Using Parametrized Propeller Data for Training Neural Networks

Maike Strecker

Fluid Dynamics and Ship Theory

TUHH

Agenda

1. Introduction
2. Propeller data
3. Neural network
4. Autoencoder
5. Conclusion and Outlook

Fluid Dynamics and Ship Theory

TUHH

Agenda

1. **Introduction**
2. Propeller data
3. Neural network
4. Autoencoder
5. Conclusion and Outlook

Fluid Dynamics and Ship Theory

TUHH

Introduction

- Accelerate the propeller design process
- Design with AI increasing
- Research for usage of ML in propeller design
- Genetic algorithms, regression trees, combinations of SVM and GA, NNs …
- Research about necessary data amount for NNs missing

Agenda

1. Introduction
2. **Propeller data**
3. Neural network
4. Autoencoder
5. Conclusion and Outlook

Propeller data

Flowdiagram

CAESES → StarCCM+ → Neural Network

CAESES
- Generate parametrized geometries
- Automatic CAD file generation

StarCCM+
- Calculate OWC for geometries automatically
- Output OWC and pressure distribution

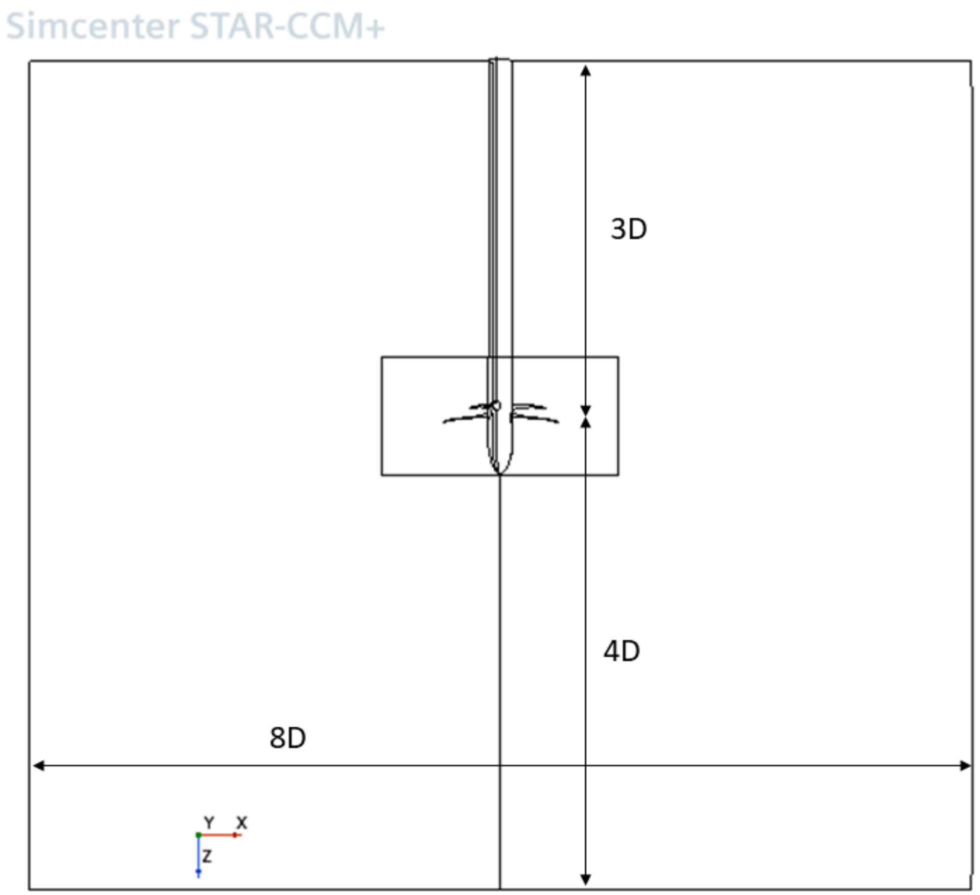
Neural Network
- Use generated data as training data
- Different dataset sizes

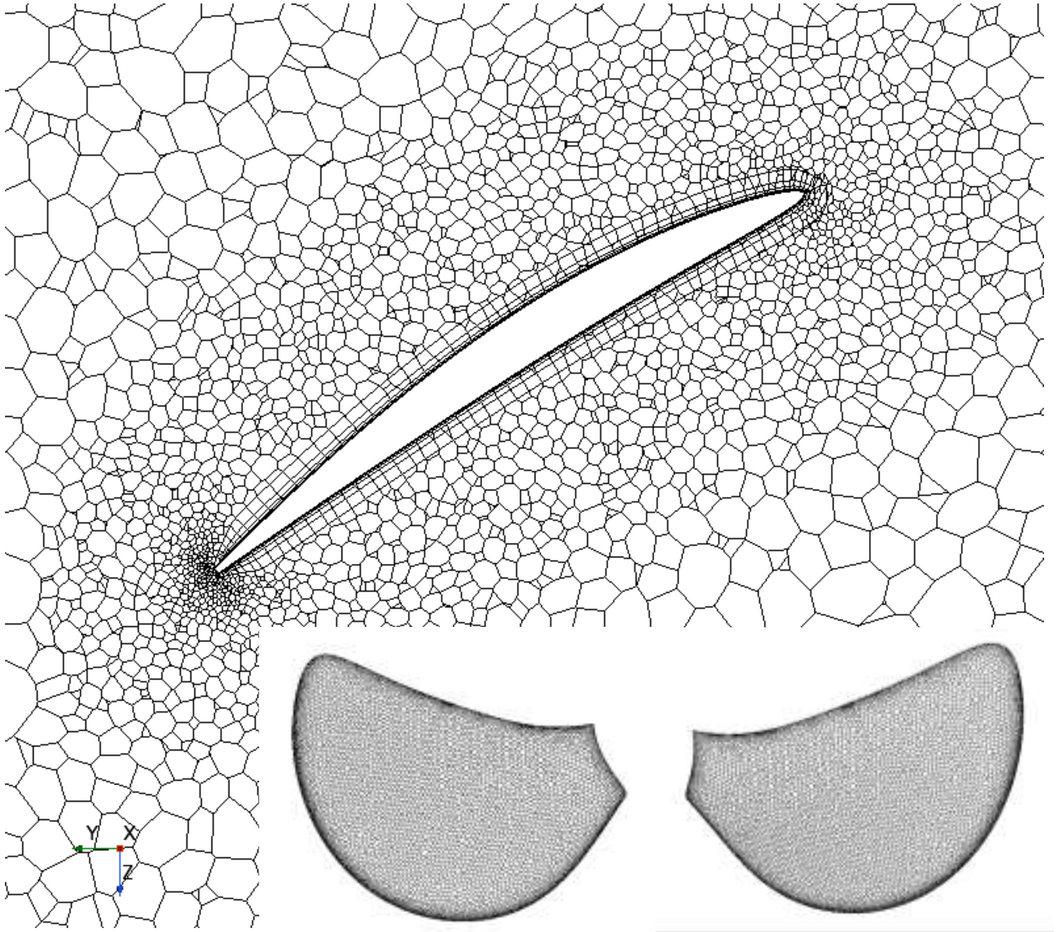
Fluid Dynamics and Ship Theory

TUHH

Propeller data

CFD setup





Fluid Dynamics and Ship Theory

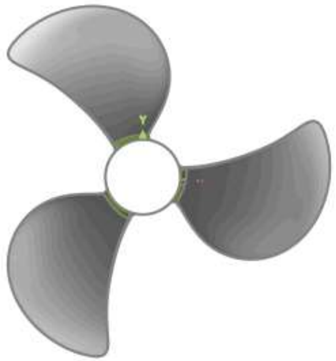
TUHH

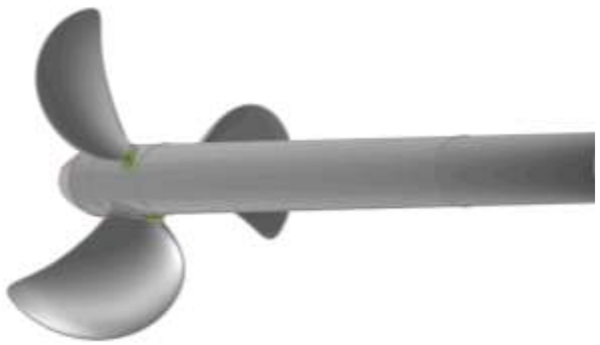
Propeller data

Geometry parameters

Parameter	Min	Max
Chord - Hub	0.25	0.4
Chord - Delta	0.0	0.2
Camber	0.01	0.06
Pitch - Base	0.9	2.0
Pitch - Delta	0.05	0.3
Skew	0.1	0.35
Number of blades	3	*6*

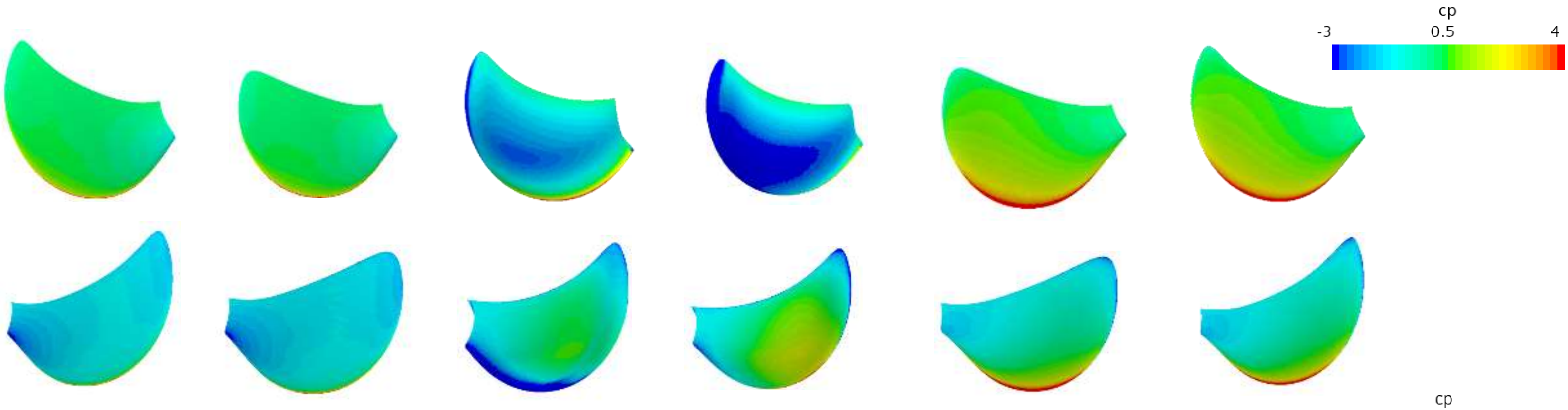
- Check for outliers using k-means-clustering
- Cut efficiency values above 1 and below 0
- Four dataset sizes: 100, 500, 1000 and 3000





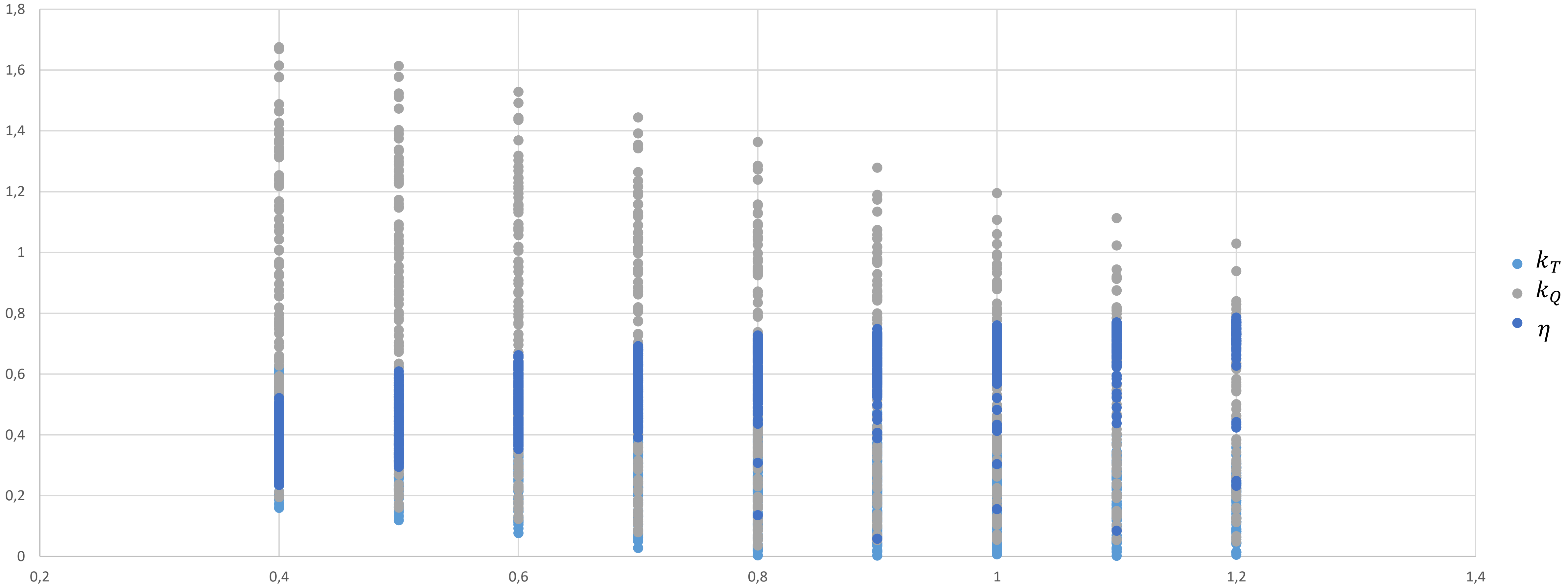
Propeller data

cp on different blade geometries



Propeller data

Open water curve scatterplot



Fluid Dynamics and Ship Theory

TUHH

Agenda

1. Introduction
2. Propeller data
3. **Neural network**
4. Autoencoder
5. Conclusion and Outlook

Fluid Dynamics and Ship Theory

TUHH

Neural network

Input and Output

Thrust

Input	Output
Z	J
${k_T}^*$	η
k_Q (variiert)	k_T

Torque

Input	Output
Z	J
${k_Q}^*$	η
k_T (variiert)	k_Q

$$ {k_T}^* = \frac{k_T}{J^2} = \frac{R_T}{\rho D^2 (1-t) v_s^2 (1-w)^2} $$

$$ {k_Q}^* = \frac{k_Q}{J^3} = \frac{\eta_R P_D}{2\pi \rho D^2 v_s^3 (1-w)^3} $$

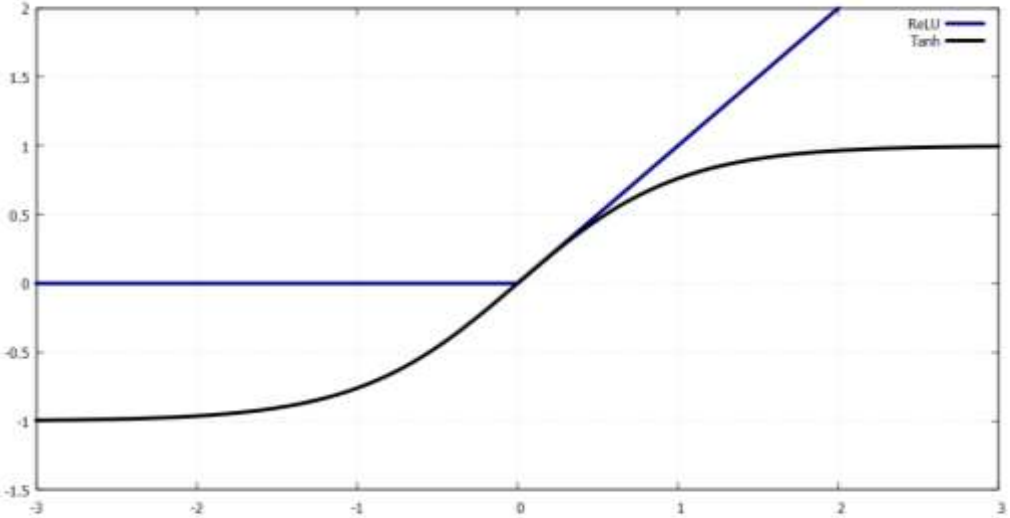
Geometry

Input	Output
J	A_e / A_o
η	P
k_T	chord
k_Q	camber
Z	skew
	cavitation

Neural network

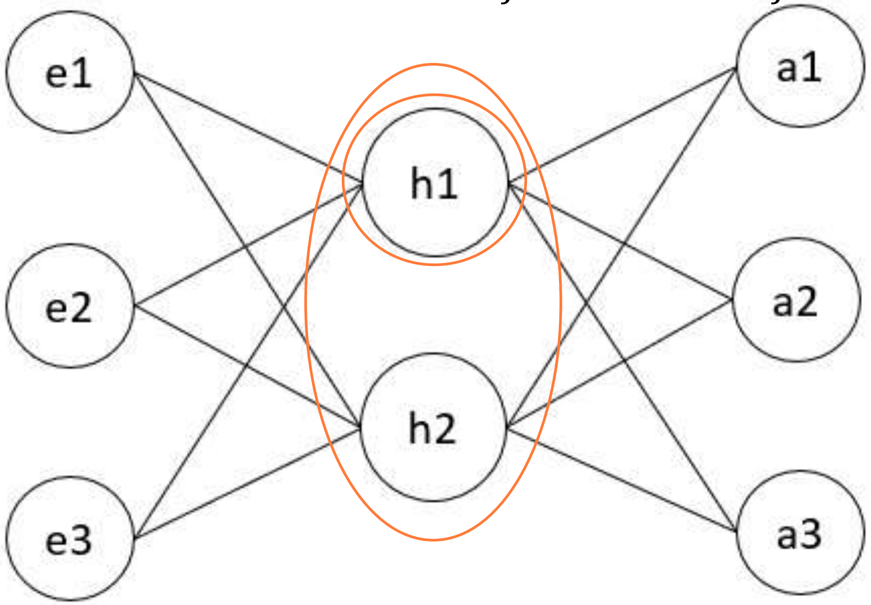
Working principle

- Neural network in PyTorch
- Activation function
- Solver with learning rate
- Number of layers
- Number of neurons



$$a = f(e1 \cdot w1 + e2 \cdot w2 + e3 \cdot w3)$$

$$\Delta w_{ij} = \sigma \cdot e_i \cdot \Delta a_j$$



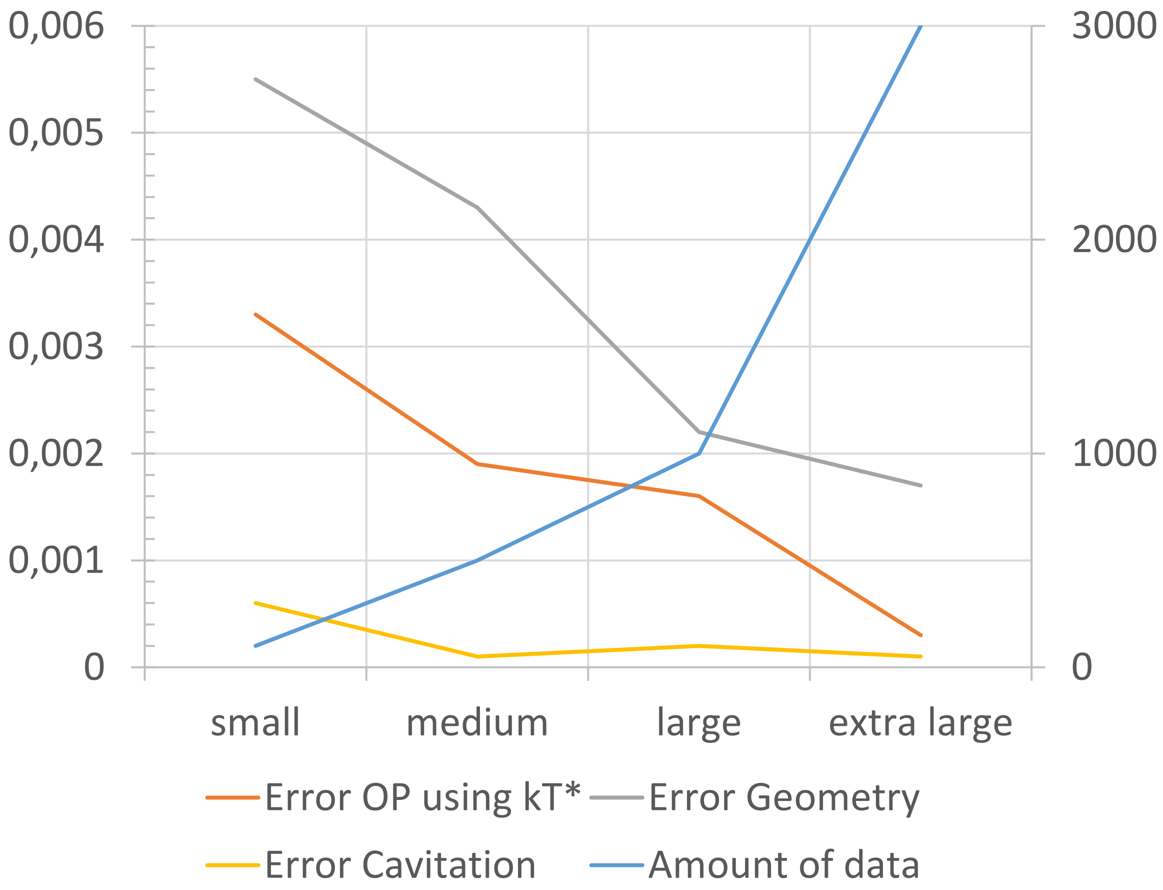
Neural network

Adjusted parameter

Parameter	Variations
Number of hidden layers	2
Neurons per layer	8, 16, 32
Activation function	ReLU, LeakyReLU, Softplus, Tanh
Optimizer	SGD, Adam, rmsprop
Learning rate	0.00001 to 0.1
Batch size	1, 2, 4, 8
Dropout probability	0, 0.1, 0.5

Neural network

Influence of training data amount on errors



Fluid Dynamics and Ship Theory

TUHH

Agenda

Autoencoder

Input and Output

Thrust

Input
- Z
- $k_T{}^*$
- k_Q (variiert)

Output
- J
- η
- k_T

Torque

Input
- Z
- $k_Q{}^*$
- k_T (variiert)

Output
- J
- η
- k_Q

Geometry

Input
- J
- η
- k_T
- k_Q
- Z

Output
- $A_e\ /A_o$
- P
- chord
- camber
- skew
- *cavitation*

Pressures

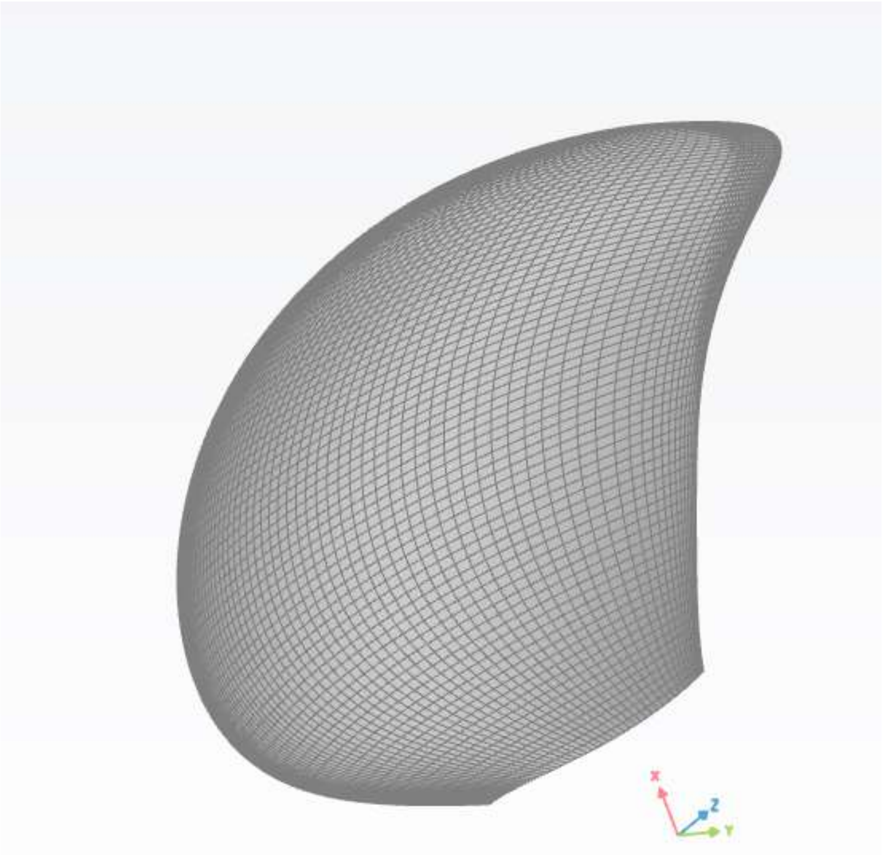
Input
- J
- P
- chord
- camber
- skew
- Z

Output
- shear and pressures:
- cp
- cavitation
- k_T
- k_Q
- η

Fluid Dynamics and Ship Theory

TUHH

Autoencoder

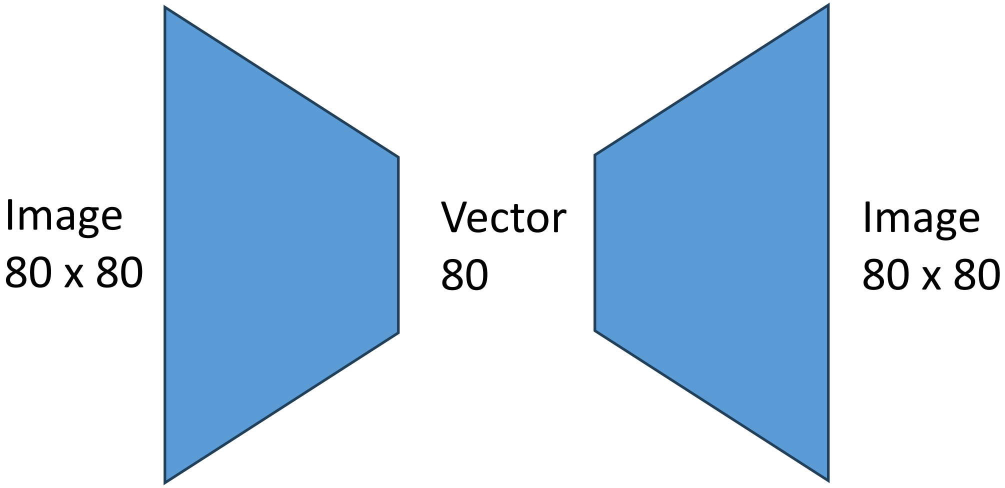


- 80x80 grid
- Seperate for face and back
- Allows 2D Convolutional Neural Network (CNN) operations

Fluid Dynamics and Ship Theory

TUHH

Autoencoder



- Convolutional Neural Network (CNN) in 2D – Autoencoder
- 80x80 grid input
- 4 Conv2D-Layer with activation function and pooling in between
- Kernel = 3
- Padding = 1
- Stride = 2

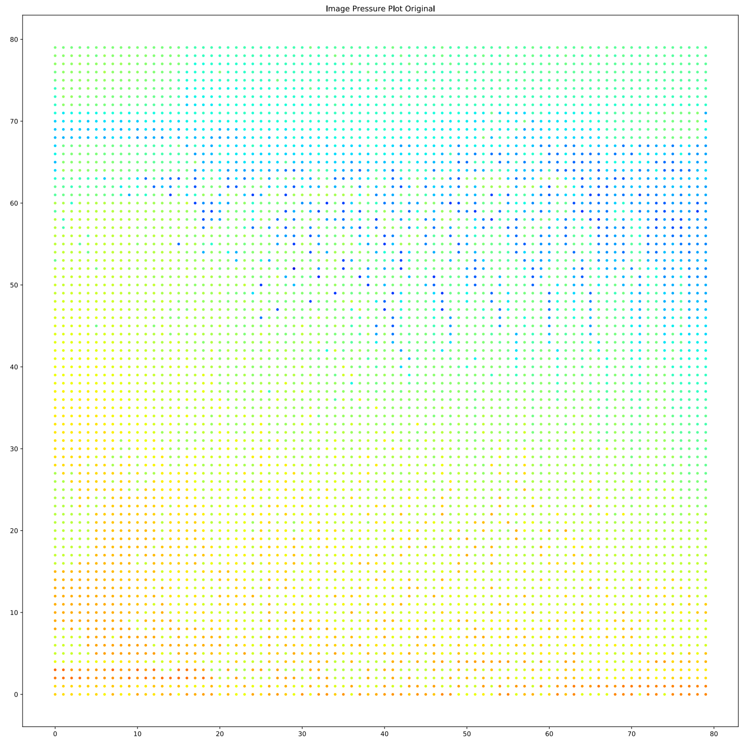
Autoencoder

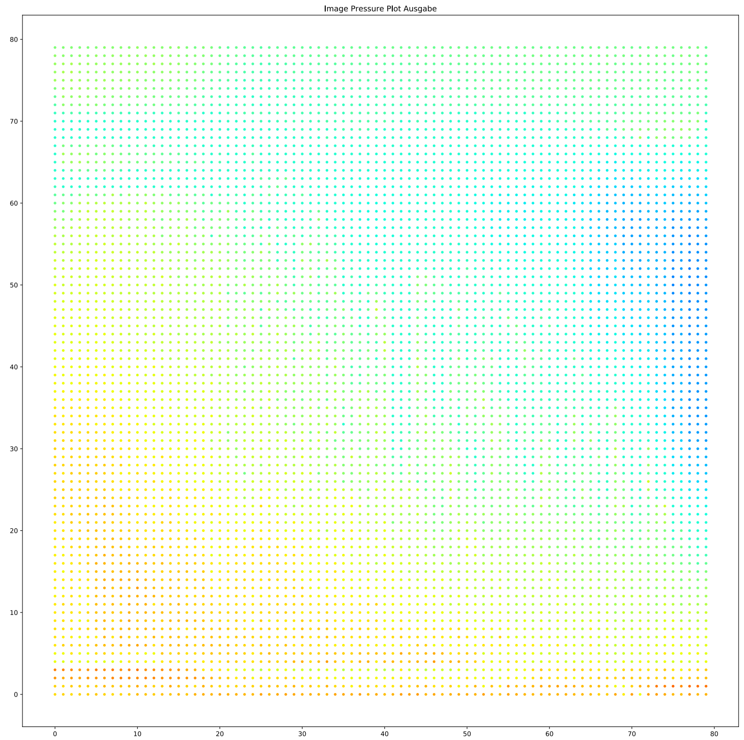
Adjusted parameter

Parameter	Variations
Activation function	ReLU, LeakyReLU, Softplus, Mish, ELU
Optimizer	SGD, Adam, rmsprop
Learning rate	0.00001 to 0.1
Batch size	1, 2, 4, 8
Latent dimension	80, 128

Autoencoder

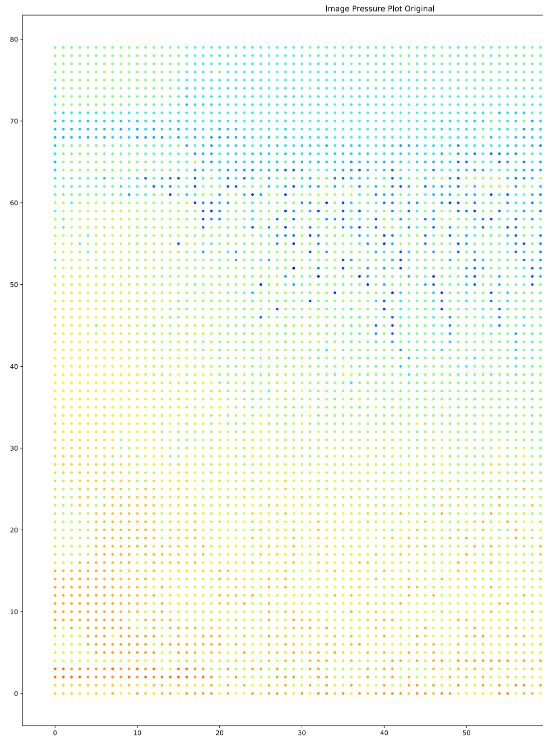
Num = 200

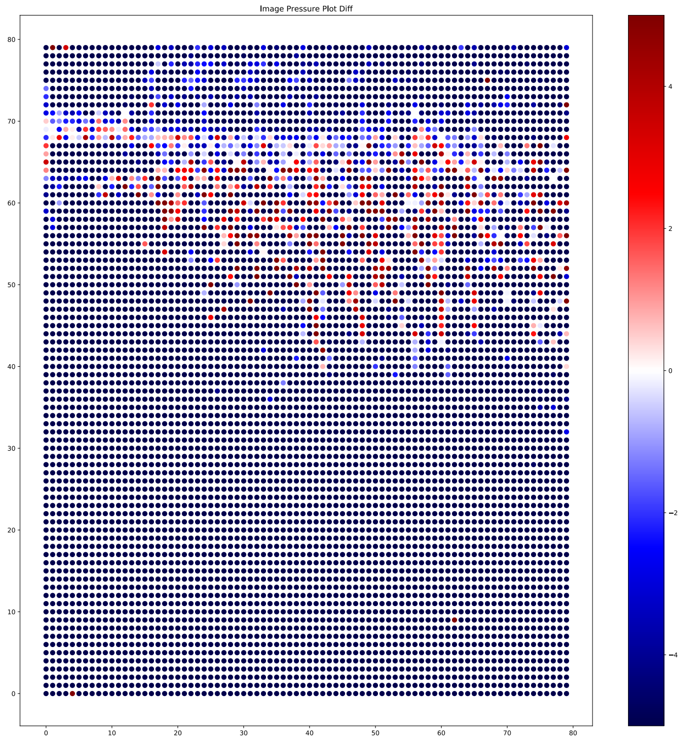


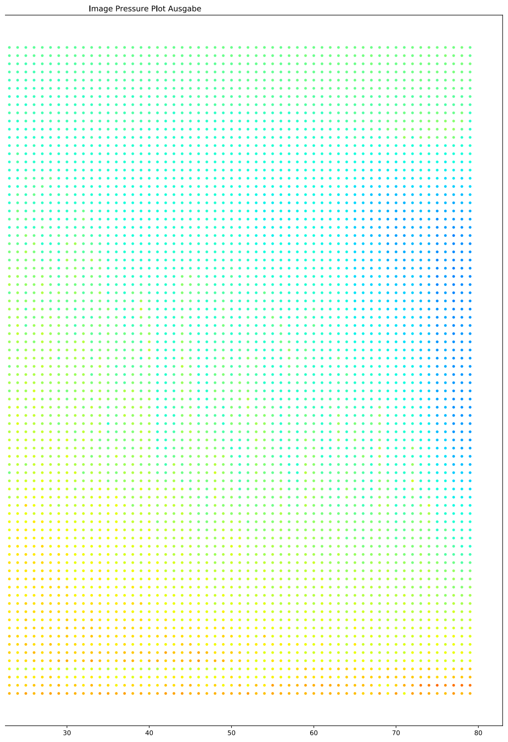


Autoencoder

Num = 200





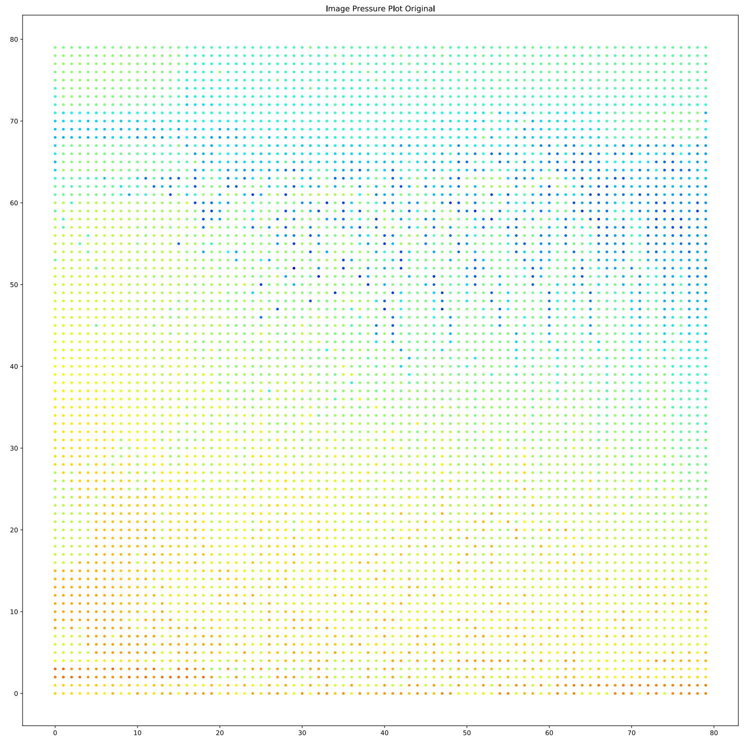


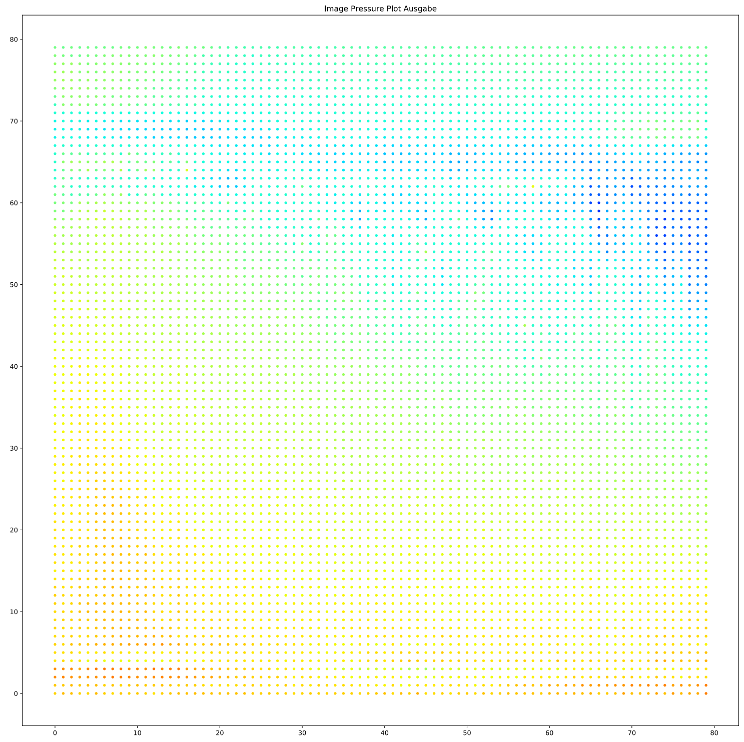
Fluid Dynamics and Ship Theory

TUHH

Autoencoder

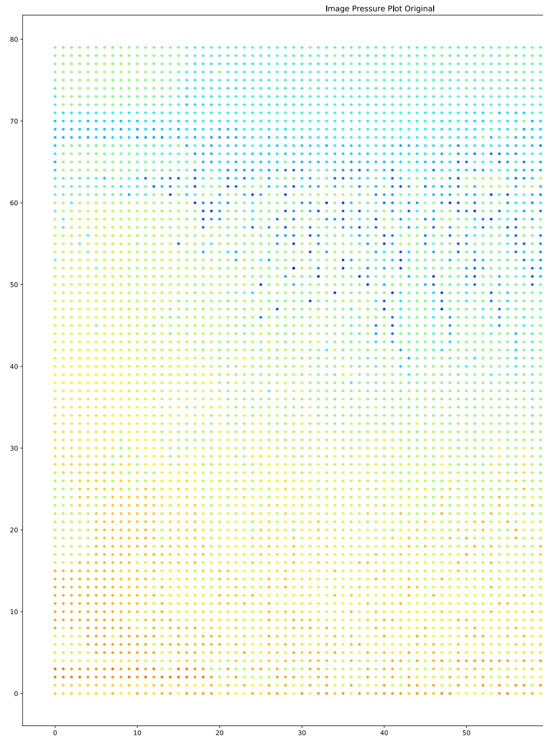
Num = 500

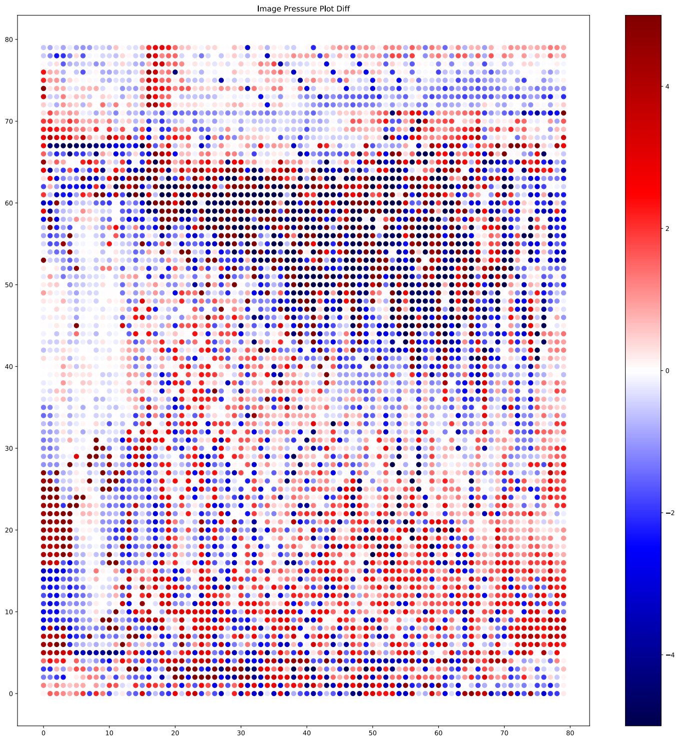


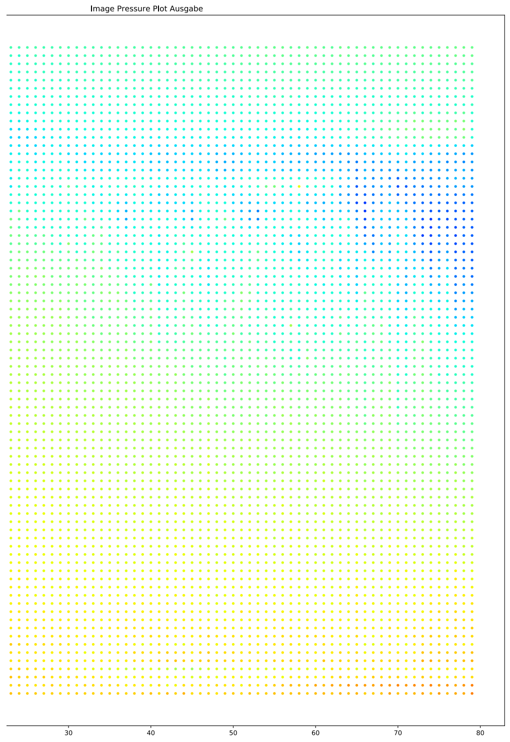


Autoencoder

Num = 500

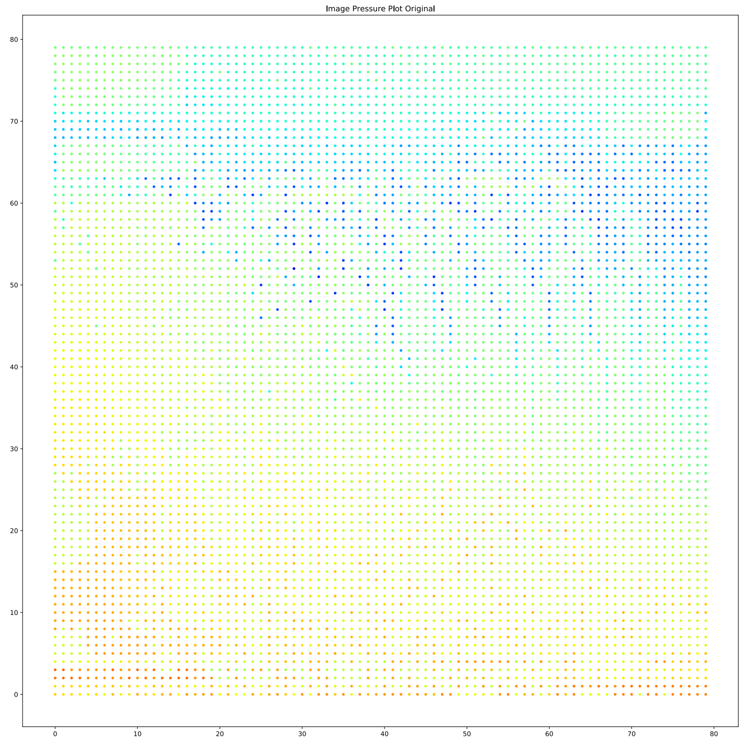


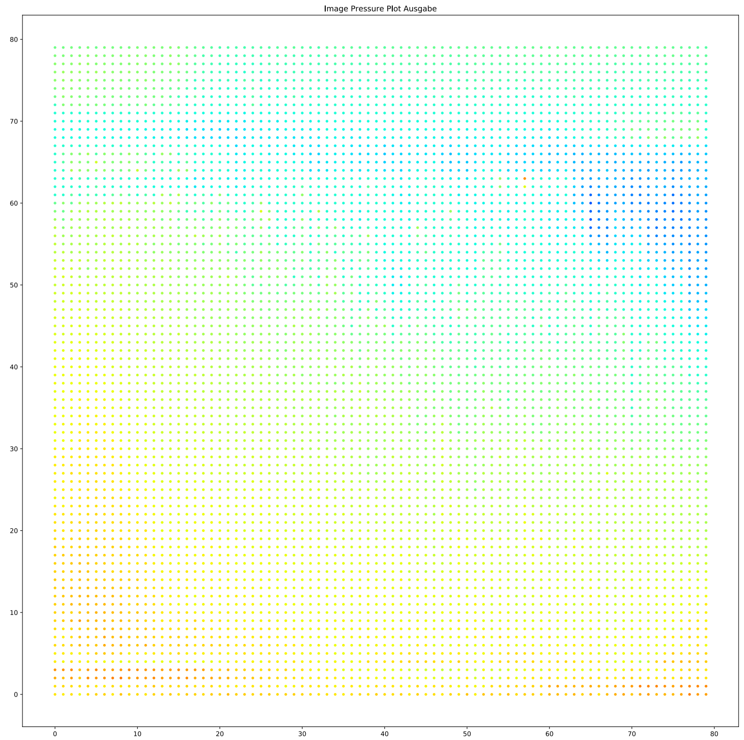




Autoencoder

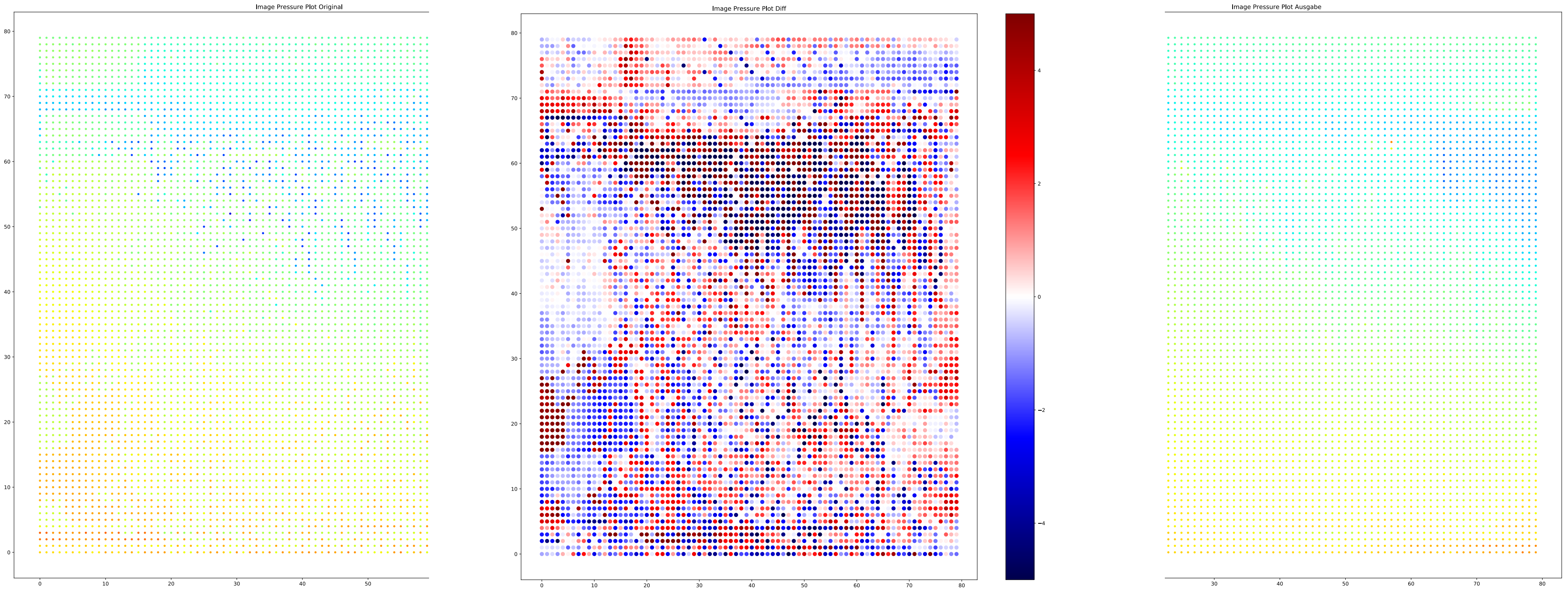
Num = 1000





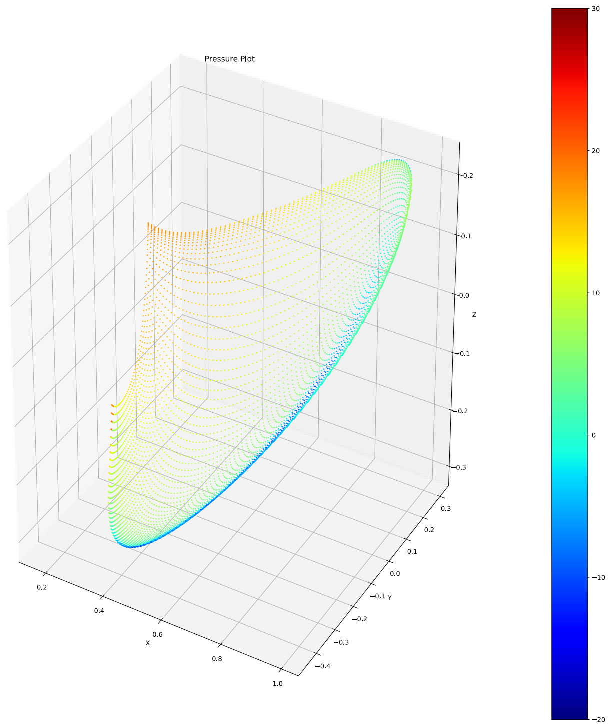
Autoencoder

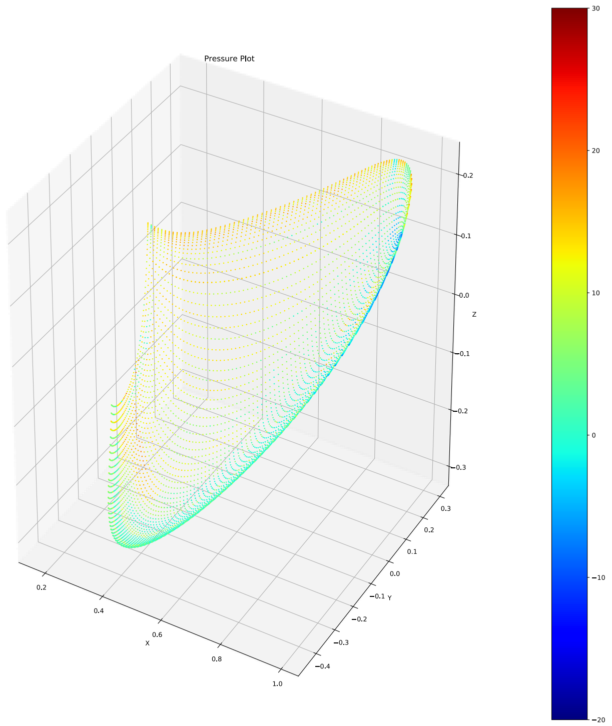
Num = 1000



Autoencoder

Num = 200



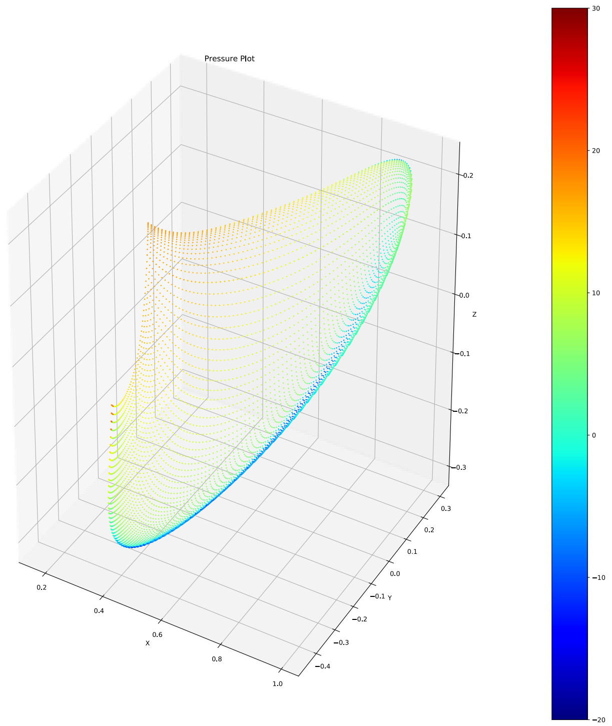


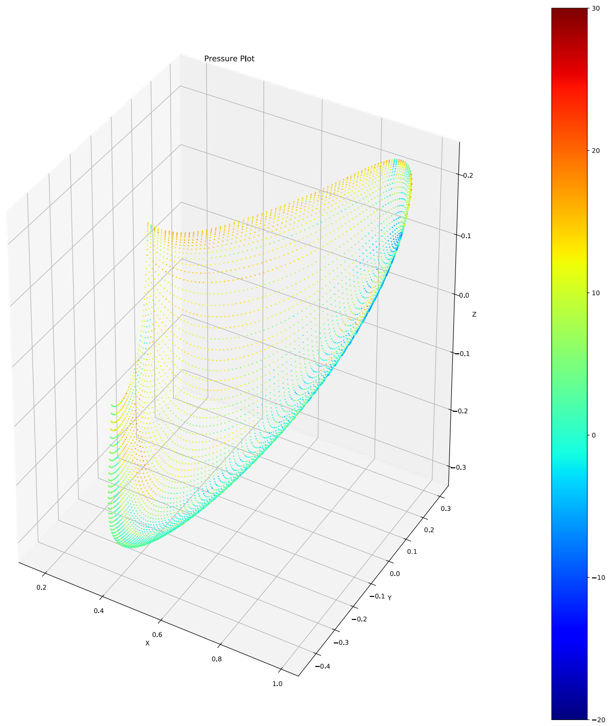
Fluid Dynamics and Ship Theory

TUHH

Autoencoder

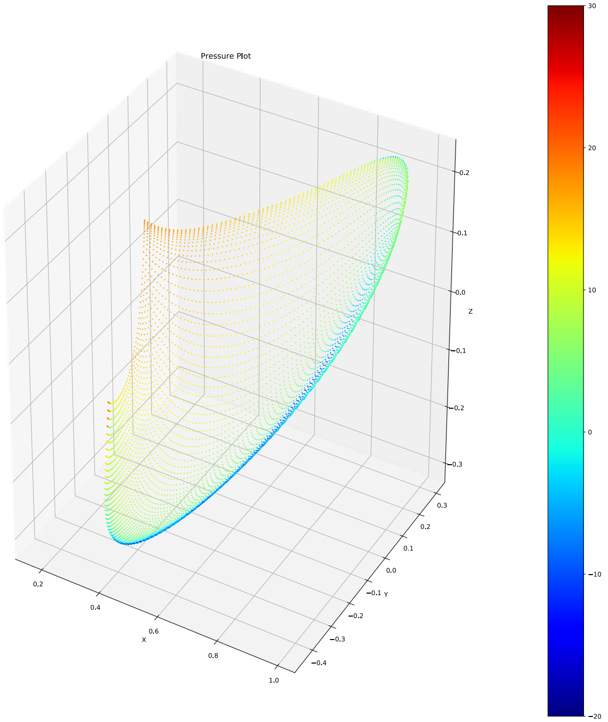
Num = 500

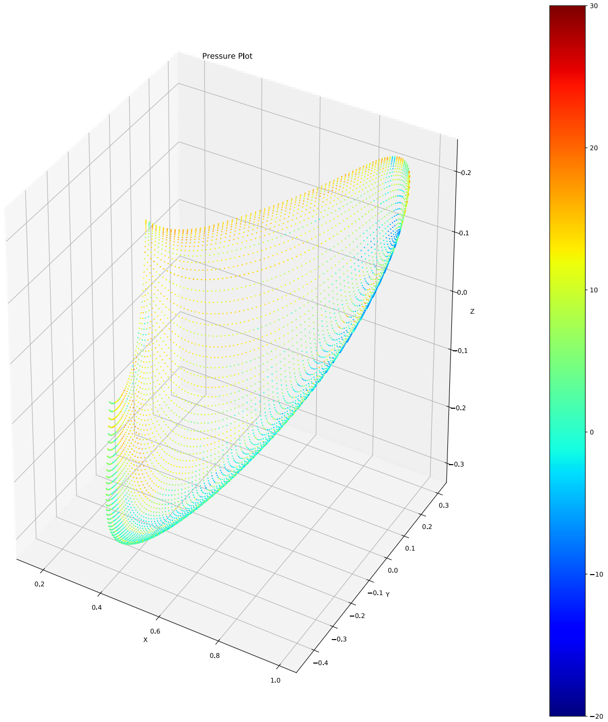




Autoencoder

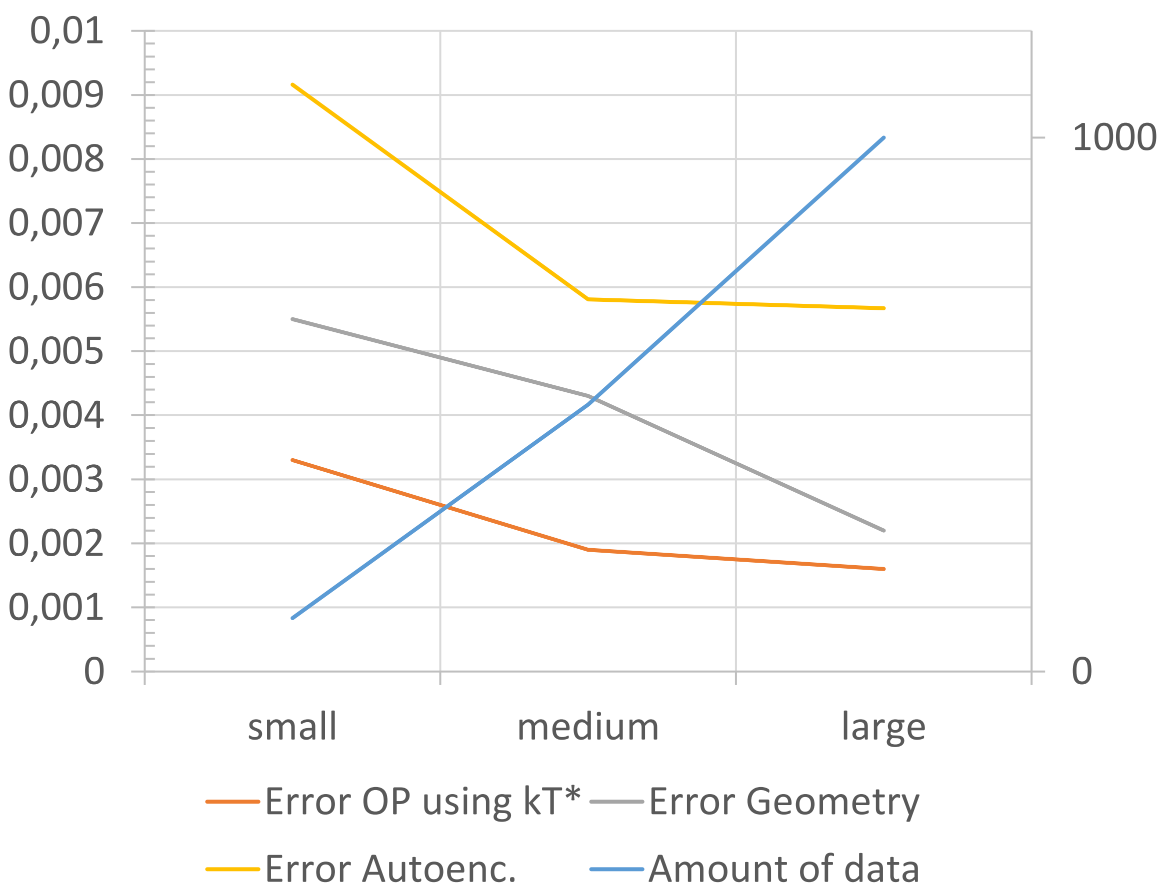
Num = 1000





Autoencoder

Influence of training data amount on errors



Agenda

1. Introduction
2. Propeller data
3. Neural network
4. Autoencoder
5. **Conclusion and Outlook**

Fluid Dynamics and Ship Theory

TUHH

Conclusion and Outlook

- Neural networks are easily usable and show reasonable quality
  - Fast results compared to CFD calculation
- Similar datasets can be trained with pre-set parameters
- Necessary amount of data 500 points or more
  - Depending on the output
- Known physical equations have partly been used
  - More equation usage
- Pressure data as 3D data
  - Data amount study
  - Combine with other networks

Thank you
for your attention!