameters are set. To account for this, each network in Table 3.10 is trained 15 times, and the best network overall is kept. The results can be seen in Figures 3.6 and 3.7.

By examining the results of test performance in Figure 3.6, one can see that the minimum recorded mean squared error fluctuates significantly from run to run. This is due to the generic training process of neural networks as discussed earlier. The test performance varies between 10^{-1} and $5 \cdot 10^{-3}$ with a concentration around $2 \cdot 10^{-2}$. There are a few outliers with a performance around $2 \cdot 10^{0}$, and the best observed test performance is $3.461 \cdot 10^{-3}$ by network 28.

Training performance presented in Figure 3.7, varies greatly between runs. Network 27 has

Table 3.10: Test network layouts in fine-tuning where various network sizes and hidden layers are tested.

Net Number	No. Hidden Layers	Node Distribution	Total Number of Nodes
1	3	15-25-15	55
2	3	16-25-16	57
3	3	16-26-16	58
4	3	17-26-17	60
5	3	17-27-17	61
6	3	18-27-18	63
7	3	18-28-18	64
8	3	19-29-19	67
9	3	19-30-19	68
10	3	20-30-20	70
11	3	21-30-21	72
12	3	21-31-21	73
13	3	22-31-22	75
14	3	22-32-22	76
15	3	22-33-22	77
16	3	23-32-23	77
17	3	23-33-23	79
18	3	24-34-24	82
19	3	24-35-24	83
20	3	25-35-25	85
21	4	7-7-7-7	28
22	4	7-8-8-7	30
23	4	8-8-8-8	32
24	4	8-9-9-8	34
25	4	9-9-9-9	36
26	4	9-10-10-9	38
27	4	10-10-10-10	40
28	4	10-11-11-10	42
29	4	11-11-11-11	44
30	4	11-12-12-11	46
31	4	12-12-12-12	48
32	4	12-13-13-12	50
33	4	13-13-13-13	52

the greatest difference, with the best performance of $4.5 \cdot 10^{-3}$ and a worst performance of $4.004 \cdot 10^{0}$. The best observed training performance is of network 12 with a performance of $2.169 \cdot 10^{-3}$. The elapsed time, presented in Figure B.5 in Appendix B.1, appears to be consistent between runs, except for a few outliers with higher and lower training time.

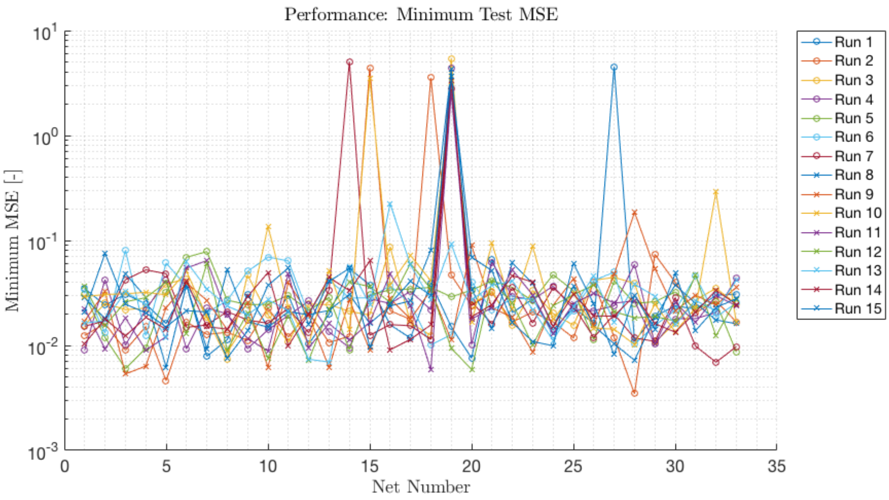

Figure 3.6: Results for fine-tuned networks, test performance.

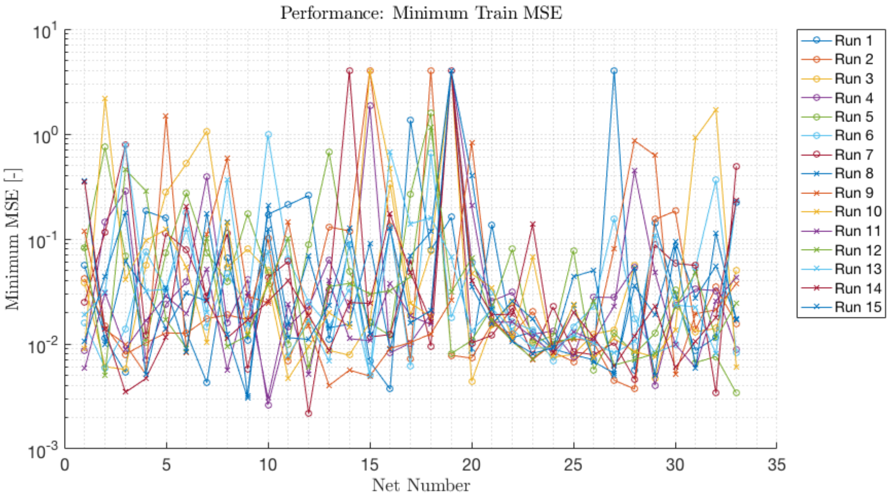

Figure 3.7: Results for fine-tuned networks, train performance.

The optimal network and its characteristics are presented in Table 3.11 and a visualisation of the network with input and output layers, is presented in Figure 3.8. The visual presentation in Figure 3.8 indicates the desired properties of the six parameters in parameter-set three going into the network, and one parameter: the residuary resistance coefficient $Cr \cdot 10^3$, going out of the network.

Table 3.11: Optimal network performance and characteristics.

Data-set	Optimal parameter-set	Optimal Network	Test MSE	Train MSE	Best Epoch
Total	Set 3	[10-11-11-10]	$3.461 \cdot 10^{-3}$	$3.725 \cdot 10^{-3}$	996

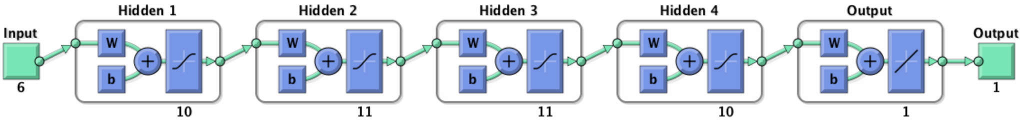

Figure 3.8: Visual presentation of the best performing network.

3.4 Evaluating the Optimised Network

In order to evaluate the optimised network found in the previous section, the validity range and performance for an independent data-set will be evaluated. As stated before, the main goal for a neural network is not to ”remember” which output to give for a given input but to represent the patterns in the problem and therefore have a large generalisation ability. This is why available data is divided into training and test data-sets as presented before. In order to test the validity range and performance for an independent data-set, the network is tested on data from resistance tests carried out by Insel and Molland [1992], which is tabulated in their paper. The parameter range of their data, called Southampton data, is compared to the data from SINTEF Ocean in Table 3.12. The parameter range of the Southampton data is also visualised in Appendix A.2.

By comparing the parameter ranges in Table 3.12, some properties are visible. First of all, the Sb/S-ratio is not available for the Southampton data, while most of the other parameters are within the validity range of the SINTEF Ocean data, except Lwl/Bwl, $S/\nabla^{2/3}$ and $s2/Lwl$. Out of the 719 data points in the Southampton sample, 75 points are outside the Lwl/Bwl-range, 385 points are outside the $s2/Lwl$-range and none are outside all.

In order to test the neural network with the Southampton data, and evaluate the effect of exceeding the validation range, values must be set for the missing Sb/S-ratio. Two feasible solutions will be compared: setting the parameter equal to zero, and using the mean value of the SINTEF Ocean sample: $Sb/S = 0.00764$.

Large MSE was observed when comparing the predicted and the actual values for residuary resistance coefficient for the Southampton data during validation. Figure 3.9 shows how the network consequently over, and underpredict residuary resistance coefficient. Results

Table 3.12: Parameter range in Southampton data sample.

Parameter		Southampton Data	SINTEF Ocean Data
Fn	Range:	[0.2000-1.0000]	[0.0920-1.5540]
	Mean:	0.6227	0.711
Lwl/Bwl	Range:	[7.0000-15.1000]	[1.9800-14.8310]
	Mean:	10.8982	5.6870
B/T	Range:	[1.5000-2.5000]	[1.1538-12.0650]
	Mean:	2.0042	2.4368
$L/\nabla^{1/3}$	Range:	[6.2700-9.5000]	[5.3100-10.8400]
	Mean:	8.3035	
$S/\nabla^{2/3}$	Range:	[6.6648-8.4609]	[7.8200-12.4000]
	Mean:	7.7378	10.5021
Sb/S	Range:	Not Available	[0-0.0216]
	Mean:	Not Available	0.00764
$s2/Lwl$	Range:	[0.2000-0.5000]	[0-0.3782]
	Mean:	0.3606	0.2067
$Cr \cdot 10^3$	Range:	[1.1210-15.4170]	[0.1210-18.3810]
	Mean:	3.7914	2.5404
Samples:		719	2313

of both large positive and negative coefficients are observed, where the worst predictions are for $Fn < 0.45$. This is a general tendency for the models outside the parameter range as well, presented in Figures B.11 to B.14 in Appendix B.3. Poor performance may be due to the network being overtrained, so the network is trained again for fewer epochs in order to minimise the mean squared error, and hopefully increase the generalisation ability for the model. The focus was on minimising the mean squared error for the data-set within parameter range, which gave 19 as the optimal number of training epochs. Mean squared errors for within- and outside parameter range analysis, are presented in Table 3.13.

The network trained for fewer epochs than found to be optimal in the earlier analysis, produces far lower mean squared errors for the Southampton data-set. Errors are on the other hand higher than the ones found during network training with the SINTEF Ocean data-set.

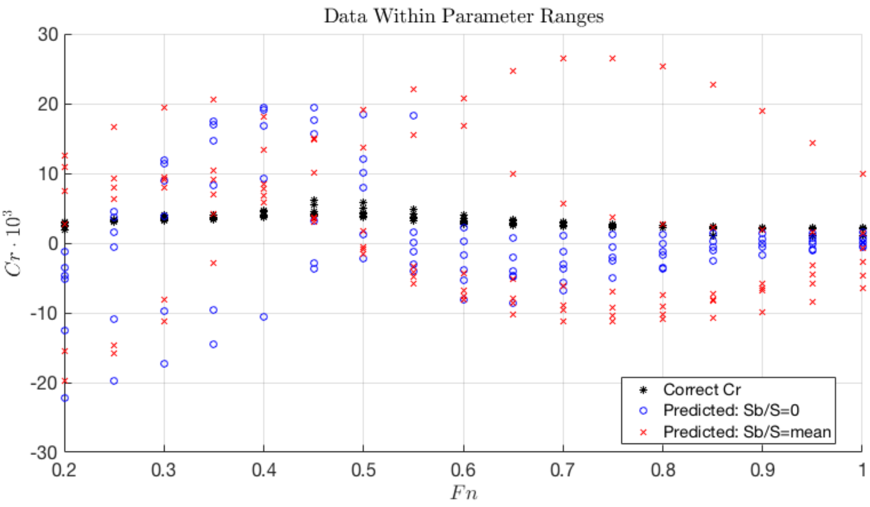

Figure 3.9: Validation results for network tested on Southampton data within parameter range, for the network trained for 996 epochs.

Table 3.13: Results for validation analysis with Southampton data-set.

Data-set	$Sb/S=$	No. samples	No. training epochs	Error (MSE)
Data within parameter range	0	102	996	72.4170
Data within parameter range	0.00764	102	996	115.8482
Data within parameter range	0	102	19	0.2298
Data within parameter range	0.00764	102	19	0.2072
Data outside Lwl/Bwl-range	0	75	996	72.3573
Data outside Lwl/Bwl-range	0.00764	75	996	55.2932
Data outside Lwl/Bwl-range	0	75	19	0.5110
Data outside Lwl/Bwl-range	0.00764	75	19	0.5560
Data outside $S/\nabla^{2/3}$-range	0	419	996	109.1139
Data outside $S/\nabla^{2/3}$-range	0.00764	419	996	126.8884
Data outside $S/\nabla^{2/3}$-range	0	419	19	0.6067
Data outside $S/\nabla^{2/3}$-range	0.00764	419	19	0.8517
Data outside $s2/Lwl$-range	0	385	996	84.7144
Data outside $s2/Lwl$-range	0.00764	385	996	65.4421
Data outside $s2/Lwl$-range	0	385	19	0.4975
Data outside $s2/Lwl$-range	0.00764	385	19	0.7644

3.4.1 Comparing Against Data Within Parameter Range

The number of data points within the valid parameter range of the neural network is 102, therefore it should be possible to show what degree of generalisation the network has. However, this requires the Southampton data to be representative of the same types of vessels included in the SINTEF Ocean data-set. The results when testing Southampton data within parameter range are presented in Table 3.13 and in Figure 3.10. The neural network predicts the general behaviour of Cr, some errors are visible for $0.4 < Fn < 0.55$ and for $Fn = 0.85$, 0.95 and 1.0. Especially the last three values are over predicted by the network in Figure 3.10.

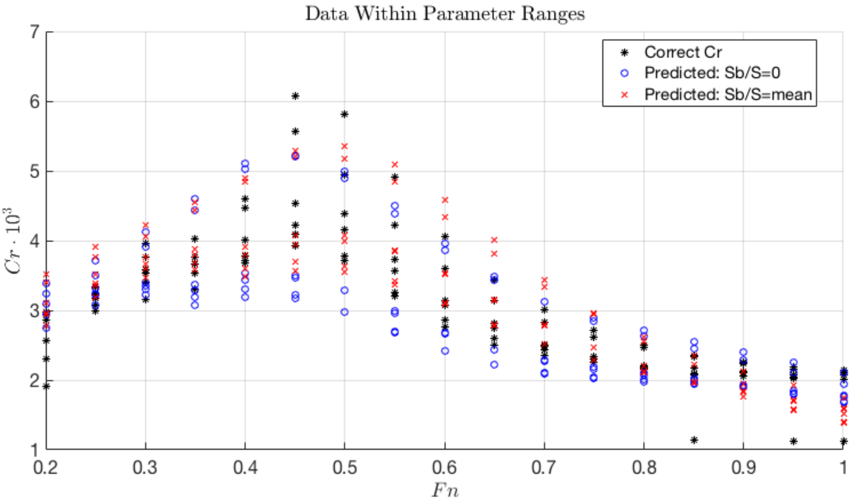

Figure 3.10: Validation results for Southampton data within network validity parameter range.

3.4.2 Comparing Against Data Exceeding Parameter Range

Next, the neural network is tested for vessels exceeding the parameter ranges of: Lwl/Bwl, $S/\nabla^{2/3}$ and $s2/Lwl$. The mean squared error for the tests are presented in Table 3.13 and the predictions are compared to the correct residuary resistance coefficients in Figures 3.11 to 3.13.

Both networks assuming $Sb/S = 0.00764$ and $Sb/S = 0$, tends to underpredict residuary resistance coefficient for models with $Lwl/Bwl > 14.831$. The network assuming $Sb/S = mean$ gives the best estimates for $Fn < 0.75$ but predicts lower Cr than the correct one and the network assuming $Sb/S = 0$ at high Froude numbers. This can be seen in Figure 3.11. At $Fn = 1.0$ both networks miss the tendency of Cr flatting out, causing under-predicted results.

For models with $S/\nabla^{2/3} < 7.82$ the same behaviour as before is observed in Figure 3.12.

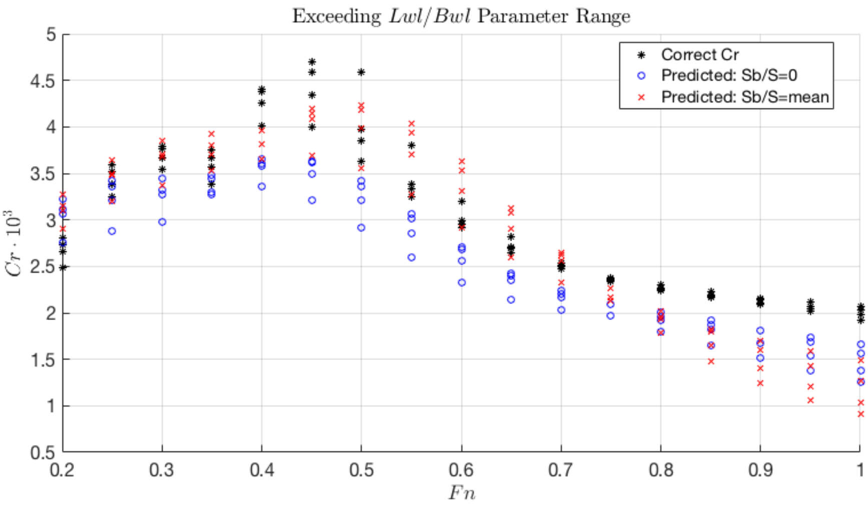

Figure 3.11: Validation results for Southampton data outside Lwl/Bwl network validity range.

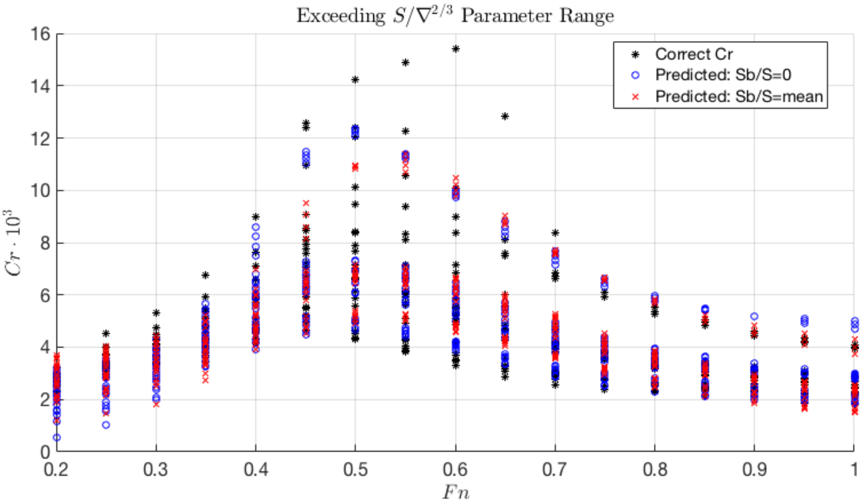

Figure 3.12: Validation results for Southampton data outside $S/\nabla^{2/3}$ network validity range.

Both models predict the general behaviour of the data sample. For $0.45 < Fn < 0.65$, the four highest coefficients are not predicted at all. By looking closer at the model data from Molland et al. [1995], these extremes appears to be from model 3b. This model has a wetted area 24% larger than model 4a, which has the second highest wetted area of these

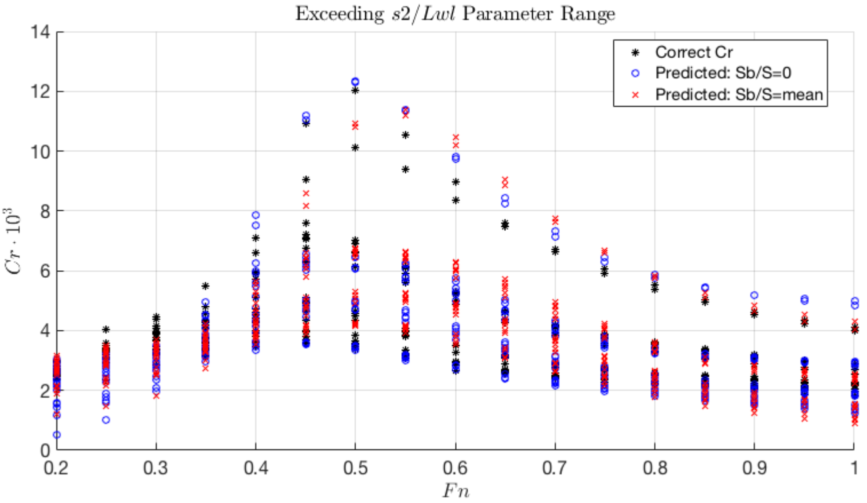

Figure 3.13: Validation results for Southampton data outside $s2/Lwl$ network validity range.

models. The $L/\nabla^{1/3}$ parameter is also 15% lower than for model 4a, indicating that model 3b is wider and with a larger wetted surface than the other models. Residuary resistance is overall higher for this model, with especially high resistance around $Fn = 0.55$. The neural network predicts Cr to a large extent for the other vessels, for both higher and lower Froude numbers.

For models with $s2/Lwl > 0.3782$ presented in Figure 3.13, one can see the model predicting the general behaviour of the vessels in the sample. Now the highest values appear to be estimated by the model. For higher Fn both using $Sb/S = 0$ and $Sb/S = mean$ yields good performance and predicts Cr close to the measured ones.

3.4.3 Comparing Networks to SINTEF Ocean Data

In the evaluation process in the previous section, the optimal network had to be trained for fewer epochs to gain better performance for the Southampton data-set. Now, the two networks will be re-tested on the SINTEF Ocean data-set to evaluate performance and hence the ability to reproduce residuary resistance coefficients from model tests. The performance is presented in Table 3.14 and in Figure 3.14. The full data sample with residuary resistance coefficients from SINTEF Ocean data sample and the predicted ones, with the networks from Table 3.14, are presented in Figure B.10, in Appendix B.2.

Table 3.14: Prediction error of the two trained networks to SINTEF Ocean data.

Network Structure	Number of Training Epochs	Error (MSE)
[10-11-11-10]	19	0.1135
[10-11-11-10]	996	0.0035

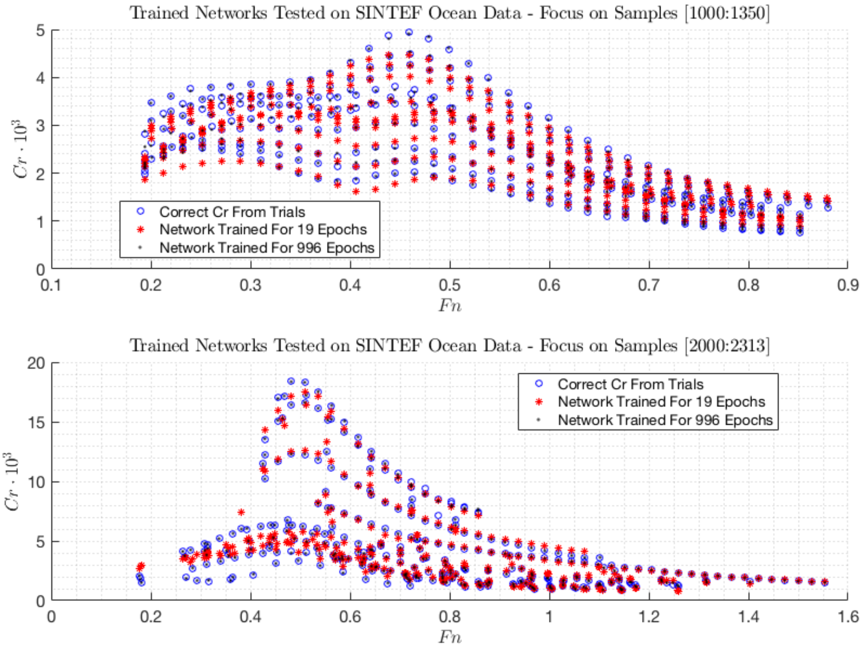

Figure 3.14: Network ability to predict Cr in SINTEF Ocean data, focused on samples [1000-1350] and [2000-2313].

In Figure 3.14 the areas [1000-1350] and [2000-2313] in the sample have been put in focus to better evaluate the ability to predict residuary resistance in the SINTEF Ocean data sample. Both networks predict Cr to a large extent, but the network trained for 996 epoch shows far better ability to predict correct results, than the network trained for 19 epochs. The latter shows a correct representation of the general behaviour but has a tendency of predicting too low or too high values for Cr. The network trained for 996 epochs does, on the other hand, predict the residuary resistance coefficient perfectly which can be seen in Figure 3.14 where the black dots hit the centre of the blue circles for almost every sample point.

The results presented above are as expected for the two networks. Training for 977 more epochs should have a significant effect on the ability to predict Cr, especially when the tested data-set is the same as training and validation data-sets together. This is clearly visible by the MSE values in Table 3.14, where the network trained for 996 epochs produce a prediction error of only 0.0035. The network trained for 19 epochs yields an estimation error of 0.1135, which is half the error obtained when doing validation analysis with the Southampton data-set in Section 3.4. This network shows good generalisation abilities and predicts both data-sets to a large extent.

3.4.4 Systematic Testing of Input Parameters

In this section, each of the input parameters is set equal to zero and SINTEF Ocean sample means, to see which ones are most crucial for performance in the estimation of residuary resistance coefficients and if it is possible to replace unknown model parameters and still get decent performance. The effect of setting Sb/S-ratio equal to zero and SINTEF Ocean sample mean when predicting Southampton data, was checked in the previous section but will be checked again for predicting SINTEF Ocean data. Detailed evaluation of performance for the networks will be presented for two parts of the data sample: samples 1000-1350 and 2000-2313. This is for illustration purposes and for easier comparing performance of the networks, as presenting the whole result gets messy due to large amounts of data. Comparison of the full data sample is however presented in Appendix B.3. Mean squared errors between predicted- and actual Cr-values are presented in Table 3.15.

Table 3.15: Mean squared error between predicted and correct Cr when different input parameters to the neural networks are set to zero and sample mean.

Missing Parameter	Network Trained for 996 Epochs		Network Trained for 19 Epochs	
	Param: 0	Param: mean	Param: 0	Param: mean
Fn	102.2505	2.0127	9.7834	1.9405
B/T	301.6845	33.7915	0.3252	0.1999
$L/\nabla^{1/3}$	1470.6	12.7429	726.9529	2.7699
$S/\nabla^{2/3}$	474.4336	7.8647	38.1246	0.3147
Sb/S	110.1305	14.7787	2.0901	0.8493
$s2/Lwl$	70.8783	3.7142	1.1424	0.1401

Testing Input Parameter: Fn

Figure 3.15 shows how the networks are behaving when Froude number are missing as an input parameter. Table 3.15 indicated large estimation errors, which are reasonable as the neural network does not know which speeds the design is experiencing. Speed and relative speed is an important factor when estimating resistance, as presented in Section 2.1, therefore poor estimation ability was expected.

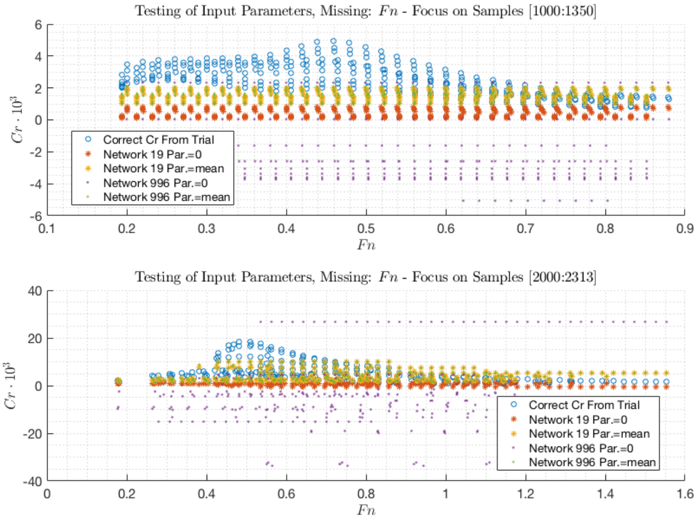

Figure 3.15: Network ability to predict Cr in SINTEF Ocean data, when input parameter Fn is missing. Results focused for illustration purposes.

Testing Input Parameter: B/T

Figure 3.16 shows how the neural networks are behaving when B/T is missing from the input. The network trained for 996 epochs over- and under-estimated the residuary resistance coefficient for most data samples, which results in the large MSE values presented in Table 3.15. The largest errors are observed for $Fn > 0.5$, while the network trained for fewer epochs handles the missing parameter significantly better.

Testing Input Parameter: $L/\nabla^{1/3}$

Figure 3.17 shows how important the $L/\nabla^{1/3}$ input parameter is for the neural network, and hence how important the parameter is for estimating resistance from these catamaran models. Both networks miss the general behaviour of the residuary resistance coefficient and significant over-estimation is observed. Large mean squared errors are produced, where the network trained for 19 epochs and $L/\nabla^{1/3} = mean$ shows the best prediction.

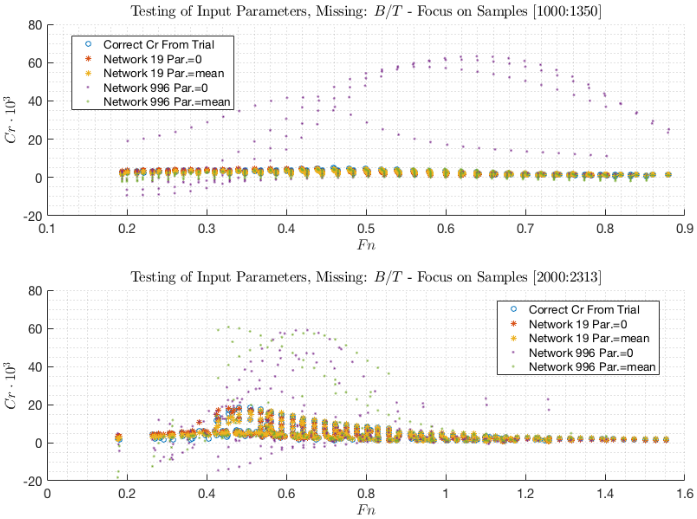

Figure 3.16: Network ability to predict Cr in SINTEF Ocean data, when input parameter B/T is missing. Results focused for illustration purposes.

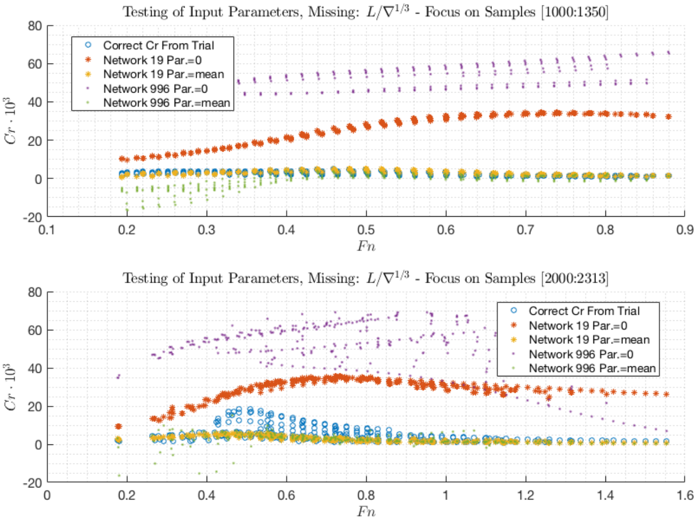

Figure 3.17: Network ability to predict Cr in SINTEF Ocean data, when input parameter $L/\nabla^{1/3}$ is missing. Results focused for illustration purposes.

Testing Input Parameter: $S/\nabla^{2/3}$

Figure 3.18 shows how the networks react when correct $S/\nabla^{2/3}$ is missing from the input. Especially the networks where $S/\nabla^{2/3} = 0$ produces large prediction errors by both networks, as indicated in Table 3.15. The network trained for 19 epochs predicts the residuary resistance coefficient significantly better than the network trained for 996 epochs, especially when $S/\nabla^{2/3} = mean$.

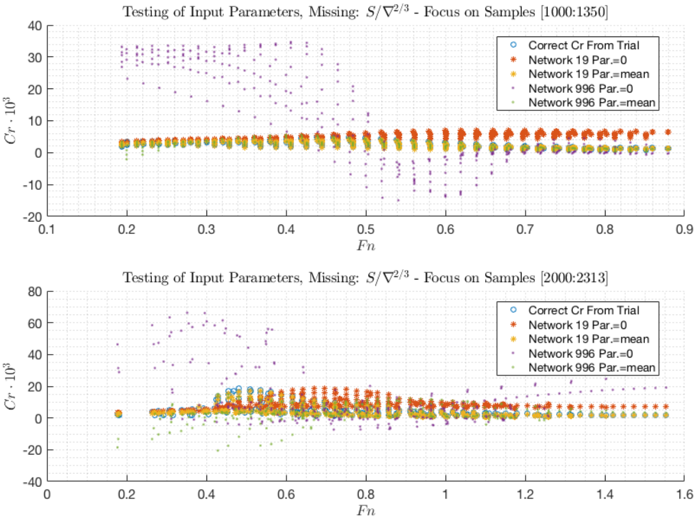

Figure 3.18: Network ability to predict Cr in SINTEF Ocean data, when input parameter $S/\nabla^{2/3}$ is missing. Results focused for illustration purposes.

Testing Input Parameter: Sb/S

Figure 3.19 shows how missing the Sb/S-value affects the performance of the networks. The network trained for 996 epochs creates significant under-estimations and over-estimations of the residuary resistance coefficient, especially when $Sb/S = mean$. The under-trained network with $s2/Lwl = mean$ appears to give the most accurate estimation for this sample, which gives a mean squared error of only 0.8493. An MSE of 70.8783 is produced by the network trained for most epochs when $Sb/S = mean$.

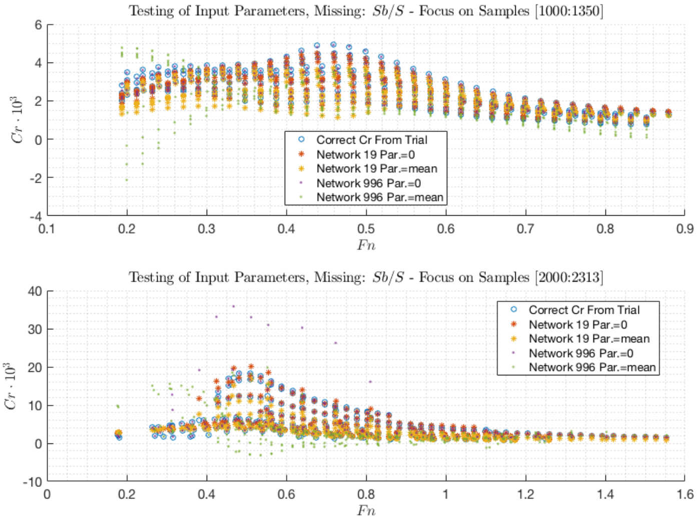

Figure 3.19: Network ability to predict Cr in SINTEF Ocean data, when input parameter Sb/S is missing. Results focused for illustration purposes.

Testing Input Parameter: $s2/Lwl$

Figure 3.20 shows how missing the $s2/Lwl$-value affects the performance of the networks. Again, the network trained for 996 epochs creates significant over-estimations and shows especially poor ability to predict Cr. The under-trained network with $s2/Lwl = mean$ produces the best predictions, with a mean squared error of only 0.1401. this is almost as good as when it was tested on the whole data-set with all input parameters, seen in Table 3.14.

From the tests carried out in this Section, it is clear that the network trained for 996 epochs suffers when input parameters are missing or estimated. Significant over-estimation and under-estimation were observed, and mean squared errors in the range of 1470.6-2.0127 was produced. The network trained for fewer epochs performs best in all cases where input parameters are missing. Setting missing values equal to the sample means appears to yield the best performance for all situations. The network trained for 996 epochs should be used with care, and the network trained for 19 epochs appears to be most suitable for predicting residuary resistance for fast catamarans.

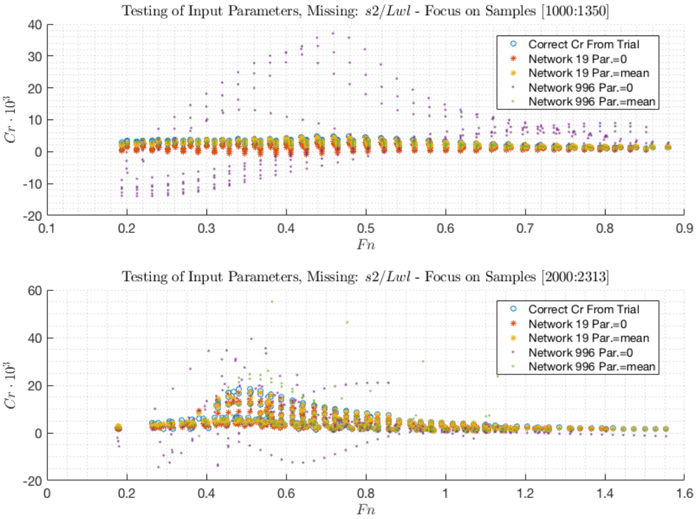

Figure 3.20: Network ability to predict Cr in SINTEF Ocean data, when input parameter $s2/Lwl$ is missing. Results focused for illustrative purposes.

Chapter 4

Discussion

In this chapter topics presented earlier in this thesis are discussed, and the possible consequences of the actions taken and choices made on the produced empirical models will be evaluated. An important topic is how well the available model data represents the design of fast displacement catamarans over all, and if the validation model data is within this category. The chosen method of constructing neural networks will be discussed and if other available methods could yield even better empirical models. Lastly, the two optimal neural networks are evaluated based on the results presented in the previous chapter.

4.1 Representability of the Available Model Data

The available model data given by SINTEF Ocean can be divided into two categories as presented before; old models which was used to develop CatRES, tested by MARINTEK from 1990 to 1997, and new models tested in the towing facility after from 1997 to 2017. The dataset with models tested before 1997 consists of 1082 samples from mostly passenger-, car ferries and cargo catamarans with a few models being fishing vessels for line fishing. The new models are mostly ferries and some other vessels, with Lpp between 19-130 meters and a concentration around $Lpp = 40$ meters.

The types of catamarans tested in the towing facility by SINTEF Ocean has not changed dramatically over the years, as the vessels from 1990-1997 and 1997-2017 are of the same types. Having an homogeneous sample concerning vessel types, yields trained neural networks specialised in predicting resistance for these types of vessels. The neural network would however not be suitable for predicting residuary resistance for other types of catamarans. Because most models tested by SINTEF Ocean are of ferry-catamaran-type with some additional designs, it is reasonable to assume the models tested in the future will be similar to these. By including recent model trials from ferries and some other vessels, 1231 data samples are added to the original dataset of 1082 samples, creating a dataset of 2313 data samples. By comparing vessel parameters for the vessels presented in Appendix A.1, some tendencies are found. New designs appears to have lower slenderness-ratio and higher Lwl/Bwl-ratio than the older designs. Higher displacement is a characteristic for planing vessels, as the hull is lifted partly out of the water when travelling at higher speeds. When screening the models together with SINTEF Ocean, planing vessels was however filtered out, as displacement and semi-displacement catamarans was desired. Lower B/T ratios are observed for some of the newer vessels designed with water-jet propulsion. This indicates either wider- or shallower hulls in the new design, but the general tendency in B/T value is similar for older and newer designs. Hull separation-ratio appears to be consistent for older and newer designs. $s2/Lwl$ are found to be in the range of $[0.10 - 0.37]$, where most models regardless of old or new design, have a hull-separation-ratio around 0.2. The neural networks trained on SINTEF Ocean data, set consisting of 46.8% old data samples and 53.2% new data samples, would be able to predict residuary resistance coefficients for both elder and newer designs. The validity range of the trained networks are presented in Table 4.1.

Table 4.1: Parameter validity range for neural networks.

Parameter	Valid Data Range
Fn	[0.0920-1.5540]
Lwl/Bwl	[1.9800-14.8310]
B/T	[1.1538-12.0650]
$L/\nabla^{1/3}$	[5.3100-10.8400]
$S/\nabla^{2/3}$	[7.8200-12.4000]
Sb/S	[0-0.0216]
$s2/Lwl$	[0-0.3782]
$Cr \cdot 10^3$	[0.1210-18.3810]

An evaluation of the trained networks was carried out using model data from Southampton presented in Appendix A.2. The purpose was to evaluate the generalisation ability of the neural networks and how they performed outside their valid parameter range. When using independent datasets for validation purposes of neural networks, one must be sure of the validity of the independent dataset as well. The Southampton model are of round bilge form with transom sterns, and are derived from the National Physical Laboratory

(NPL) round bilge series. These mono-hulls was mounted together at desired $s2/Lwl$ ratio, into catamaran form [Molland et al., 1995]. The round bilge hull form is characterised by straight entrance waterlines, rounded afterbody sections and straight buttocks line terminating sharply at a transom [Bailey, 1976]. By looking at the hull lines for the models in the SINTEF Ocean data sample, most models appears to have round bilge hulls. However, assembling two mono-hulls together to make a catamaran design, may yield other flow characteristics than a modern catamaran design, made for the purpose of being an optimal catamaran. This dataset is on the other hand, one of the few datasets available to the public, and hence one of the few available dataset for verification of the neural networks. Therefore the performance of the neural network on the SINTEF Ocean dataset is prioritised, while the performance on the Southampton dataset is used as a guideline to reveal overtraining.

The effect of erroneous data in neural network training, would be severe if the amount of erroneous data is significant in the sample. In this work unknown parameters are set equal to zero, which mislead the network during training, breaking patterns in the data and creating noise. When the parameter range in the SINTEF Ocean data sample was presented in Section 3.1.3, the possibility of using data samples without any missing values was evaluated. This lead to a reduction in sample size of 46.78%, and exclusion of most resistance tests of older catamarans. Significantly better performance was observed of the network trained on the reduced dataset, as presented in Section 3.2.3, where the best test mean squared error was ten times better than for the network trained on the full dataset. However, after discussing with co-supervisor Hans Jørgen Rambech, it was decided to use the network with largest validity range and lower performance. One must bear in mind that the empirical residuary resistance estimation method is to be used in the early stage of the design process. The model should therefore be applicable for a wide range of model parameters with as large validity range as possible, even if this goes on expense of accuracy.

The dataset provided by SINTEF Ocean appears to be representative for fast displacement catamarans designed as ferries and transport catamarans. By looking at the parameter ranges, the design has evolved from the tests carried out from 1990 to 1997, to the ones carried out from 1997 to 2017. A total number of 52 vessels are included, making several parameter combinations and designs present in the dataset. Independent resistance tests carried out by Molland et al. [1995] was on the other hand tested for the sake of testing the effect of design parameters on catamaran resistance. Generalisation ability can, however, be assessed using this dataset, which has been done in this thesis.

4.2 Neural Networks Compared to Polynomial Curve Fitting

In Section 2.3, empirical models based on polynomial curve fitting was presented together with models based on machine learning. The opacity concept of different empirical models was outlined, where the white box model is a model based on an analytic mathematical expression describing the phenomena, and the black box model is a model without a physical relationship between the phenomena and the mathematical model describing the phenomena.

Most current empirical resistance prediction models are of the polynomial curve fitting-type, such as CatRES and Molland et al. [1995] method. When choosing to fit polynomials to resistance data, coefficients can be found through regression based on model parameters, while the ”driving” parameter in the model can be another parameter. This is the case in CatRES method presented in Section 2.2.1. The driving parameter is the slenderness ratio $L/\nabla^{1/3}$, and correlation on C_R and C_W can be used for correction. Methods 1 to 5 makes it possible to estimate resistance both for designs where $L/\nabla^{1/3}$ is the only known parameter, but also if other parameters such as $S/\nabla^{2/3}$, B_{demi}/T and s/L are known.

When neural networks are used to develop empirical methods, the input parameters to the finished model needs to be the same as the input parameter used during training. However, during the first stage of the design process of catamarans parameters such as wetted area and area of wetted stern may not be known. A way of bypassing this problem is to estimate unknown input-parameters to the empirical model. This was analysed in Section 3.4.2, where the missing Sb/S-ratio in the Southampton dataset was set equal to zero and equal to mean of the training data sample. Both solutions gave reasonable estimates, but variation between the estimated values was significant at times. This would be a defect in the empirical resistance estimation model, as stable performance is desired.

4.3 Matlab Compared to Other Available Tools

Throughout this work Matlab has been used for data validation, neural network training and post-processing, but there are other suitable software available. Yegulalp [2017] did a short review on available deep learning libraries and how the larger technology companies are developing their software to take the growing market, and a selection of these are presented next.

Microsoft Cognitive Toolkit is developed by Microsoft and provides a Java application programming interface (API), allowing more direct integration with other processing framework. The framework supports both Python and C++, and claims to be both faster and more accurate than the Google flagship: TensorFlow. Microsoft are also offering cloud-based computing and user friendly software, which makes the methods available to other users than experienced machine learning specialists [Microsoft Research, 2018].

TensorFlow is the open source software library for high performance numerical computation, developed by the Google Brain team [TensorFlow, 2018]. The framework is more low-level programming-vise, and can be used with Python. In order to make TensorFlow and other machine learning frameworks more user-friendly the **Keras** high-level neural networks API was developed. Keras lowers the required programming knowledge to use TensorFlow significantly, and has made advanced machine learning available to the public [Keras, 2018].

There are multiple advantages of choosing one of the frameworks mentioned above, as both are in continuous development, testing and use of some of the largest technology companies in the world. Another advantage is the ability of cloud computing, in Azure Cloud Platform and Google Cloud Platform respectively. Cloud computing makes it possible to buy computational power from high performance computing facilities when needed, instead of investing in expensive hardware. EY (former Ernst and Young) made a survey on the maturity of cloud computing with focus on Norway for 2018, and made the following conclusion: cloud computing is too immature today, but will play an important role in the future. Machine learning is a relative small part of cloud computing with 17% only, but it is expected to increase drastically in the future [Mjaanes and Slaata, 2017].

By choosing Matlab over the frameworks presented above, better performing networks may have been missed. The author feels a lot more comfortable with programming in Matlab, as this tool have been frequently used throughout his study. Getting on the required programming level in either Python or C++ to make use of the mentioned tools, would have been time consuming and profit was not guaranteed. Conducting an extensive parameter study of neural network design in Matlab was prioritised over learning other software in this project.

4.4 Effect of Training and Chosen Parameters

Training is an important part of neural network design as presented in Section 2.4.4. Pit-falls when designing and training neural networks were presented in Section 2.4.5, with emphasis on over- and undertraining and over- and underfitting. To address these issues, the available dataset was divided into training-, validation- and test datasets and the performance of the network tested on these datasets was monitored during training. Some measures for optimising network training is included in Matlab such as early-stopping, regularisation and backpropagation, which was presented in Section 2.4.4. Additionally, the generalisation ability of two networks was checked by comparing their predicted residuary resistance coefficient against the measured residuary resistance from model tests carried out in Southampton.

The network performing best is the one presented in Section 3.3, with four hidden layers, a node distribution if [10-11-11-10] and trained for 996 epochs. The number of training epochs is large, and indicates the network being overtrained. Test performance was however best at 996 epochs with a mean squared error of $3.461 \cdot 10^{-3}$ and a training performance of $3.725 \cdot 10^{-3}$. Because best test performance is observed at this point, 996 should be the point where this model is at its most generalised. When testing the model to the Southampton dataset, significantly high mean squared errors was observed. During the study of missing input parameters in Section 3.4.4, the ”overtrained” network produced significant prediction errors, and proved to be practically unusable. This would be a clear sign of overtraining and lack of generalisation ability.

When training the same network for fewer epochs and looking for the lowest error when comparing with Southampton data, a network trained for 19 epochs performed best. The results presented in Table 3.13 indicates a better ability of predicting the Southampton dataset. When testing the ”undertrained” network on the whole SINTEF Ocean dataset, a mean squared error of 0.1135 is found between the predicted and the correct Cr values.

The gain of training the network 977 more epoch was clearly visible in Figure 3.14, where the ”overtrained” network predicted the residuary resistance coefficients perfectly, and the ”undertrained” network found the general behaviour, but generally missed the correct values in the sample. However, the ”undertrained” network performed better overall, than the ”overtrained” network which missed the data in the Southampton data sample entirely.

Another issue, which is often addressed when designing neural networks is correlation within the dataset. The optimal parameter-set for input values was found to be set three from Table 3.6. Table 4.2 shows the linear correlation between the parameters in SINTEF Ocean data sample used for training, validation and testing.

The highest positive correlation is between Sb/S and Fn of 0.3230, and the highest negative correlation is between $L/\nabla^{1/3}$ and Sb/S of -0.6721, which is a strong negative correlation. Positive linear correlation is on the other hand rather low, but the parameter in the dataset cannot be categorised as uncorrelated. When the short list with input-parameters was presented in Section 3.1.4, a comment was made on parameter-set five in Table 3.6

Table 4.2: The linear correlation between parameters in SINTEF Ocean data sample.

	Fn	B/T	$L/\nabla^{1/3}$	$S/\nabla^{2/3}$	Sb/S	$s2/Lwl$	$Cr \cdot 10^3$
Fn	1	0.2868	-0.3352	-0.1109	0.3230	0.1456	-0.3869
B/T	Sym	1	-0.1414	0.2388	0.3183	-0.0618	-0.2555
$L/\nabla^{1/3}$	Sym	Sym	1	0.2598	-0.6721	-0.2155	-0.2864
$S/\nabla^{2/3}$	Sym	Sym	Sym	1	0.1209	-0.1981	-0.2135
Sb/S	Sym	Sym	Sym	Sym	1	0.1310	-0.1761
$s2/Lwl$	Sym	Sym	Sym	Sym	Sym	1	0.1476
$Cr \cdot 10^3$	Sym	Sym	Sym	Sym	Sym	Sym	1

being a typical neural network test set which is given all available data. These kinds of sets often yield good performance as the network gets lots of data for training. This parameter-set did however not perform best of the proposed ones. By looking at the linear correlation of the boolean value ($Jet/Prop$) and Lwl/Bwl to the residuary resistance coefficient Cr, correlations of -0.2486 and -0.1789 are observed respectively. It would have been interesting to make an additional parameter-set study, where parameter-set five from Table 3.6 are used as a basis, and the parameters are excluded from the set in order of lowest correlation to Cr. So the order of exclusion would be: $s2/Lwl$, Sb/S, Lwl/Bwl, $S/\nabla^{2/3}$, ($Jet/Prop$), B/T and lastly $L/\nabla^{1/3}$. Maybe a better performing parameter-set would be found through this analysis.

The effect of correlation between input parameters, and between input- and output parameters, on neural networks, have been analysed by researchers. Halkjær and Winther [1997] found a correlation between input parameters to increase the time to convergence for the neural networks. Wendemuth et al. [1999] found a significant increase in required capacity when there was a correlation between patterns in the network, but also a decrease in the error and then also an increased performance of the networks. Cook [1995] states that neural networks, in general, are "smart" enough to detect the linear correlation between parameters, and hence take this into account when trained. He also finds the effect of correlation between input parameters and output to affect the network for severely strong correlation. The correlation seen in this data sample cannot be categorised as the latter, and the neural network training function is assumed to find and act on correlation. Acceptable performance has been shown through training, and poor performance due to correlation appears to be absent.

The two input parameters which affected the networks ability to predict Cr the most, from Section 3.4.4, are also the two with the highest negative correlation to Cr from Table 4.2, namely Fn and $L/\nabla^{1/3}$. The networks appear to have the ability to predict the linear correlation between parameters, or it may be a coincidence that missing the most correlated parameters creates the largest prediction errors.

The effect of setting missing values in the datasets equal to zero when training neural networks have been discussed earlier. An important consideration is how this has affected the parameter study, and how the optimal parameter-set and algorithm was found. Even though almost half the samples have Sb/S set equal to zero due to missing data, parameter-set three from Table 3.6 was performing best in the analysis. In theory, this is also the noisiest dataset. When data samples with Sb/S was excluded from the data sample in Section 3.2.3, parameter-set two performed far better than parameter-set three, even though parameter-set three was made less noisy. For the tests with reduced datasets, parameter-set five was expected to be the best as more data often yield the best performance in machine learning. This was however not the result, even though the overall performance of dataset five is good which can be seen in Figure 3.5. A significant amount of networks have been tested in this study, and the optimal parameter-set and network, presented in Table 3.11 was found.

4.5 Evaluating the Two Optimal Neural Networks

Two networks presented in Table 3.14, was trained and tested on the independent dataset from Southampton. The results from the verification tests carried out in Section 3.4, indicated poor validation performance for the network trained for 996 epochs, while the network trained for 19 epochs performed far better. In the Southampton dataset, the wetted-stern-wetted-surface-area-ratio Sb/S was not recorded for the models. Comparing the effect of estimating the missing input for this dataset was carried out, setting $Sb/S = 0$ and $Sb/S =$ sample mean to find which gave the best prediction ability for the network.

The Southampton dataset contains models with parameters outside the validity range of the networks trained on SINTEF Ocean data, as presented in Table 3.12. For data samples inside the validity range and data samples exceeding the validity range, using $Sb/S = 0$ gave the best performance, which can be seen in Table 3.13. The results presented in Figures 3.10 to 3.13, shows how the "undertrained" network performs good for analysis when data is within and outside network validity range. It is capable of predicting the general behaviour in the independent data sample but is not able to predict the residuary resistance perfectly. The "overtrained" network performed poorly, which can be seen in Figures B.3. Mean squared error between predicted and actual residuary resistance coefficient was in the range of $55.29 - 126.89$ for the "overtrained" network as presented in Table 3.13, while it was between 0.21 and 0.85 for the "undertrained" network. Indicating a far better generalisation for the latter.

After the analysis of the Southampton data sample, the two networks were tested on the SINTEF Ocean data sample. Good performance was expected as this is the training, validation and test datasets combined. The ”overtrained” network predicted the SINTEF Ocean data perfectly as visible in Figure 3.14 giving a mean squared error of 0.0035, while the ”undertrained” network gave a prediction error of 0.1135.

A missing parameter analysis was carried out in Section 3.4.4 to simulate the case when input-variables are unknown in the early design stage. Setting missing input values equal to SINTEF Ocean sample mean yielded far better estimation ability for both networks, than setting missing input-parameters equal to zero. Considering the result of the analysis carried out in this thesis, using SINTEF Ocean sample mean is recommended in cases where input values are unknown. This requires the tested design to be similar to the vessels in the SINTEF Ocean dataset, of course.

Through verification tests, the network trained for 996 epochs shows signs of being significantly overtrained. Overtrained networks are characterised by their lack of generalisation ability and low accuracy for data samples except for the training dataset, which leads to an unpredictable and poor empirical model. In order to decide on which model to use, SINTEF Ocean could use both models for some time when carrying out new resistance tests for model catamarans, and compare the predicted residuary resistance coefficient from the empirical models to the results from the towing tests. After the testing period, one can find which yields the better prediction of Cr from fast catamarans. If this is not possible, using the network trained for 19 epochs is recommended.

Chapter 5

Conclusion

In this thesis theory concerning resistance of fast catamarans, machine learning and artificial neural networks have been presented, before two currently used empirical resistance prediction methods for fast catamarans was reviewed. The main task in this thesis has been to develop a new empirical resistance prediction method for fast catamarans, as the current method used by SINTEF Ocean, named CatRES, yields conservative estimates. This method was presented in Section 2.2.1 and is of polynomial curve fitting type. In this thesis, machine learning have been investigated, in order to see whether such an approach would give an even better performing empirical resistance prediction method.

After digitising model resistance data provided by SINTEF Ocean, empirical models made with artificial neural networks were designed and tested. The resistance data could be divided into two categories: ”old data” which is the data-set CatRES is based on tested by MARINTEK in the period 1990-1997, and ”new models” which consists of model trials carried out in the period 1997-2017. The vessels can be divided into two categories: vessels designed with propeller- and water jet propulsion system. Data range was a central issue, as the data range in the training sample of machine learning algorithms defines the validity range for the empirical model. The validity range is presented in Table 4.1. Applying artificial neural networks proved to be a suitable approach and good prediction abilities were found. In Section 3.2 a parameter study was carried out to find the optimal input-parameters to neural network training in Matlab using the Neural Network Toolbox. The optimal input parameters proved to be; Training function *trainbr*, four hidden layers with a node distribution of [10-11-11-10] and parameter set tree from Table 3.6.

The optimal neural network training epochs were 996, but the network showed clear signs of overtraining during the evaluation process. An independent data-set with model parameters and measured residuary resistance coefficients from Molland et al. [1995], was used to verify the performance of the networks. A significant mean squared error was produced between the predicted residuary resistance coefficient Cr, and measured ones for the best performing network from the parameter study. A new network was trained for fewer epochs with same network characteristics as before, being continuously validated with the Southampton data sample to see the true generalisation ability of the network. The optimal number of training epochs proved to be 19, as shown in Section 3.4. Both networks were thoroughly tested to evaluate generalisation ability, performance when validity parameter range is exceeded and how the performance is when input parameters are missing. These tests provided more evidence of the network trained for 996 epochs being overtrained. When simulating missing inputs and setting the missing values equal to zero and SINTEF Ocean sample mean, the network produced high estimation errors, as presented in Table 3.15.

By carrying out an input-parameter correlation analysis in Section 4.4, the effect of correlation between inputs and between input- and output parameters were presented and discussed. Research indicated that linear correlation between parameters did not affect the networks ability to obtain good performance, and because good performance has been observed for the networks, problems with a linear correlation between parameters are assumed to be unexciting. Input parameters Fn and $L/\nabla^{1/3}$ was also found to be the most important input parameters to the model to obtain accurate residuary resistance estimations.

After designing, training and evaluating the two neural networks, the network trained for 19 epochs is found to give the best prediction of residuary resistance. The network is converted into a Matlab function and delivered to SINTEF Ocean together with this report. The author recommends testing the empirical resistance model to future model tests in order to verify the generalisation ability of the empirical model. Actual model data and the actual working Matlab function is not enclosed to this thesis, due to the non-disclosure agreement between the author and SINTEF Ocean. An example of how the function would look like is presented in Appendix C, where weights and biases are made anonymous. After a certain test period, the empirical model can be implemented into a larger software system.

Chapter 6

Recommendations for Further Work

Throughout this thesis, several neural networks and network parameters have been tested to find the optimal network. Below are some thoughts on how the method can be improved and validated further.

Increasing training- and validation data-sets. “In machine learning, is more data always better than better algorithms?” is one of the most famous quotes and questions in machine learning, by Google’s Research Director Peter Norvig. The effect of increasing a data-set several times, but with low-quality data, is discussed in an article by Halevy et al. [2009]. The concrete example is of the Brown Corpus containing one million English words, where Google have made a corpus with an increased number of words to trillions from unfiltered web pages. This data-set is significantly larger, but it is also unfiltered and contains spelling errors and grammatical errors. Halevy et al. [2009] says unsupervised learning will be better with larger data-sets, even if the data contains errors. For the case investigated in this thesis, high-quality data is essential. This is because supervised learning is used, and the network is misguided during training if the data-sets contains lots of poor data samples. Research by Shaikhina and Khovanova [2017] shows on the other hand how ”similar” data can be added to the training data-set to obtain better performing neural networks. By adding large amounts of fracture data for concrete, to their data-set with fracture characteristics for human bones, they obtained a significantly better performance for their model describing fracture in bones. Adding model trial data from monohulls could yield a better performing model, or this could at least be tested and compared to the model presented in this thesis.

Another possibility is to contact other test facilities and ask for their resistance data for fast catamaran models. This may, however, be a non-feasible proposal, as the resistance data bank is one of their most valuable and precious items, built up over years of testing models. This has been done in marine applications before, such as when NCE Seafood [2017] managed to get data from seven fish farming companies in their project on sea lice management. If the industry is interested in a common empirical residuary resistance estimation method, is another discussion.

Comparing the proposed empirical model from this thesis to models developed with other software as discussed in Section 4.3, is desirable to find out whether the proposed model is the optimal one. Several tools are available as open source frameworks continuously developed by some of the finest technology companies in the world, and using these could provide better performing empirical models. Regardless whether other empirical models are developed or not, the proposed empirical resistance prediction method should be tested, in order to verify its validity in day-to-day-use before it is implemented into a larger software system.

Bibliography

Abu-Mostafa, Y. S., Magdon-Ismail, M., and Lin, H.-T. (2012). *Learning from data : a short course.* AMLbook, S.l.

Adams, R. P. and Gorman, K. (2015). Podcast: Talking machines. Online url: https://www.thetalkingmachines.com. Season One, Episode Sixteen.

Amdahl, J., Endal, A., Fuglerud, G., Hultgreen, L. R., Minsaas, K., Rasmussen, M., Sillerud, B., Sortland, B., and Valland, H. (2013). *TMR4105 - Marin Teknikk Grunnlag Kompendium.* Marin Teknisk Senter, 4th edition.

Bailey, D. (1976). *The NPL High Speed Round Bilge Displacement Hull Series: Resistance, Propulsion, Manoeuvring and Seakeeping Data.* Maritime technology monograph. Royal Institution of Naval Architects.

Beale, M. H., Hagan, M. T., and Demuth, H. B. (2017). *Neural Network Toolbox: User's Guide.* MathWorks. Url: https://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Information Science and Statistics. Springer.

Cook, N. D. (1995). Correlations between input and output units in neural networks. *Cognitive Science*, 19:563–574.

de Pina, A. A., da Fonseca Monteiro, B., Albrecht, C. H., de Lima, B. S. L. P., and Jacob, B. P. (2016). Artificial neural networks for the analysis of spread-mooring configurations for floating production systems. *Applied Ocean Research*, 59:254 – 264.

Famili, A., Shen, W.-M., Weber, R., and Simoudis, E. (1997). Data preprocessing and intelligent data analysis. *Intelligent Data Analysis*, 1(1):3 – 23.

Fathi, D., Ringen, E., Alterskjaer, A., and Berget, K. (2012). *ShipX Ship Speed & Powering Plug-In.* SINTEF Ocean, Marine Technology Centre, Otto Nilsens Vei 10, 7052 Trondheim.

Foresee, D. F. and Hagan, M. T. (1997). Gauss-newton approximation to bayesian learning. *Proceedings of the 1997 International Joint Conference on Neural Networks.*

Gauch, H. G. (2003). *Scientific method in practice.* Cambridge University Press, Cambridge.

Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE intelligent systems.*, 24(2):8–12.

Halkjær, S. and Winther, O. (1997). The effect of correlated input data on the dynamics of learning. In *Advances in neural information processing systems*, pages 169–175.

Insel, M. and Molland, A. F. (1992). An investigation into the resistance components of high speed displacement catamarans. *The Royal Institution of Naval Architects.*

ITTC (1987). Ittc - recommended procedures and guidelines: Performance, propulsion 1978 performance prediction methods. *International Towing Tank Conference*, 18:266–273.

Keras (2018). Keras: The python deep learning library. Online Url. https://keras.io.

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K. R. (1998). *Efficient BackProp*, pages 9–50. Springer Berlin Heidelberg, Berlin, Heidelberg.

MacKay, D. J. C. (1992). Bayesian interpolation. *Neural Computation*, 4(2):415–447.

Mathworks INC. (2018a). Mathworks documentation: Choose a multilayer neural network training function. Online User Manual. https://se.mathworks.com/help/nnet/ug/choose-a-multilayer-neural-network-training-function.html.

Mathworks INC. (2018b). Mathworks documentation: trainbr. Online User Manual. https://se.mathworks.com/help/nnet/ref/trainbr.html.

Mathworks INC. (2018c). Mathworks online user manual: ”choose neural network input-output processing functions”, ”divide data for optimal neural network training” and ”improve neural network generalization and avoid overfitting”.

Microsoft Research (2018). Unlock deeper learning with the new microsoft cognitive toolkit. Online Url. https://www.microsoft.com/en-us/cognitive-toolkit/.

Mjaanes, C. and Slaata, S. (2017). Ey norwegian cloud maturity survey 2018. current and planned adoption of cloud services. *EY Publications.*

Molland, A., Wellicome, J., and Couser, P. (1995). Resistance experiments on a systematic series of high speed displacement catamaran forms: variation of length-displacement ratio and breadth-draught ratio. ISSN 0140-3818.

Molland, A. F., Turnock, S. R., and Hudson, D. A. (2011). *Ship resistance and propulsion: practical estimation of ship propulsive power*. Cambridge University Press.

Murphy, K. P. (2012). *Machine Learning : A Probabilistic Perspective*. Adaptive Computation and Machine Learning. The MIT Press, Cambridge.

Najafi, A., Nowruzi, H., and Ghassemi, H. (2018). Performance prediction of hydrofoil-supported catamarans using experiment and anns. *Applied Ocean Research*, 75:66 – 84.

NCE Seafood (2017). Aquacloud - the use of artificial intelligence in sea lice management. Online Url: http://www.seafoodinnovation.no/article/213/AquaCloudThe_use_of_artificial _intelligence_in_sea_lice_management.

Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.

Rambech, H. J. (1998). Empirisk motstandsberegning for hurtiggående katamaraner. Project thesis, Norwegian University of Science and Technology.

Robards, S. W. and Doctors, L. J. (2003). Transom hollow prediction for high-speed displacement vessels. *FAST 2003*, Volume I, Section A1:19–26.

Sahoo, P. K., Salas, M., and Schwetz, A. (2007). Practical evaluation of resistance of high-speed catamaran hull forms—part i. *Ships and Offshore Structures*, 2(4):307–324.

Shaikhina, T. and Khovanova, N. A. (2017). Handling limited datasets with neural networks in medical applications: A small-data approach. *Artificial Intelligence in Medicine*, 75:51 – 63.

Shora, M. M., Ghassemi, H., and Nowruzi, H. (2017). Using computational fluid dynamic and artificial neural networks to predict the performance and cavitation volume of a propeller under different geometrical and physical characteristics. *Journal of Marine Engineering & Technology*, 0(0):1–26.

Steen, S. (2014). *Marin teknikk 3 - Hydrodynamikk: Motstand og propulsjon, Propell og Fiolteori*. Akademika Forlag.

Steen, S. and Minsaas, K. (2013). Lecture notes in tmr4220 naval hydrodynamics: Ship resistance.

Subramanian, V. A. and Joy, P. (2004). A method for rapid hull form development and resistance estimation of catamarans. *Marine Technology Society Journal*, 38(1):5–11.

TensorFlow (2018). About tensorflow. Online Url. https://www.tensorflow.org.

Tzafestas, S., Dalianis, P., and Anthopoulos, G. (1996). On the overtraining phenomenon of backpropagation neural networks. *Mathematics and Computers in Simulation*, 40(5):507 – 521. Neural Network/Neural Computing.

Wanjawa, B. and Muchemi, L. (2014). Ann model to predict stock prices at stock exchange markets.

Wendemuth, A., Opper, M., and Kinzel, W. (1999). The effect of correlations in neural networks. *Journal of Physics A: General Physics*, 26:3165.

Xuan, P. P., Kantimahanthi, K., and Sahoo, P. (2001). Wave resistance prediction of hard-chine catamarans through regression analysis. *In Proceedings of 2nd International Euro Conference on High Performance Marine Vehicles (HIPER'01)*, pages 382–394.

Yegulalp, S. (2017). Easier, faster: The next steps for deep learning. *Computerworld Hong Kong*. Name - Yahoo Inc; Google Inc; Copyright - Copyright Questex, LLC Jun 15, 2017; Last updated - 2017-08-29.

Appendix A

Model Parameter Ranges

A.1 Parameter Range: SINTEF Ocean Data

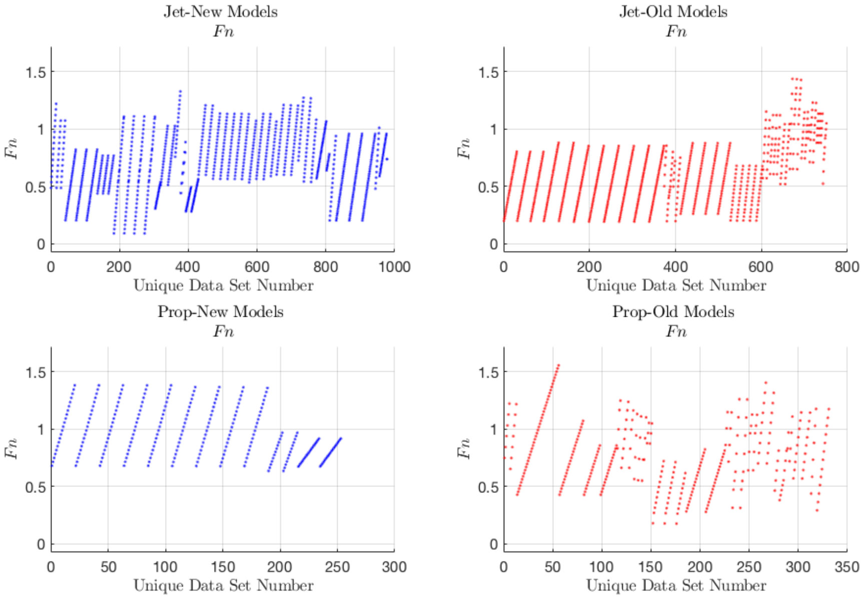

Figure A.1: Parameter range of Fn in data sets.

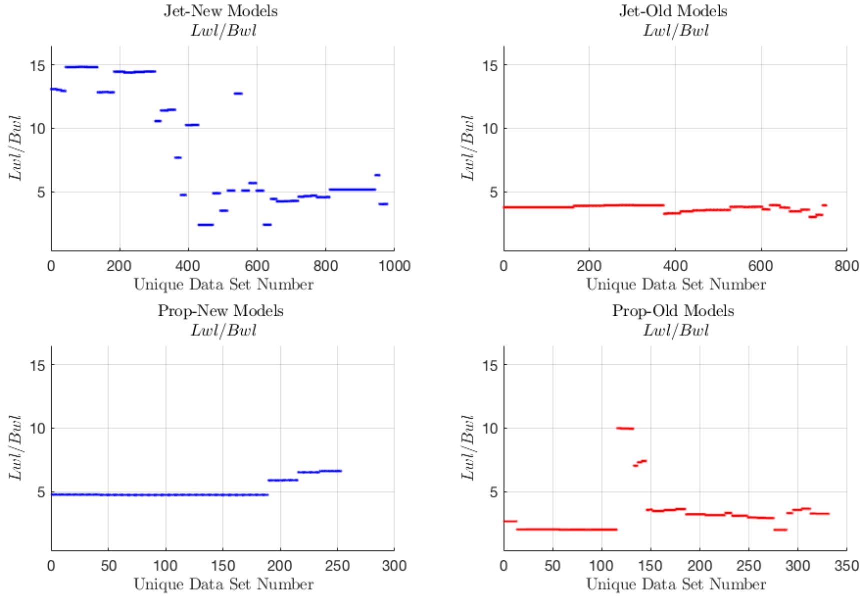

Figure A.2: Parameter range of Lwl/Bwl in data sets

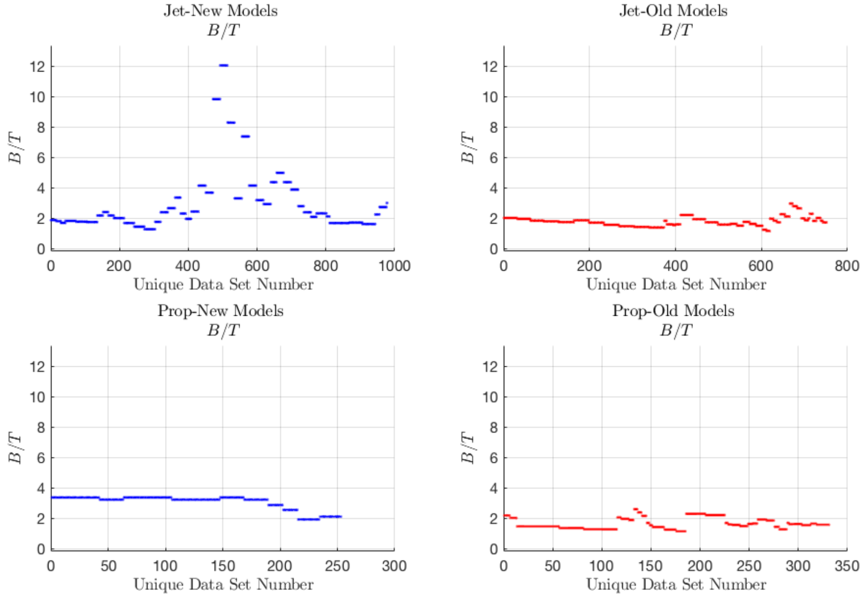

Figure A.3: Parameter range of B/T in data sets

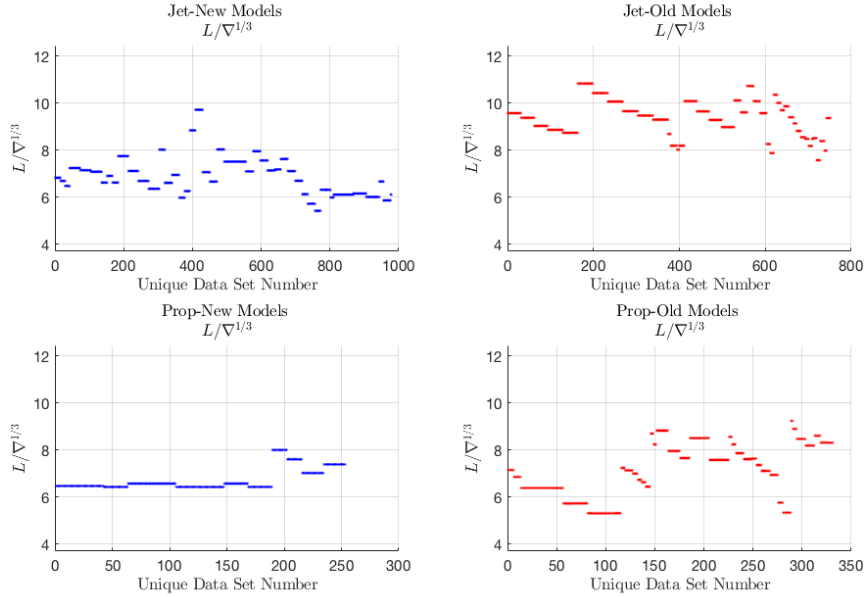

Figure A.4: Parameter range of $L/\nabla^{1/3}$ in data sets

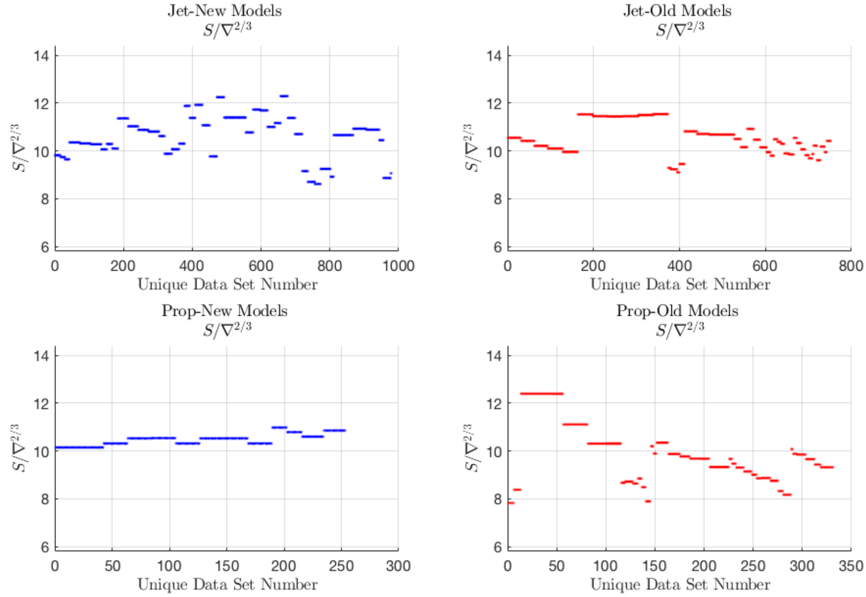

Figure A.5: Parameter range of $S/\nabla^{2/3}$ in data sets

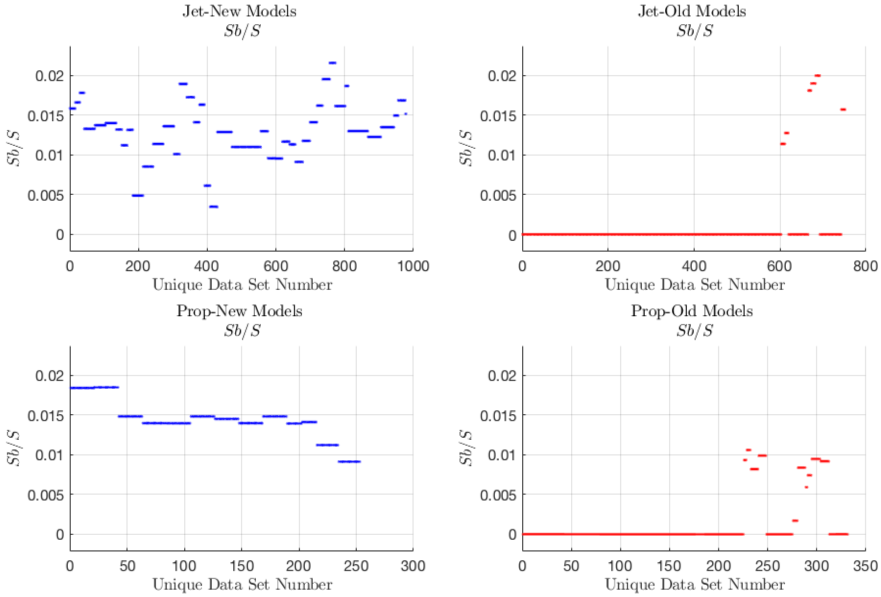

Figure A.6: Parameter range of Sb/S in data sets

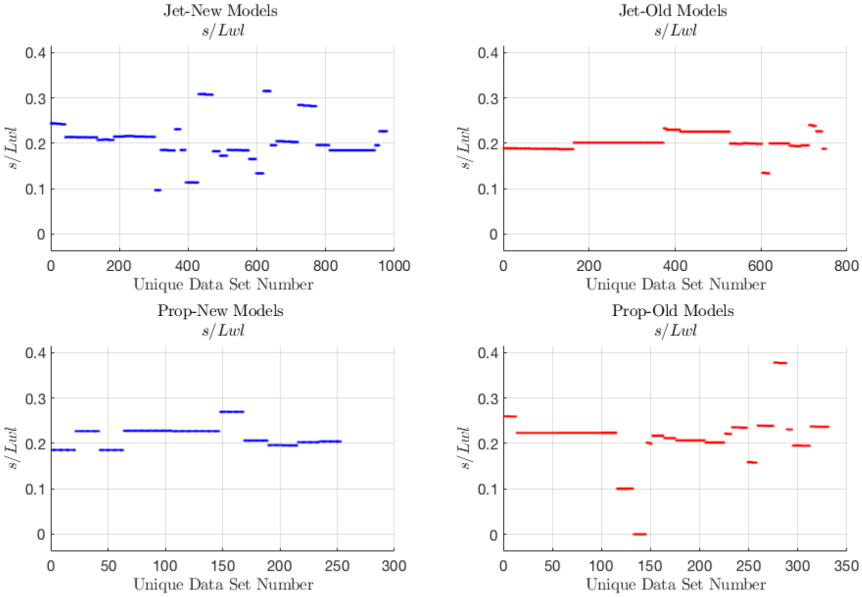

Figure A.7: Parameter range of s/Lwl in data sets

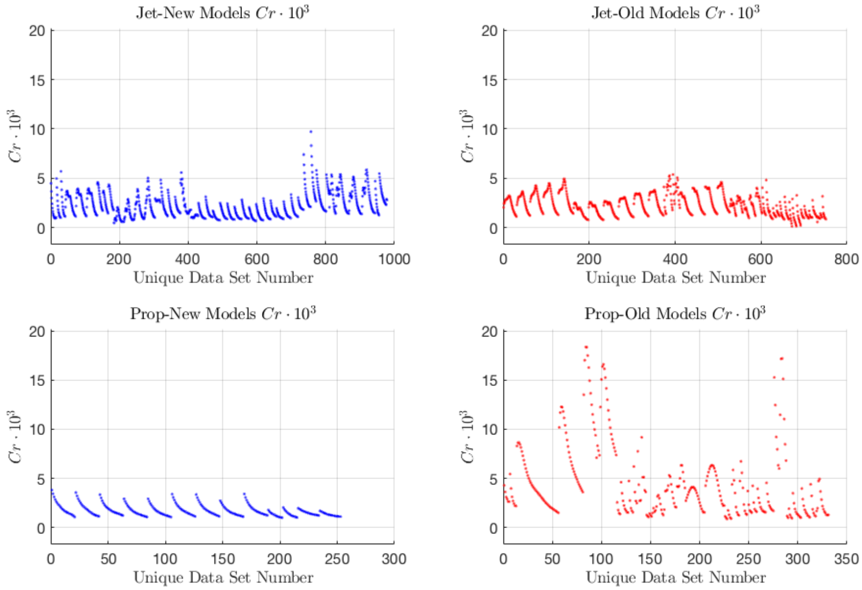

Figure A.8: Parameter range of $Cr \cdot 10^3$ in data sets

A.2 Parameter Range: Southampton Data

Table A.1: Model parameters in Southampton data set Molland et al. [1995]. All models are tested for hull separations $s/Lwl = 0.2$, 0.3, 0.4 and 0.5.

Model	$L[m]$	L/B	B/T	$L/\nabla^{1/3}$	C_B	C_P	C_M	$S[m^2]$	LCB[% from centre]
3b	1.6	7.0	2.0	6.27	0.397	0.693	0.565	0.434	-6.4
4a	1.6	10.4	1.5	7.40	0.397	0.693	0.565	0.348	-6.4
4b	1.6	9.0	2.0	7.41	0.397	0.693	0.565	0.338	-6.4
4c	1.6	8.0	2.5	7.39	0.397	0.693	0.565	0.340	-6.4
5a	1.6	12.8	1.5	8.51	0.397	0.693	0.565	0.282	-6.4
5b	1.6	11.0	2.0	8.50	0.397	0.693	0.565	0.276	-6.4
5c	1.6	9.9	2.5	8.49	0.397	0.693	0.565	0.277	-6.4
6a	1.6	15.1	1.5	9.50	0.397	0.693	0.565	0.240	-6.4
6b	1.6	13.1	2.0	9.50	0.397	0.693	0.565	0.233	-6.4
6c	1.6	11.7	2.5	9.50	0.397	0.693	0.565	0.234	-6.4

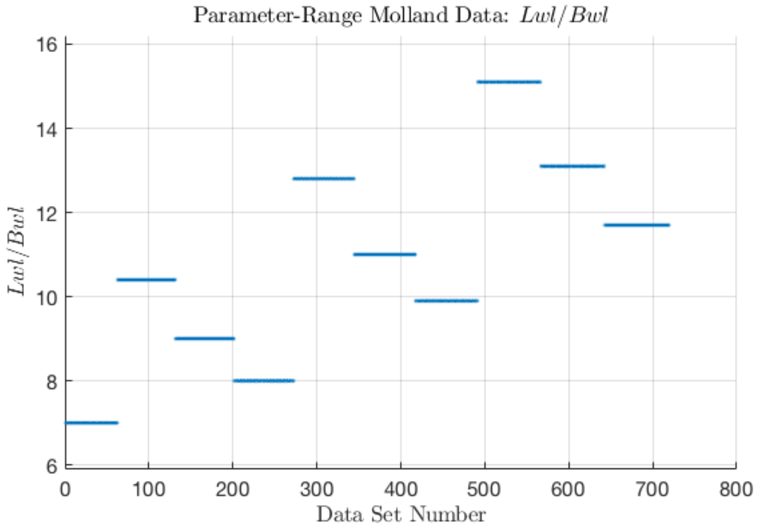

Figure A.9: Parameter range of Lwl/Bwl in additional data set.

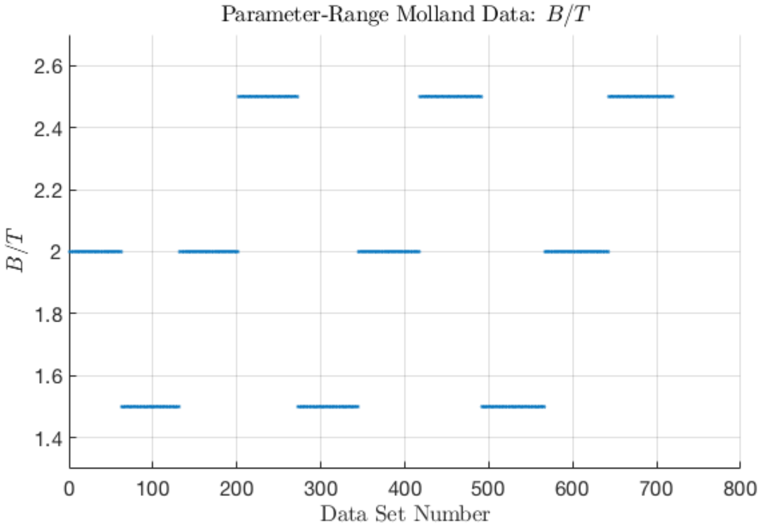

Figure A.10: Parameter range of B/T in additional data set.

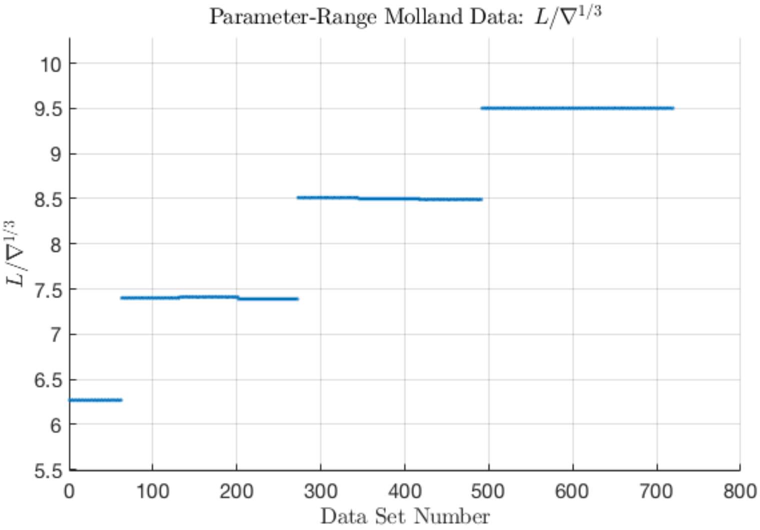

Figure A.11: Parameter range of $L/\nabla^{1/3}$ in additional data set.

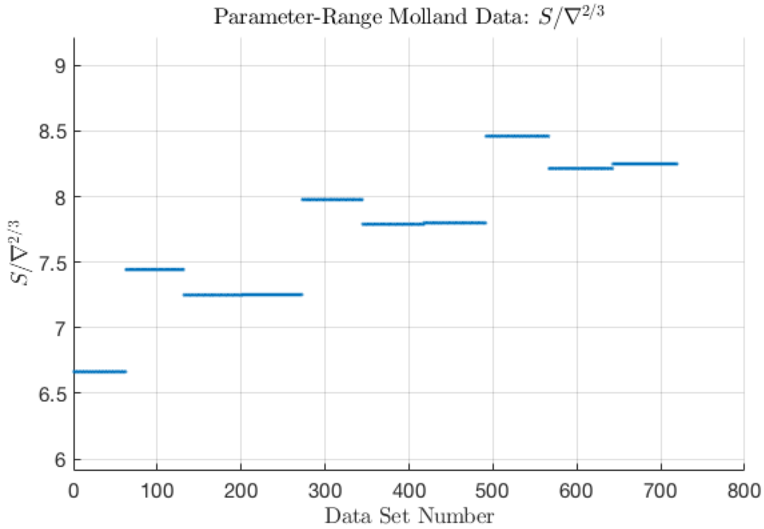

Figure A.12: Parameter range of $S/\nabla^{2/3}$ in additional data set.

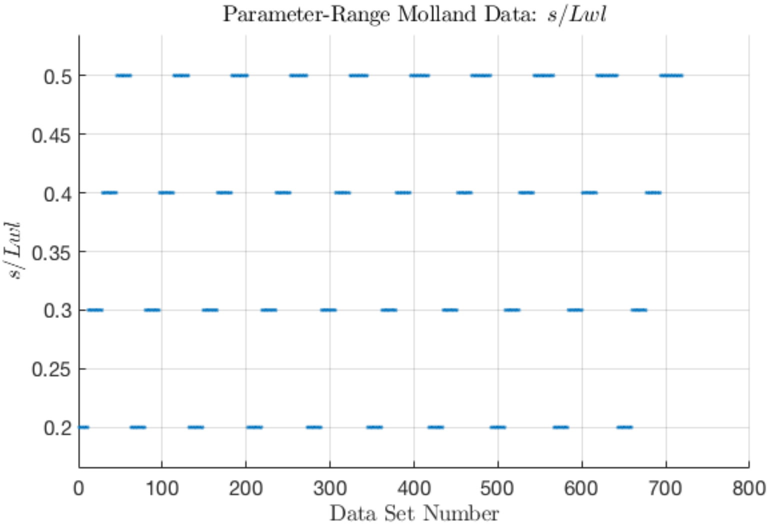

Figure A.13: Parameter range of s/Lwl in additional data set.

Appendix B

Additional Results

B.1 Additional Results from Parameter Study

Note: The elapsed time is plotted with a logarithmic y-scale. This makes it easier to compare results with similar time consumption, but the elapsed time does also seem more similar for the networks.

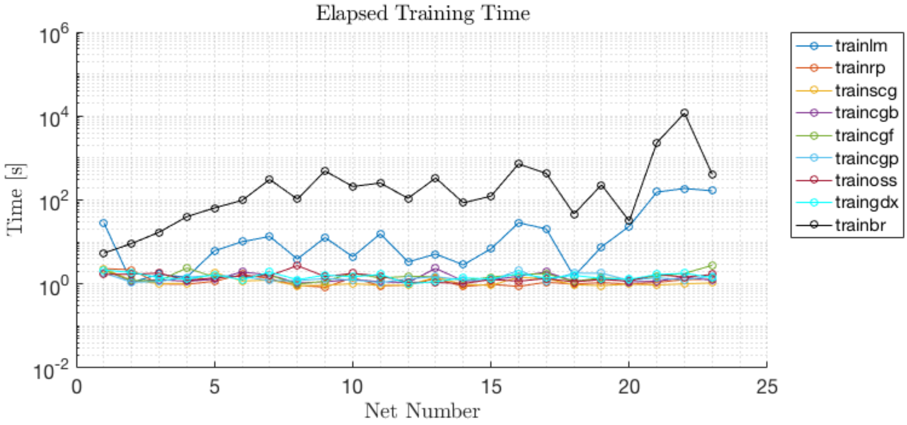

Figure B.1: Elapsed time during parameter study of training functions.

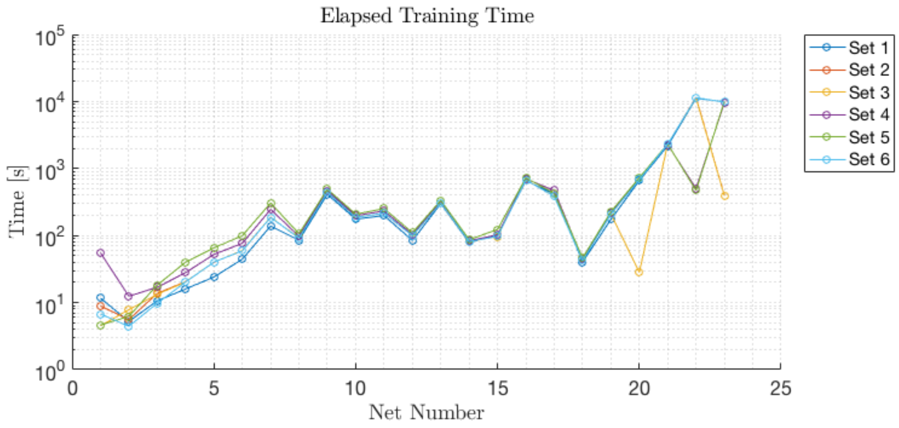

Figure B.2: Elapsed time during parameter study of parameter sets.

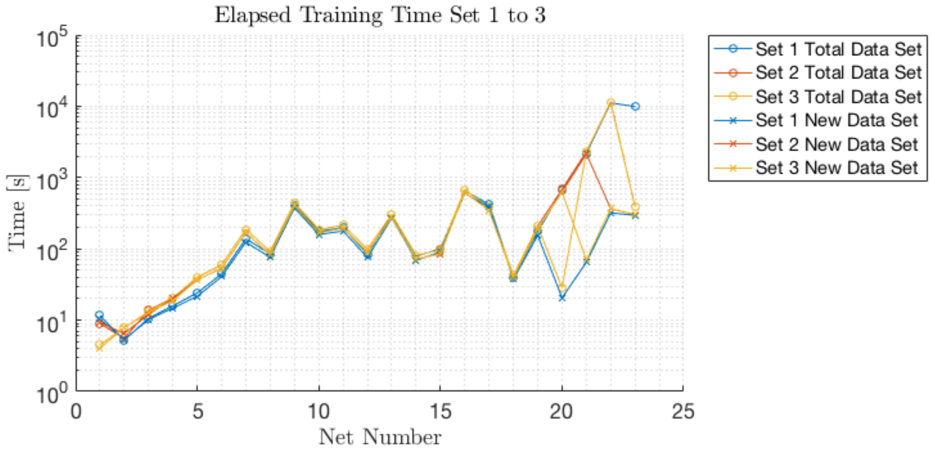

Figure B.3: Elapsed time during parameter study of data sets, part 1.

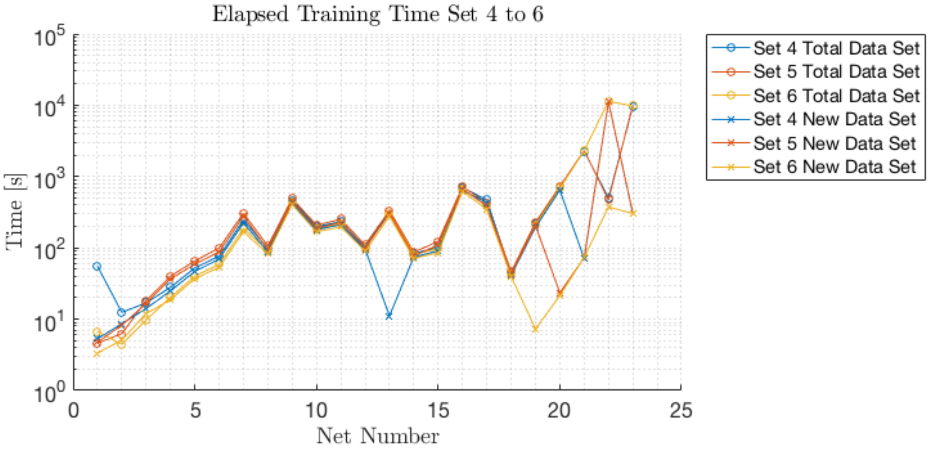

Figure B.4: Elapsed time during parameter study of data sets, part 2.

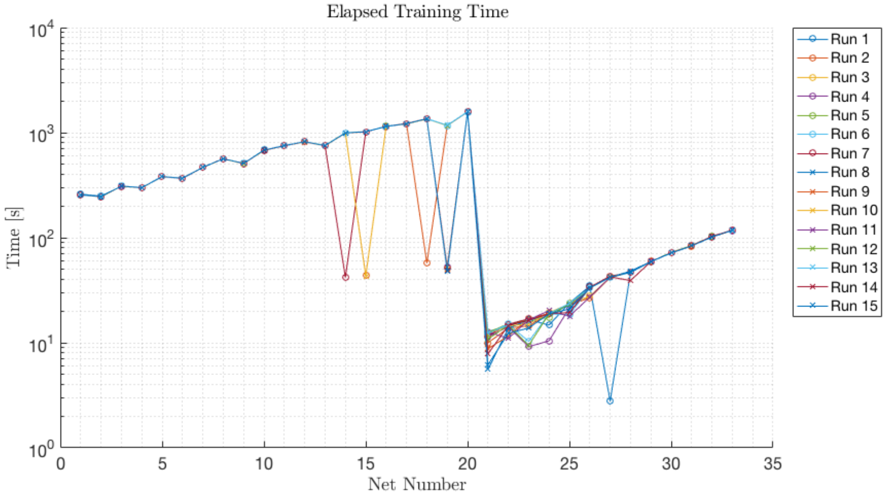

Figure B.5: Elapsed time during fine-tuning of networks.

B.2 Additional Results for Fine Tuned Networks

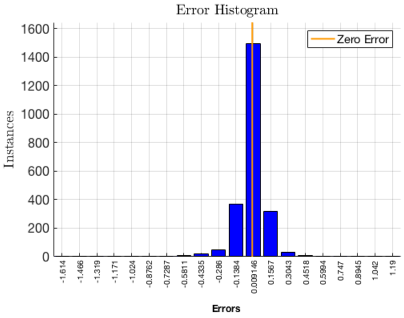

Figure B.6: Error histogram for network trained for 19 epochs.

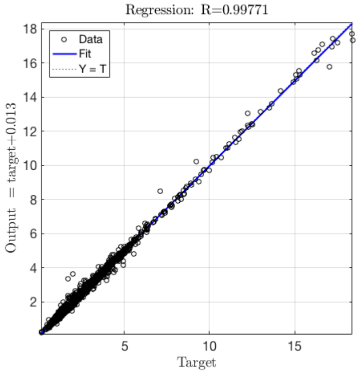

Figure B.7: Regression plot for network trained for 19 epochs.

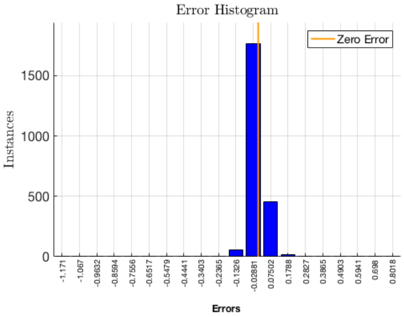

Figure B.8: Error histogram for network trained for 996 epochs.

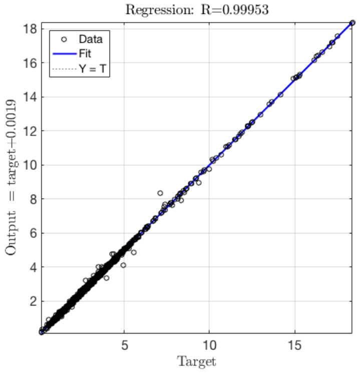

Figure B.9: Regression plot for network trained for 996 epochs.

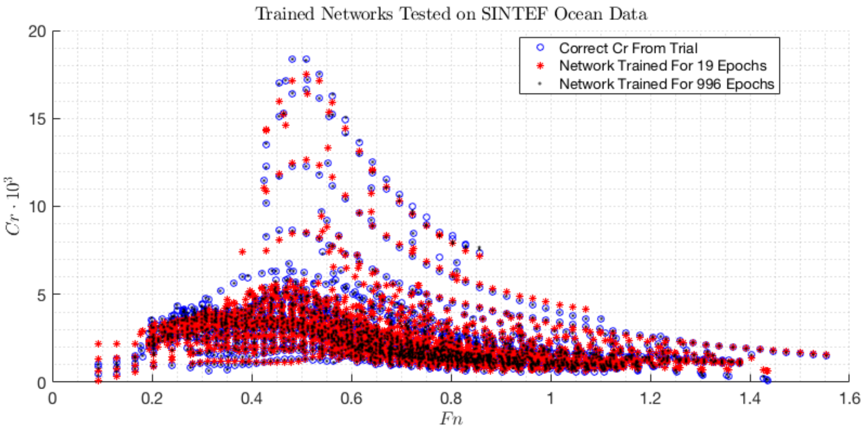

Figure B.10: Network ability to predict *Cr* for SINTEF Ocean data, full set.

B.3 Over-trained Network Tested on Southampton Data Set

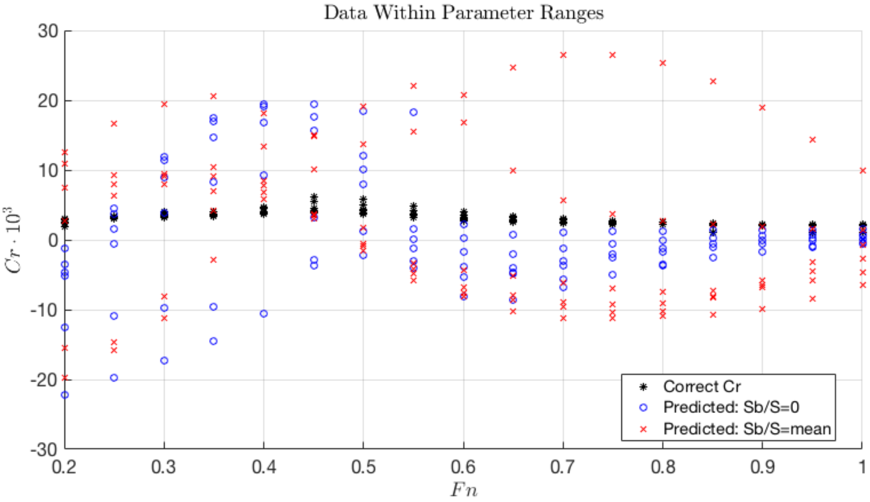

Figure B.11: Validation results for data within parameter range - 'over-trained' network.

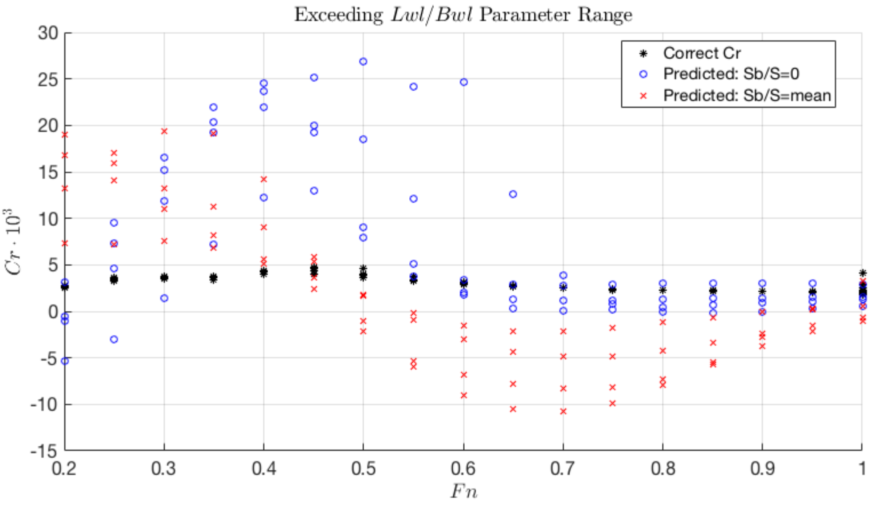

Figure B.12: Validation results for data exceeding Lwl/Bwl parameter range - 'over-trained' network.

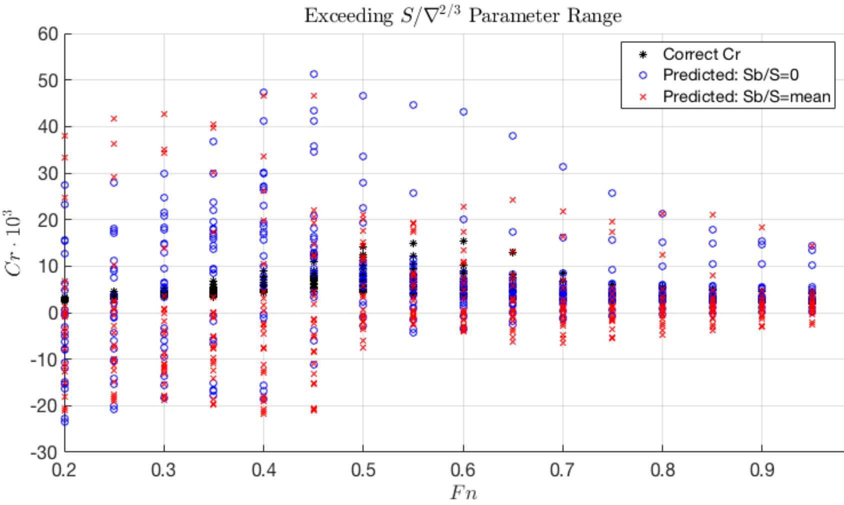

Figure B.13: Validation results for data exceeding $S/\nabla^{2/3}$ parameter range - 'over-trained' network.

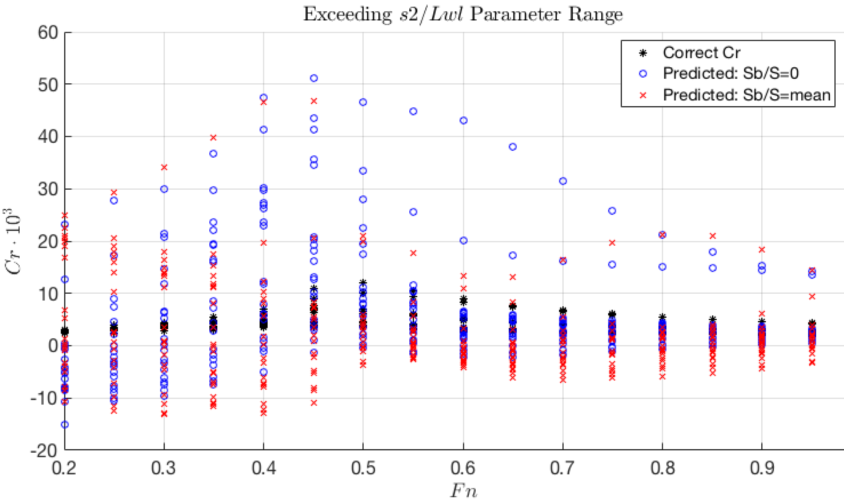

Figure B.14: Validation results for data exceeding $s2/Lwl$ parameter range - 'over-trained' network.

B.4 Missing Input Analysis, Presentation of Full Data Set

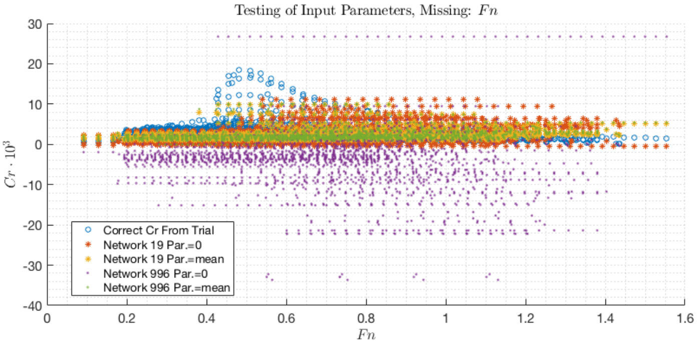

Figure B.15: Network ability to predict Cr in SINTEF Ocean data, when input parameter Fn is missing. Results for full data set.

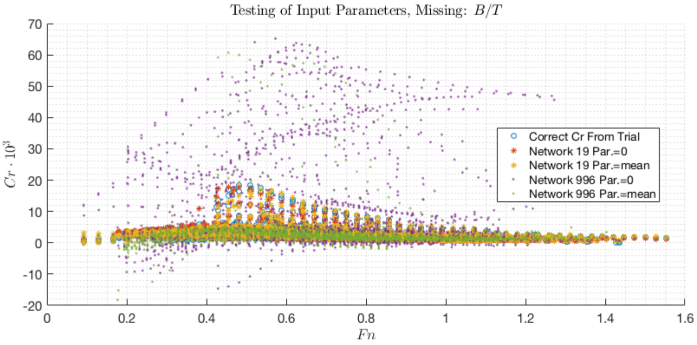

Figure B.16: Network ability to predict Cr in SINTEF Ocean data, when input parameter B/T is missing. Results for full data set.

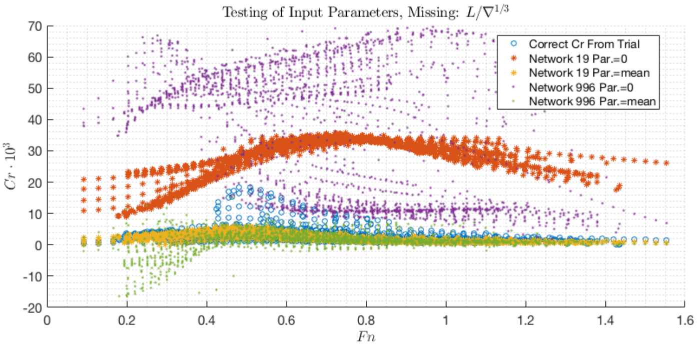

Figure B.17: Network ability to predict Cr in SINTEF Ocean data, when input parameter $L/\nabla^{1/3}$ is missing. Results for full data set.

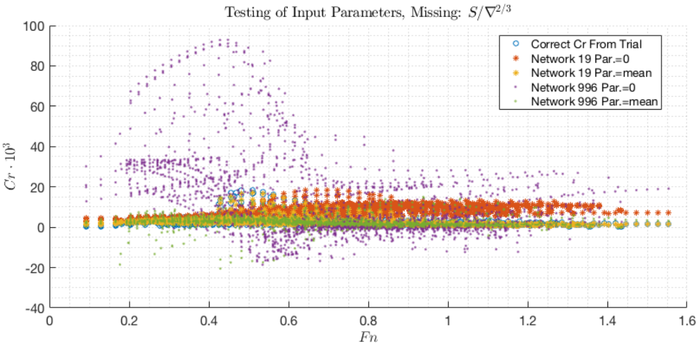

Figure B.18: Network ability to predict Cr in SINTEF Ocean data, when input parameter $S/\nabla^{2/3}$ is missing. Results for full data set.

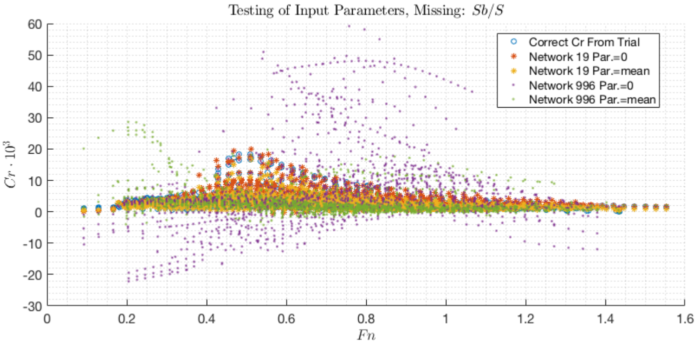

Figure B.19: Network ability to predict Cr in SINTEF Ocean data, when input parameter Sb/S is missing. Results for full data set.

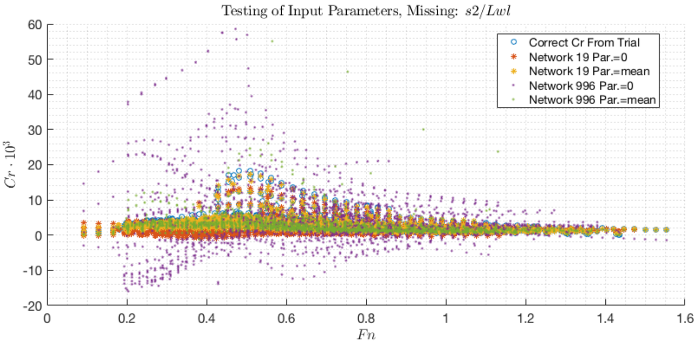

Figure B.20: Network ability to predict Cr in SINTEF Ocean data, when input parameter $s2/Lwl$ is missing. Results for full data set.

Appendix C

Exported Empirical Resistance Function

The weights and biases in the function below are made anonymous as mentioned in Chapter 5, due to the non-disclosure agreement between the author and SINTEF Ocean.

```
function [y1] = CatRES2_anonymous(x1)
%CATRES2 neural network simulation function.
% NOTE: ONES INSERTED FOR ANONYMISATION OF TRAINED MODEL.
%
% Generated by Neural Network Toolbox function genFunction, 05-Jun-2018 11:22:32.
%
% [y1] = CatRES2(x1) takes these arguments:
%   x = 6xQ matrix, input
%           x(1,Q) = Froude number Fn
%           x(2,Q) = B/T ratio
%           x(3,Q) = L/(volumeDisplacement^(1/3))
%           x(4,Q) = S/(volumeDisplacement^(2/3))
%           x(5,Q) = s2/Lwl
%           x(6,Q) = Sb/S
% and returns:
%   y = 1xQ matrix, output
%           y(1,Q) = Predicted residuary resistance  coefficients  (x1000)
%
% where Q is the number of samples.
%
% Example:
%   innEX =   [0.71;
%              2.43;
%              7.73;
%              10.50;
%              0.01;
%              0.21]
%
%   CatRES2([innEX) = 1.71
%
%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset  =  [1;1;1;1;1;1];
x1_step1.gain  =  [1;1;1;1;1;1];
```

```
x1_step1.ymin = −1;

% Layer 1
b1 =  [1;1;1;1;1;1;1;1;1;1];
IW1_1 = [1 1 1 1 1 1; 1 1 1 1 1 1; 1 1 1 1 1 1; 1 1 1 1 1 1; 1 1 1 1 1 1; 1 1 1 1 1 1; 1 1 1 1 1 1; 1 1 1 1 1 1; 1 1 1 1 1 1;
      1 1 1 1 1  1];

% Layer 2
b2 =  [1;1;1;1;1;1;1;1;1;1;1];
LW2_1 = [1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1
      1 1 1 1; 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1  1];

% Layer 3
b3 =  [1;1;1;1;1;1;1;1;1;1;1];
LW3_2 = [1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1
      1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1
      1 1 1 1 1 1 1  1];

% Layer 4
b4 =  [1;1;1;1;1;1;1;1;1;1];
LW4_3 = [1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1
      1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1 1; 1 1 1 1 1 1 1 1 1 1  1];

% Layer 5
b5 = 1;
LW5_4 = [1 1 1 1 1 1 1 1 1  1];

% Output 1
y1_step1.ymin = −1;
y1_step1.gain = 1;
y1_step1.xoffset = 1;

% ===== SIMULATION ========

% Dimensions
Q = size(x1,2); % samples

% Input 1
xp1 = mapminmax_apply(x1,x1_step1);

% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1∗xp1);

% Layer 2
a2 = tansig_apply(repmat(b2,1,Q) + LW2_1∗a1);

% Layer 3
a3 = tansig_apply(repmat(b3,1,Q) + LW3_2∗a2);

% Layer 4
a4 = tansig_apply(repmat(b4,1,Q) + LW4_3∗a3);

% Layer 5
a5 = repmat(b5,1,Q) + LW5_4∗a4;

% Output 1
y1 = mapminmax_reverse(a5,y1_step1);
end

% ===== MODULE FUNCTIONS ========

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
  y = bsxfun(@minus,x,settings.xoffset);
  y = bsxfun(@times,y,settings.gain);
  y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
  a = 2 ./ (1 + exp(−2∗n)) − 1;
end

% Map Minimum and Maximum Output Reverse−Processing Function
function x = mapminmax_reverse(y,settings)
  x = bsxfun(@minus,y,settings.ymin);
  x = bsxfun(@rdivide,x,settings.gain);
  x = bsxfun(@plus,x,settings.xoffset);
end
```