Politecnico di Torino

Politecnico di Torino

MSc in Energy and Nuclear Engineering

Development of artificial intelligence (AI) techniques for Critical Heat Flux predictions

Candidate: Daniele Castrignanò

Supervisor: Prof. Nicola Pedroni

Academic Year: 2024/2025

Abstract

Safety is a fundamental aspect of the nuclear industry, especially in the period after Chernobyl and Fukushima accidents. One of the most critical phenomena affecting reactor safety is the Critical Heat Flux (CHF), which marks the transition to a deteriorated heat transfer regime in nuclear reactor cores, potentially leading to severe damage. Accurate CHF prediction is therefore essential for enhancing nuclear safety and optimizing reactor performance.

Since 2006, CHF prediction has primarily relied on the Look-Up Table (LUTs) method, a well-established empirical approach. However, with advancements in computational techniques, machine learning (ML) has seen as a promising tool to improve prediction accuracy. In this context, the Task Force on Artificial Intelligence and Machine Learning for Scientific Computing in Nuclear Engineering, under the supervision of the Expert Group on Reactor Systems Multi-Physics (EGMUP), has been actively developing ML-based models for CHF prediction.

This study follows a similar approach by first providing a theoretical framework on CHF, analysing its governing physical mechanisms and the key parameters influencing CHF in Pressurized Water Reactor (PWR) operations. An extensive data set is analysed to identify the most relevant input variables through correlation analysis performed in MATLAB. An initial neural network model is then developed and subsequently refined through a series of improvements aimed at enhancing prediction accuracy. The different versions of the model are evaluated and compared against empirical methods to assess their reliability and potential advantages.

The results highlight the benefits of ML-based approaches, demonstrating their capability to enhance safety margins and optimize reactor operation. Given the increasing complexity of nuclear systems and the growing demand for safe and efficient energy production, the development of advanced predictive models will play a strategic role in the future of nuclear engineering, enabling more reliable safety assessments and improved reactor design.

"There must be no barriers to freedom of inquiry. There is no place for dogma in science. The scientist is free, and must be free to ask any question, to doubt any statement, to seek any proof, to correct any error."

– Julius Robert Oppenheimer

Contents

1 The importance of Safety in Nuclear Energy **1**
1.1 The Contribution of this work 1
1.2 Organization of the Thesis 2

2 CHF Predictive Model Development **3**
2.1 CHF mechanisms 3
2.1.1 CHF at low quality (Departure from Nucleate Boiling (DNB) 4
2.1.2 CHF at High Quality (Dryout) 6
2.2 CHF Empirical models 7
2.2.1 Parameters that affect CHF 7
2.2.2 The 2006 CHF Look-Up table 7
2.3 The prediction of CHF for subcooled flow boiling 9
2.3.1 Vapour Blanket Analysis 10
2.3.2 Axial distribution of bulk temperature and void fraction in a heated channel with inlet subcooling 12

3 Machine Learning to predict Critical Heat Flux **14**
3.1 The Artificial Neural Network 14
3.1.1 The simplest part: the perceptron 15
3.2 ANN classification and the Feed-Forward Neural Network 16
3.3 State of the Task Force and Case Study 17

4 The construction of the model **19**
4.1 CHF Database 23
4.1.1 Feature extraction and feature selection 24
4.1.2 Analysis of Linear Relationships and R^2 Coefficient Between Input Variables and CHF 27
4.2 Training, Validation and Test 30
4.2.1 Normalization of data 30
4.3 Problem of overfitting 31
4.4 Train and execution of the Neural Network 31

4.4.1 Performance . . . 33
4.4.2 plottrainstate . . . 34
4.4.3 Error Histogram . . . 35
4.4.4 Plot regression . . . 36
4.5 Post-processing . . . 37
4.5.1 CHF LUTs Results . . . 37
4.5.2 NN results . . . 38
4.5.3 Slice division of data set . . . 46

5 The split model: a way to improve the effectiveness of the neural network 50
5.1 Filtering and specialization of the Neural Network . . . 50
5.1.1 The number of layer hidden . . . 51
5.2 Split model with training and validation sets . . . 55
5.3 Split model with training. validation and test sets . . . 60
5.4 Slice division of data set . . . 65

6 The Log-transformed Neural Network 68
6.1 Results . . . 71
6.2 Final comparison between the three models . . . 74

7 Conclusion and perspective 77

A RMSE e MAE for the original model 81
A.1 Model with training and validation sets . . . 82
A.2 Model with entire set . . . 84

B Configurations for P/M ratio threshold 86

C RMSE e MAE for split model 89
C.1 Split model with training and validation set - filtered points . . . 90
C.2 Split model with training and validation set - remaining points . . . 92
C.3 Split model with training, validation and test set - filtered points . . . 94
C.4 Split model with training, validation and test set - remaining points 96

D RMSE e MAE for the log-net model 98

List of Figures

2.1 CHF mechanisms. (a) Dryout, (b) DNB [5] 4
2.2 Effect of heat flux spike on $q"_{cr}$ [1] 5
2.3 Effect of mass flux on critical heat flux [1] 6
2.4 Conceptual view of liquid sublayer dryout mechanism [2] 9
2.5 Schematic representation of a stable vapour blanket 10
2.6 Scheme of the three regions of heated channel [3] 12

3.1 Perceptron model [4] 15
3.2 Sketch of a step function 15
3.3 framework for artificial neural networks classification [5] 16
3.4 Sketch of Neuron Network Method [6] 16

4.1 Flow chart of the NN 20
4.2 Flowchart of ensemble iterations 21
4.3 Flow chart of best net choice iterations 22
4.4 Scatter plot matrix of the NRC CHF database showing the relationship between pairs of variables [7]. 23
4.5 Pearson Correlation Matrix 25
4.6 Spearman Correlation Matrix 26
4.7 Graphs showing the difference between underfitting (on the left), good fit and overfitting (on the right) [8] 31
4.8 Neural network training dashboard 32
4.9 Plot of the performance of training of the neural network 33
4.10 Evolution of gradient (top figure) and validation checks (bottom figure) 34
4.11 Plot of the error histogram 35
4.12 Plot of network regression 36
4.13 Measured vs LUTs predicted CHF [7] 37
4.15 RMSE and MAE percentage error distribution for neural network with training and validation set 41

4.16 RMSE and MAE percentage error distribution for neural network with rain, Validation and Test set 41
4.17 LUTs (blue) and NN (orange) Predicted versus Measured CHF scatter plots, versus selected independent parameters, on the left we have the NN model with only training and validation sets, on the right we have the "complete" model 45
4.18 Comparison of the model and LUTs behavior as a function of different parameters. 49

5.1 CHF database with points where the $\frac{P}{M}$ ratio is greater than the 20% of threshold value 51
5.2 Comparison of the different configurations based on the total number of neurons 52
5.3 Comparison of the different configurations based on the total number of neurons (4 layers) 53
5.4 Comparison of the different configurations based on the total number of neurons (3 layers) 53
5.5 Measured vs predicted CHF for a NN with only train and validation data subdivision 55
5.6 RMSE and MAE percentage error distribution for filtered points of database of the split network with training and Validation set . . . 57
5.7 RMSE and MAE percentage error distribution for remaining points of database of the split network with training and Validation set . . 57
5.8 Relative error difference between the original NN model and the split one with training and Validation set 58
5.9 P/M Ratio vs Tube Diamete 59
5.10 P/M Ratio vs Heated Length 59
5.11 P/M Ratio vs Pressure 59
5.12 P/M Ratio vs Mass Flux 59
5.13 P/M Ratio vs Outlet Quality 59
5.14 Original NN (orange) and split NN (yellow) Predicted over Measured CHF scatter plots, vs selected independent parameters. NN model with only training and Validation 59
5.15 Measured vs predicted CHF for a NN with training validation and test subdivision 60
5.16 RMSE and MAE percentage error distribution for remaining points of database of the split network with training and Validation set . . 62
5.17 RMSE and MAE percentage error distribution for remaining points of database of the split network with training and Validation set . . 62
5.18 Relative error difference between the original NN model and the split one with training, Validation and Test 63

5.19 P/M Ratio vs Tube Diameter 64
5.20 P/M Ratio vs Heated Length 64
5.21 P/M Ratio vs Pressure 64
5.22 P/M Ratio vs Mass Flux 64
5.23 P/M Ratio vs Outlet Quality 64
5.24 Original NN (orange) and split NN (yellow) Predicted over Measured CHF scatter plots, vs selected independent parameters. NN model with training, Validation and Test 64
5.25 Comparison of the model and LUTs behaviour as a function of different parameters 67

6.1 Distribution of heated length values 69
6.2 Distribution of pressure values 69
6.3 Distribution of mass flux values 70
6.4 Distribution of CHF values 70
6.5 Measured vs predicted CHF for the log-net with train validation and test subdivision 71
6.6 RMSE and MAE percentage error distribution for neural network with Train, Validation and Test set 73
6.7 Relative error difference between the original NN model and the log-transformed one with training, validation and test sets 73
6.8 Network models final metrics comparison 75
6.9 P/M Ratio vs Tube Diameter 76
6.10 P/M Ratio vs Heated Length 76
6.11 P/M Ratio vs Pressure 76
6.12 P/M Ratio vs Mass Flux 76
6.13 P/M Ratio vs Outlet Quality 76
6.14 Original NN (orange), split NN (yellow) and log-net (green) Predicted over Measured CHF scatter plots, vs selected independent parameters. NN model with Train, Validation and Test 76

A.1 RMSE and MAE percentage error distribution for neural network with training and validation set (Part 1) 82
A.1 RMSE and MAE percentage error distribution for neural network with training and validation set (Part 2) 83
A.2 RMSE and MAE percentage error distribution for neural network with training, validation and Test set (Part 1) 84
A.2 RMSE and MAE percentage error distribution for neural network with training, validation and Test set (Part 2) 85

B.1 CHF database with points where the $\frac{P}{M}$ ratio is greater than the 10% of threshold value 87
B.2 CHF database with points where the $\frac{P}{M}$ ratio is greater than the 30% of threshold value 87
B.3 CHF database with points where the $\frac{P}{M}$ ratio is greater than the 40% of threshold value 88
B.4 CHF database with points where the $\frac{P}{M}$ ratio is greater than the 50% of threshold value 88

C.1 RMSE and MAE percentage error distribution for neural network with training and validation set (Part 1) 90
C.1 RMSE and MAE percentage error distribution for neural network with training and validation set (Part 2) 91
C.2 RMSE and MAE percentage error distribution for neural network with training and validation set (Part 1) 92
C.2 RMSE and MAE percentage error distribution for neural network with training and Validation set (Part 2) 93
C.3 RMSE and MAE percentage error distribution for neural network with training and Validation set (Part 1) 94
C.3 RMSE and MAE percentage error distribution for neural network with training and Validation set (Part 2) 95
C.4 RMSE and MAE percentage error distribution for neural network with training and Validation set (Part 1) 96
C.4 RMSE and MAE percentage error distribution for neural network with training and Validation set (Part 2) 97

D.1 RMSE and MAE percentage error distribution for log-net with training, Validation and Test set (Part 1) 98
D.1 RMSE and MAE percentage error distribution for log-net with training, Validation and Test set (Part 2) 99

List of Tables

3.1 Task Force Timeline [7] 18

4.1 Parameter spans of the NRC CHF database [7] 23
4.2 Pearson's Linear Correlation Coefficient for various parameters . . . 25
4.3 Spearman Correlation Coefficients 26
4.4 Linear regression model between the whole group of inputs and the CHF 28
4.5 Statistical Summary 28
4.6 Linear regression model between CHF and the most influencing inputs 28
4.7 Statistical Summary 29
4.8 Linear regression model (considering Rank of matrix) between the whole group of inputs and the CHF 29
4.9 CHF LUTs prediction performance [7] 37
4.10 NN with only training and Validation prediction performances . . . 43
4.11 NN with training, validation and Test prediction performances . . . 43
4.12 Slice data sets [7] 46

5.1 split NN model with only training and Validation prediction performances 56
5.2 split NN model with training, Validation and Test prediction performances 61

6.1 Performance Evaluation for TEST and ALL data sets 72

Chapter 1

The importance of Safety in Nuclear Energy

Geopolitical events and the ecological transition support government to sign sustainable politics towards to clean sources of energy. In this case, nuclear energy plays a fundamental role, as it is the real competitor to oil, gas and coal, and gives a great deal of support to renewable energy sources. But public concern about nuclear risk is an important issue, especially after the Three Miles Island, Chernobyl and Fukushima accidents and the war in Ukraine, although statistical data show that it is extremely low compared to human activities. **Safety** is a key driver in the design and operation of any nuclear plant, with its excellent records in recent years, result of the attention given by industry and authorities (such as the Nuclear Regulatory Commission, NRC, and the International Atomic Energy Agency, IAEA) to it, protecting people and environment from the release of radioactive and fission products [9]. One of the aspects of nuclear safety depends on the effective management of heat generated by fission reactions. When inefficiencies happen, we can have significant operational issues. For this reason, it is important to describe the concept of **Critical Heat Flux, CHF**. It defines **the maximum heat flux at which the coolant can remove heat from the core without boiling, preventing overheating and potential damage**. Ensuring that reactor operates within this thermal design limit is so important for prevention and mitigation.

1.1 The Contribution of this work

The aim of this Thesis is to make a significant contribution to reactor safety, in particular on the core design. As said previously, the CHF can cause deterioration of the heat transfer in the core of nuclear reactors, potentially leading to damage,

with a consequent releasing of fission products into the environment. So its accurate prediction, although is tricky, is a crucial issue for safety of nuclear plant. Lots of prediction models have been conducted during past years, depending on the study and comprehension of the heat transfer mechanism inside the reactor. With the development of first prototypes of artificial intelligence (AI) the research approach changes, enhancing the works of technicians. In this case study, after exploring artificial intelligence and understanding the features of each methodology, a new model, based on neural networks, has been conducted and its performance is compared to the preexistent ones. Furthermore, two possible evolution of the model has been developed in way to emphasize the pros and cons of each evolution of the net. The final result must demonstrate how the use of artificial intelligence is a fundamental instrument that can improve the predictive safety not only for mitigate the risks, but also to improve reactor efficiency.

1.2 Organization of the Thesis

In Chapter 2 we have a theoretical description of the CHF with its mechanisms inside a Pressurised Water Reactor (PWR), focusing on the different models that predict it. In Chapter 3 there is a brief description review about the artificial intelligence, focusing on the feed-forward neural network, model adopted for this work and the description of the Task Force that gives support to this work. Chapter 4 is the central part, since it describes the steps methodology how the net is built and the performance. Chapter 5 and 6 present two possible scenarios for improving the neural network: the first is an adaptation of the network to the points of the database where the performance is poor. The second shows a small change in the construction of the net. An overall comparison was then made. In the last chapter we have a summary of the whole work, with some suggestions for future activities.

Chapter 2

CHF Predictive Model Development

2.1 CHF mechanisms

The concept of CHF is used to describe the conditions under which the heat transfer coefficient of the two-phase flow deteriorates significantly. In a system where the heat flux is independently controlled, the consequences of the CHF are a rapid rise in the wall temperature. For systems in which the wall temperature is independently controlled, the occurrence of CHF implies a rapid decrease in the heat flux. [1]. Dynamic changes in the boiling regime associated with the exceeding of the critical heat flux are known as **boiling crisis** and it can be classified as 2.1:

- Dryout: term now used for the drying out of a liquid film in a annular flow;
- Departure from Nucleate Boiling **(DNB)**: used to describe the CHF conditions in pool boiling, or when the bubble formation is rapid enough to cause a continuous vapour film to form on the wall in flow boiling.

Understanding the physics behind these two mechanisms is important for CHF prediction, as it enhances safe operating condition under certain ranges.

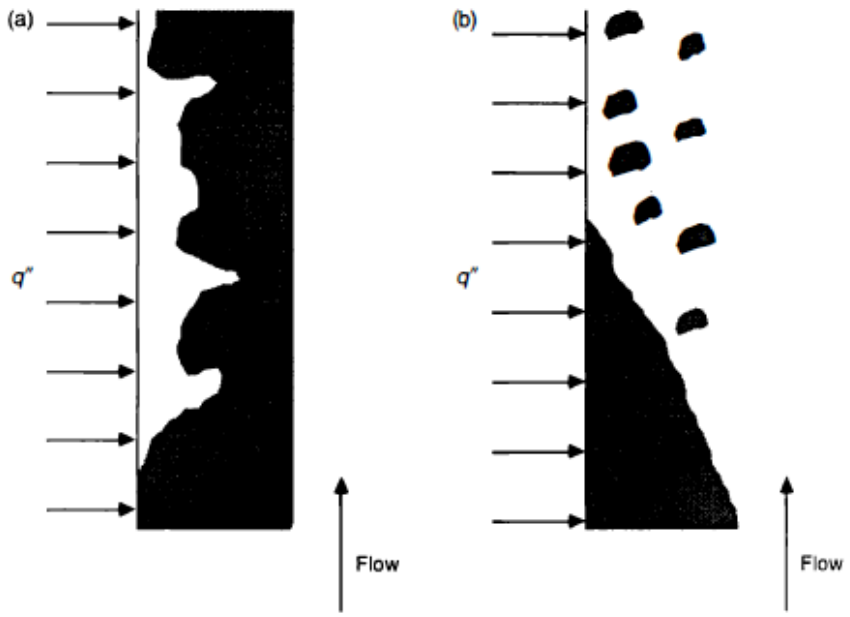


Figure 2.1: CHF mechanisms. (a) Dryout, (b) DNB [5]

2.1.1 CHF at low quality (Departure from Nucleate Boiling (DNB)

1 Pool boiling. The earliest studies of CHF involved placing a heating surface in a static liquid pool. The observed heat flow - wall temperature behaviour shows a CHF condition leading to transition boiling. This pool boiling CHF provides lower heat fluxes than does flow boiling, because the critical heat flux of the DNB type increases with mass flux. This CHF mechanism (DNB) can occur for both saturated and subcooled liquid conditions. [1]
2 Flow boiling. The nucleate regime exists at low quality flow conditions. If the heat flux is very high, vapour blanketing of the surface can occur. It is possible to set two limits for the CHF ($q"_{cr}$). At the lower end, it must be high enough to cause the wall temperature to reach saturation ($T_W = T_{sat}$) while the bulk flow is still undercooled. Hence:

$$(q"_{cr})_{min} = h_{lo}(T_w - T_{bulk}) = h_{lo}(T_{sat} - T_{bulk}) \tag{2.1}$$

For axial uniform heat flux:

$$(q"_{cr})_{min}\pi DL = \frac{G\pi D}{4}c_p(T_{bulk} - T_{in}) \tag{2.2}$$

where L = tube length. Therefore eliminating T_{bulk}

$$(q"_{cr})_{min} = \frac{(T_{sat} - T_{in})}{\frac{1}{h_{lo}} + \frac{4L}{GDc_p}} \tag{2.3}$$

At the high end, $q"_{cr}$ should be sufficiently high to cause an equilibrium density ($\chi_e = 1.0$). Hence:

$$(q"_{cr})_{max} = \frac{GD}{4L}[c_p(T_{sat} - T_{in} + h_{fg}] \tag{2.4}$$

From these equations it is clear that the higher the inlet temperature (or thermodynamic quality) the lower is the value of $q"_{cr}$ in a uniformly heated tube. Generally it would be expected that:

$$q"_{cr} = q"_{cr}[p, G, D, L, (\Delta T_{sub}), q"(z)] \tag{2.5}$$

The dependence on the inlet subcooling and the heat flux profile may be replaced

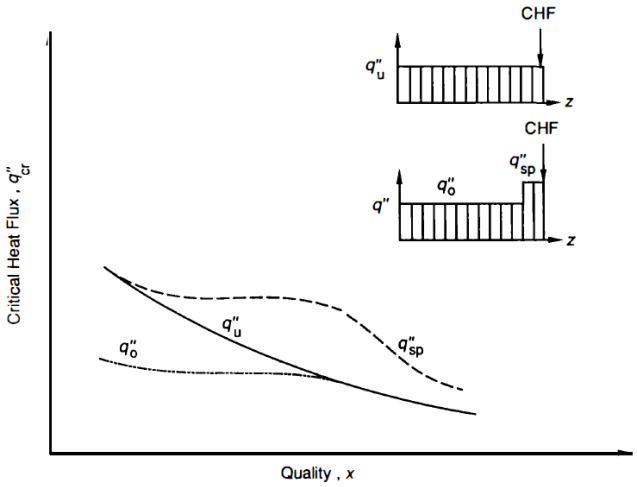


Figure 2.2: Effect of heat flux spike on $q"_{cr}$ [1]

by a dependence on the local value of the quality at the point of CHF (i.e., at L):

$$q"_{cr} = q"_{cr}[p, G, D, L, x(L)] \tag{2.6}$$

It should be noted that the vapour blanketing is mainly influenced by the local evapouration rate and therefore the effect of heating time on DNB is small. However, for the dryout form at CHF, the hydrodynamic effects on the liquid film behaviour are pronounced. The development of a particular form of hydrodynamic behaviour is influenced by the length of the flow [1].

2.1.2 CHF at High Quality (Dryout)

At high quality flow conditions, the vapour is mostly present in the core. Both the shear action of the vapour and the local evaporation rate can cause the liquid film to be stripped from the wall. In uniformly heated tubes, the value of $q"_{cr}$ at high flow qualities is, roughly speaking, lower than that at low flow qualities. It is also observed that $q"_{cr}$ is lower for lunger tubes. However, the total critical power input increases with length. Therefore the critical quality (x_{cr}) at the tube outlet also increases with length. Due to the difference between the DNB and the dryout mechanisms, they show an inverse dependence on the mass flux. As shown in the figure below, the higher mass flux leads to a higher $q"_{cr}$ for low quality flow, but decreases the $q"_{cr}$ for high quality flow. Total critical power in an evenly heated tube of fixed length L increases as mass flux increases. For high quality CHF, the critical channel power is not sensitive to the distribution of the heat flux along a fixed length [1].

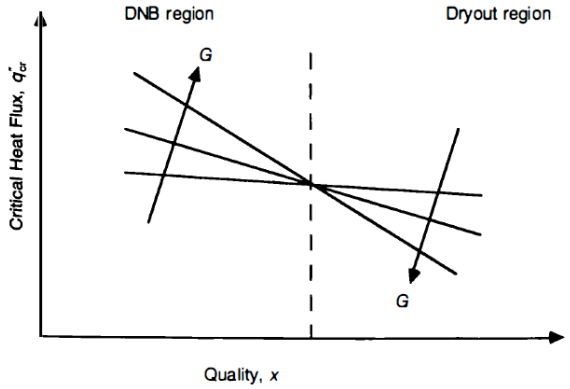


Figure 2.3: Effect of mass flux on critical heat flux [1]

We have three different approached to predicting CHF:

1. CHF correlations;
2. CHF Look-Up table (LUTs);
3. mechanistic model.

2.2 CHF Empirical models

2.2.1 Parameters that affect CHF

Traditionally, CHF predictions have been based on empirical models developed from extensive experimental data sets that capture the influence of operating parameters. The primary independent variables affecting CHF are: tube internal diameter *(D)*, tube length *(L)*, system pressure *(P)*, inlet flow rate or mass velocity *(G)*, quality *(X)* and the inlet temperature or inlet subcooling $(\Delta T_{sub,i} = T_{sat} - T_i)$.

$$q_c = f(G, \Delta T, p, D, L)$$

Secondary independent variables influencing the CHF can be the flow orientation, the material and roughness of the pipe surface and the method of heating, or the presence of cold walls. If the heat flux is uniform, overheating of the tube surface at CHF almost always begins at the tube exit. CHF can also be expressed as a function of the outlet conditions: instead of the inlet subcooling, either the specific enthalpy at the outlet or the outlet quality can be used as an independent parameter.

2.2.2 The 2006 CHF Look-Up table

The complexity of predicting the CHF increases significantly when additional factors such as transients, non-uniform flux distributions, and asymmetric cross sections are introduced. This has led to the development of the CHF Look-up table, adopted in 2006. The CHF Look-Up table method has many advantages over other CHF prediction methods:

- simple to use;
- no iterations required;
- wide range of applications;
- based on a very large database;
- eliminates the need to choose between among the many CHF prediction methods currently available for water-cooled tubes.

Engineers can estimate CHF interpolating between reference data without using iterative calculations. However, with the increasing complexity of reactor designs and the drive for higher efficiencies, the limitations of traditional models have become more apparent, particularly as they may not fully adapt to advanced operating conditions. Traditional Look-Up Tables (LUTs) have certain limitations:

- **Static and inflexible**: LUTs are inherently static, as they depend on predefined experimental parameters. This limits their adaptability when applied conditions beyond the reference scope, such as novel reactor designs or extreme operating environments.
- **Limited applicability**: The applicability of LUTs is often limited to specific ranges. For example, the Groeneveld table is only accurate within certain pressure and temperature ranges, making it less effective outside its original operational scope.
- **Scalability and Adaptability**: As reactor designs and configurations evolve, LUTs may not scale or adapt adequately, requiring frequent updates of experimental data to maintain predictive accuracy

These limitations have paved the way for the integration of artificial intelligence techniques to further refine CHF predictions [10].

2.3 The prediction of CHF for subcooled flow boiling

This model has since been developed for its accuracy and feasibility. It is based on the concept of the **subcooled flow boiling**. Initially we have a subcooled liquid ($T_{in} < T_{sat}$ entering the channel. At a certain point we have the formation of vapour bubbles as the liquid becomes superheated at the wall. This is the **subcooled boiling**. Then the bubbles agglomerate into larger ones and finally they form a vapour blanket that divides the flow region into two parts: a near-wall region and the core region, as shown in the figure below [2].

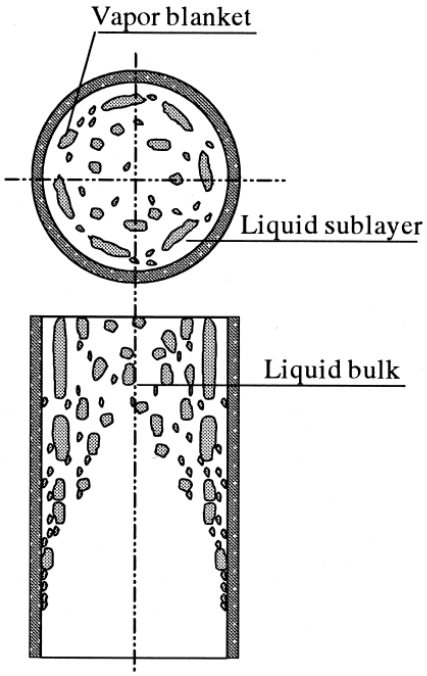


Figure 2.4: Conceptual view of liquid sublayer dryout mechanism [2]

The CHF occurs when the liquid sublayer is completely dry.

$$CHF = \frac{\rho_f \delta H_{fg}}{L_B} U_B \tag{2.7}$$

where U_B, L_B and δ are the vapour blanket velocity, vapour blanket length and thickness of liquid layer thickness respectively.

2.3.1 Vapour Blanket Analysis

First, we consider the flow area divided by the vapour blanket, as shown in Figure 2.5. The two waves correspond to the interfaces, and their wavelengths are equal to those predicted by the Helmholtz instability at each interface. This assumption ensures that a stable blanket contains only one full wavelength. In this way, the vapour blanket velocity can be written as a simple function of the average velocity of the two-phase flow in the core region.

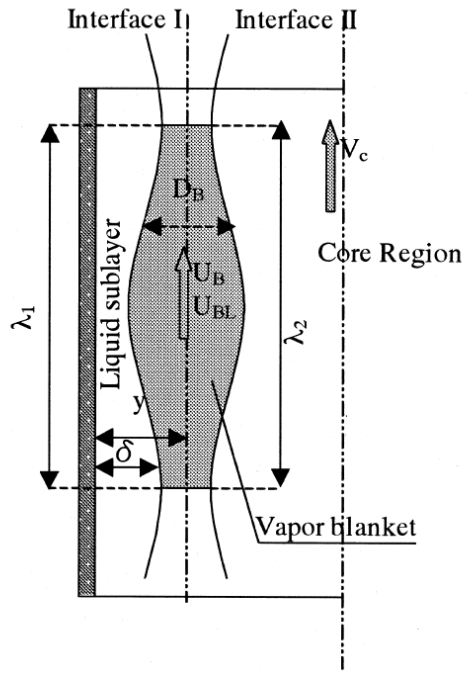


Figure 2.5: Schematic representation of a stable vapour blanket

Thanks to these assumptions we can evaluate the blanket velocity;

$$U_B = \frac{V_c}{1+b} \tag{2.8}$$

where b is

$$b = \sqrt{\frac{(\rho_c + \rho_g)}{\rho_c}} \tag{2.9}$$

The diameter of the vapour blanket is:

$$D_B = 0.015 \left(\frac{\sigma D}{\tau_w} \right)^{0.5} \tag{2.10}$$

where

$$\tau_w = \frac{fG^2}{8\rho_f} \tag{2.11}$$

All of these parameters are important for the friction factor, calculated by the Colebrook equation [2][11]:

$$\frac{1}{\sqrt{f}} = 1.14 - 2.0\log\left(\frac{\varepsilon}{D} + \frac{9.35}{Re\sqrt{f}}\right) \tag{2.12}$$

The velocity of the vapour blanket can be obtained by a rearrangement of force balance equation:

$$U_{BL} = U_B - \left(\frac{2L_B g(\rho_f - \rho_g)}{\rho_f C_D}\right)^{0.5} \tag{15}$$

Drag coefficient C_D can be obtained either by Harmathy or Chan and Prince expressions. The former determined by buoyancy and surface tension forces is recommended in the present model at low pressure ($P < 1$ MPa). The latter one, proposed for small bubbles dominated by viscous forces, is recommended at medium and high pressures ($P \geq 1$ MPa).

Harmathy:

$$C_D = \frac{2}{3}\frac{D_B}{\left(\frac{\sigma}{g(\rho_f - \rho_g)}\right)^{0.5}} \tag{2.13}$$

Chan and Prince:

$$C_D = \frac{48\mu_f}{\rho_f D_B(U_B - U_{BL})} \tag{2.14}$$

If the CHF occurs at the tube exit, then V_c and ρ_c are the core region two-phase average velocity and average density at the tube exit, respectively.

$$V_c = \frac{G}{\rho_c} \tag{2.15}$$

$$\rho_c = (1 - \alpha_c)\rho_{lout} + \alpha_c \cdot \rho_g \tag{2.16}$$

where α_c and ρ_{lout} are the exit core region void fraction and liquid density, respectively. For simplicity, we assume $\alpha_c = \alpha_{lout}$; the latter is obtained by the **Ahmad model**.

$$\alpha_c = \frac{\chi_{out}}{\chi_{out} + (\frac{\rho_g}{\rho_f})S(1 - \chi_{out})} \tag{2.17}$$

where S is the slip ratio

$$S = \left(\frac{\rho_f}{\rho_g}\right)^{0.205}\left(\frac{GD}{\mu_f}\right)^{-0.016} \tag{2.18}$$

where χ_{out} is the exit true quality and can also be calculated from the Ahmad model described below [2].

2.3.2 Axial distribution of bulk temperature and void fraction in a heated channel with inlet subcooling

The proposed Ahmad model [12] satisfactorily correlates the measured bulk temperature profiles. It plays a fundamental role in the evaluation of void fraction and quality at the net vapour generation (NVG) point, where bubble detachment from the wall occurs. It first divides the entire length of the tube into three regions as shown in figure 2.6[3]:

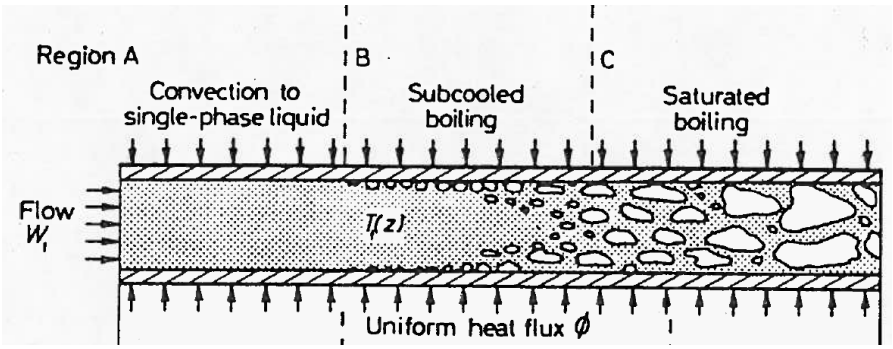


Figure 2.6: Scheme of the three regions of heated channel [3]

1. Single phase region: where all the heat flux is used to raise the liquid bulk temperature and no vapour is formed, until the nucleation.

2. Transitional region or subcooled boiling: where bubbles are formed but do not grow large enough to detach from the wall and subsequently collapse at the surface.

3. Bulk boiling: where the detachment occurs.

From this scheme we can obtain the *significant boiling length* Z_{sb}

$$Z_{sb} = L - Z_0 \tag{2.19}$$

where L is the total heated length and Z_0 is the corresponding position of the onset of bubble detachment.

$$Z_0 = GDC_p \frac{\Delta T_{in} - \Delta T_d}{4q} \tag{2.20}$$

.
Here special attention is given to ΔT_d, which is related to the temperature at which onset of bubble detachment begins.

$$\Delta T = \frac{q}{h_{1-A}} \tag{2.21}$$

where q is the *quenching heat flux* and h_{1-A} the *liquid phase heat transfer coefficient*:

$$h_{1-A} = 2.44\frac{k_f}{D}\left(\frac{GD}{\mu_f}\right)^{\frac{1}{2}}\left(\frac{C_p\mu_f}{k_f}\right)^{\frac{1}{3}}\left(\frac{H_{lin}}{H_f}\right)^{\frac{1}{3}}\left(\frac{H_{fg}}{H_f}\right)^{\frac{1}{3}} \tag{2.22}$$

In some cases the value of ΔT may exceed ΔT_{in}: this means that the bubble detachment occurs from the beginning of the heated length. So we set $\Delta T_d = \Delta T_{in}$ After developing the temperature distribution, the main objective is to predict the **void fraction**. To do this, it is necessary first to find the true vapour weight quality and the slip ratio. For the first one we use two dimensionless parameters:

$$A = \frac{q^* Zsb}{GDC_{pl}\Delta T_d/4} \tag{2.23}$$

A is the ratio of the heat absorbed by the liquid from the NGV point to the tube outlet to the total heat required to bring the liquid at the NGV to saturation.

$$B = \frac{h_{fg}}{C_{pl}\Delta T_d} \tag{2.24}$$

From this we have:

$$\chi_{eqout} = \frac{A-1}{B} \tag{2.25}$$

$$\chi_d = -\frac{1}{B} \tag{2.26}$$

$$\chi_{\text{out}} = \frac{\chi_{\text{eqout}} - \chi_d \exp\left(\frac{\chi_{\text{eqout}}}{\chi_d} - 1\right)}{1 - \chi_d \exp\left(\frac{\chi_{\text{eqout}}}{\chi_d} - 1\right)} \tag{2.27}$$

After all of this procedure we can calculate the CHF [12].

Chapter 3

Machine Learning to predict Critical Heat Flux

Traditional methods explained earlier, such as empirical correlations and LUTs are widely used, but show problems as needing constant updating to mantain accuracy and efficiency to new reactor designs. With advances in computing power, nuclear research has increasingly focused on machine learning as a new way to develop more sophisticated and accurate CHF prediction models. Machine learning methods, particularly neural networks and other advanced algorithms, offer a way to handle large data sets and uncover underlying patterns that may be missed by traditional approaches. This modern approach allows for the construction of models that adapt to different operational parameters, ultimately providing a more refined and responsive tool for CHF prediction [13].

This chapter briefly describes the characteristics of Artificial Neural Networks (ANNs) with a focus on Feed-Forward Neural Networks (FFNNs). These have been chosen for their ability to approximate complex functions and handle large data sets.

3.1 The Artificial Neural Network

Artificial Neural Networks (ANNs) are a type of model of Machine Learning that is becoming so competitive. It can be evaluated with respect to data analysis on several factors like accuracy, processing speed, performance, convergence and other, ANNs are used for their excellent response to solve the problems since they are so fast, adaptive and good self-learning response [5]. ANN can be compared to our brain for its unique design of information processing and capability. Inside the brain we have the "neurons" that carry out human activities by receiving and sending signals between each other.

3.1.1 The simplest part: the perceptron

The modern neural network is based on a simple structure that is the perceptron, which was designed to illustrate the fundamental properties of intelligent systems. In general it consists of a single a neuron capable of processing input synaptic signals by producing an output signal corresponding to the inhibition or excitation states. Synaptic signals are represented by the components of an input vector as real values. They are linearly combined through the use of synaptic weights to converge, together with a threshold value, to a function non-linear function[4].

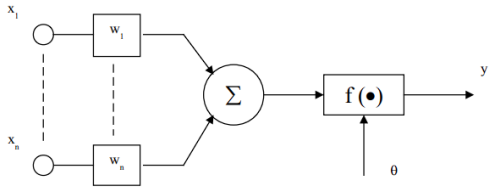


Figure 3.1: Perceptron model [4]

The output y in figure 3.1 is expressed by

$$y = f\left(\sum_{i=1}^{n} w_i x_i - \theta\right) \tag{3.1}$$

where $x = (x_1, x_2, ..., x_n)$ is the vector with input values, $w = (w_1, w_2, ..., w_n)$ are the weigths and they can be fixed or modified during training phase. θ is the offset threshold. $f(\cdot)$ is the activation function, characterized by a step function.

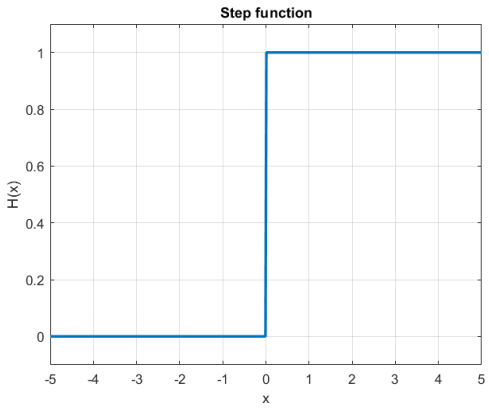


Figure 3.2: Sketch of a step function

During the training, couple of input/output (x, y) are send to the perceptron and the goal is to minimize the error function (3.2), updating each iteration the

weigths.

$$e(t) = y(t) - g(t) \tag{3.2}$$

where $g(t)$ is the perceptron response at a generic time t. When the error function is lower than a certain threshold, the training step is finished [4].

3.2 ANN classification and the Feed-Forward Neural Network

ANN can be classified into several model like in figure 3.3, but we are mainly interested in the Feed-forward neural network (FFNN)

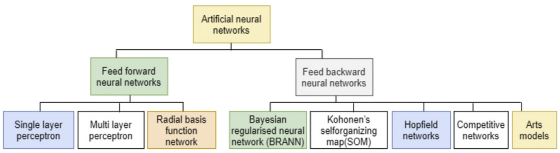


Figure 3.3: framework for artificial neural networks classification [5]

The Feed-Forward Neural Network is made up of a collection of multilayers (at least three: input, hidden layer and output), where each unit in a layer communicates with all the others. The connection between these layers are not equal, since there are weights and biases randomly chosen from a uniform distribution. The weights measure the potential amount of the knowledge in the network. In FFNN, the information is only transmitted in one direction, from input to output.

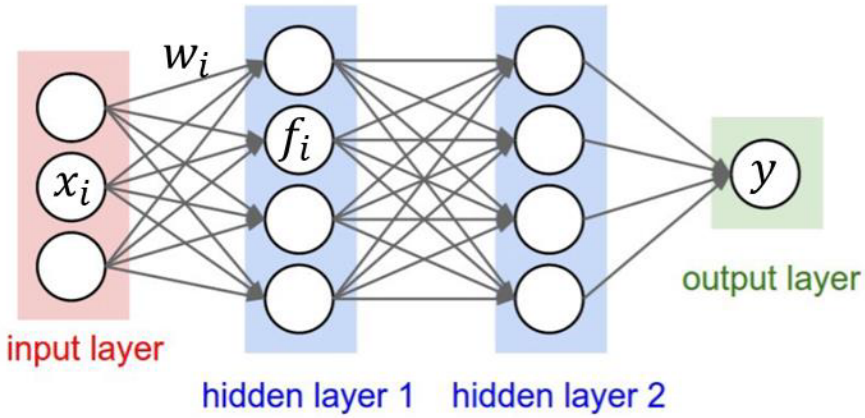


Figure 3.4: Sketch of Neuron Network Method [6]

The architecture of a neural network is shown in 3.4. There are 3 input units, 2 hidden layers and 1 output unit with the connections between them. The hidden layers are called so called because they are not visible from either the input or the output, although they are essential for the functioning of the model. In fact, these layers introduce a certain complexity in the patterns that the network is able to analyse, thus extend the applications to non-linear mapping functions.

3.3 State of the Task Force and Case Study

The role of AI in nuclear engineering has led the development of a new research approach aimed at integrating Machine Learning models into safety applications. The Working Party on Scientific Issues and Uncertainty Analysis of Reactor Systems (WPRS) analyses all the components of a nuclear plant and its Expert Group on Reactor Systems Multiphysics (EGMUP) has set up a Task Force on Artificial Intelligence and Machine Learning for Scientific Computing in Nuclear Engineering in 2022. The aim is to develop a benchmark that provides guidelines for the use of artificial intelligence and Machine Learning to [7]:

- improve the know-how in the field of AI and ML applications;
- contribute to the development and performance evaluation of ML methods;
- to provide experience with these models and provide guidelines for future applications in nuclear engineering.

This activity is divided into two phases[7]:

1. Phase 1: Regression, Classification and Verification, Validation and Uncertainty Quantification (VVUQ);
2. Phase 2: Generative Deep Learning and Data Augmentation; Design Optimization

The main activities are expected to respect the following schedule:

Event	**Data**
CHF benchmark introduction at TF meeting	December 2022
Phase 1 draft specification and distribution	May 2023
Presentation at 2023 OECD/WPRS Annual Workshops	May 2023
Phase 1 final specifications and distribution	September 2023
Phase 1 online kickoff meeting	October 2023
Phase 1 online Q&A meeting (optional)	December 2023
Phase 2 draft specifications and distribution	May 2024
Presentation at 2024 OECD/WPRS Annual Workshops	May 2024
Phase 1 submission	August 2024
Phase 2 final specifications and distribution	September 2024
Phase 2 online kickoff meeting	October 2024
Phase 1 results draft report and online meeting	December 2024
Phase 2 (fuel bundle) draft specifications and distribution	May 2025
Presentation at 2025 OECD/WPRS Annual Workshops	May 2025
Phase 2 submission	August 2025
Phase 2 (fuel bundle) final specifications and distribution	September 2025
Phase 2 (fuel bundle) online kickoff meeting	October 2025
Presentation at 2026 OECD/WPRS Annual Workshops	May 2026
Phase 2 (fuel bundle) submission	August 2026

Table 3.1: Task Force Timeline [7]

This work belongs to Phase 1, since the aim is to develop a neural network model to make the CHF that can predict better than the empirical methods. In the next chapter the entire algorithm is described, with attention to each step that must be followed to obtain a good result.

Chapter 4

The construction of the model

Having explained the most important feature of a neural network, we now move on the practical part, the construction of a neural network model for predicting CHF. A brief summary of the sequential instructions, drawn in 4.1 for the NN model used in this study is described here as a small guide for what will be explained below. First, an overall framework of the algorithm is presented, followed by a detailed breakdown of the execution steps within the network. As explained in Chapter 3, the "neurons" of a network need the input unit. So we read the CHF database from the Task Force [7] and divide it into training, validation and test sets. A small note: we make two different models, one with only training and validation and the other with the whole set. This division is important for the training phase of the network. Then we pre-allocate the metrics, because we don't have just one output, but we simulate our network 100 times. So we have an ensemble of networks and it is useful to get a consistent result and we can do metrics analysis. Going into each of these 100 iterations, we have an internal scheme, described by 4.2: we have a permutation of the index of the sets and the successive normalization of the input variables. Then we have the construction and the training of the network. Here there is the choice of the best net, based on the best performance. So, before the execution, we make another pre-allocation of the metrics related to the specific net. After 5 iterations (4.3) we choose the best one, and we record the metrics for the specific net, and the iterations go on.

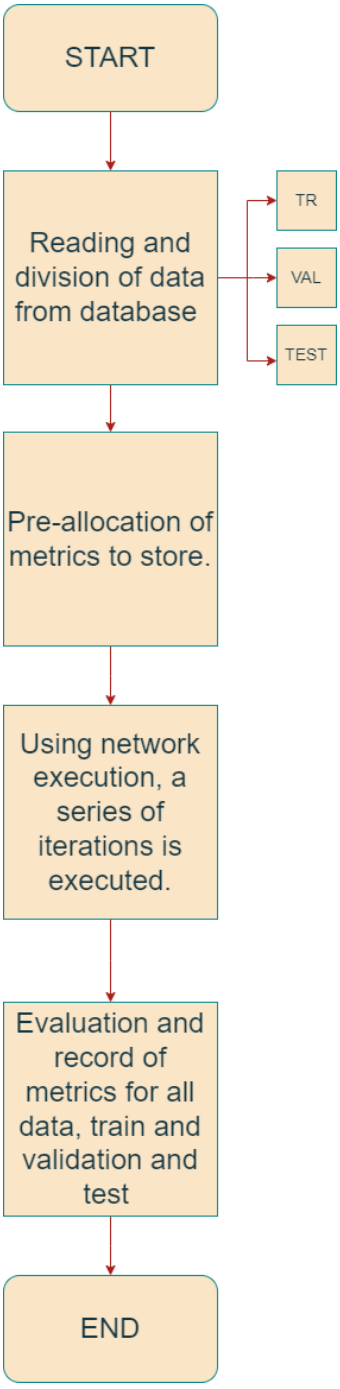


Figure 4.1: Flow chart of the NN

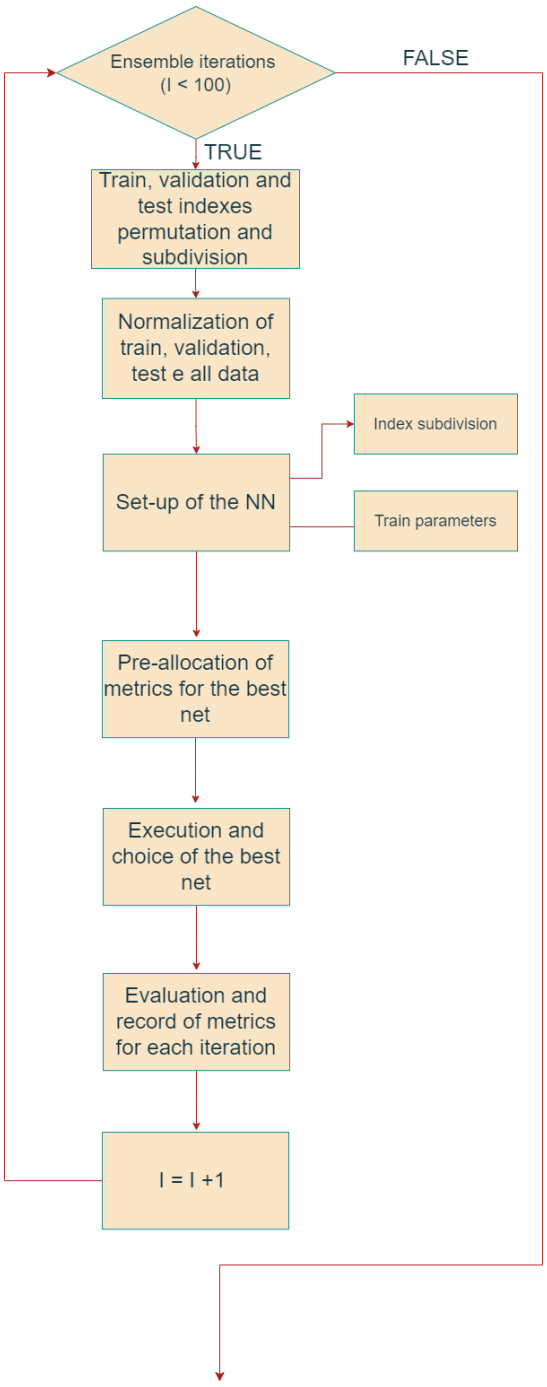


Figure 4.2: Flowchart of ensemble iterations

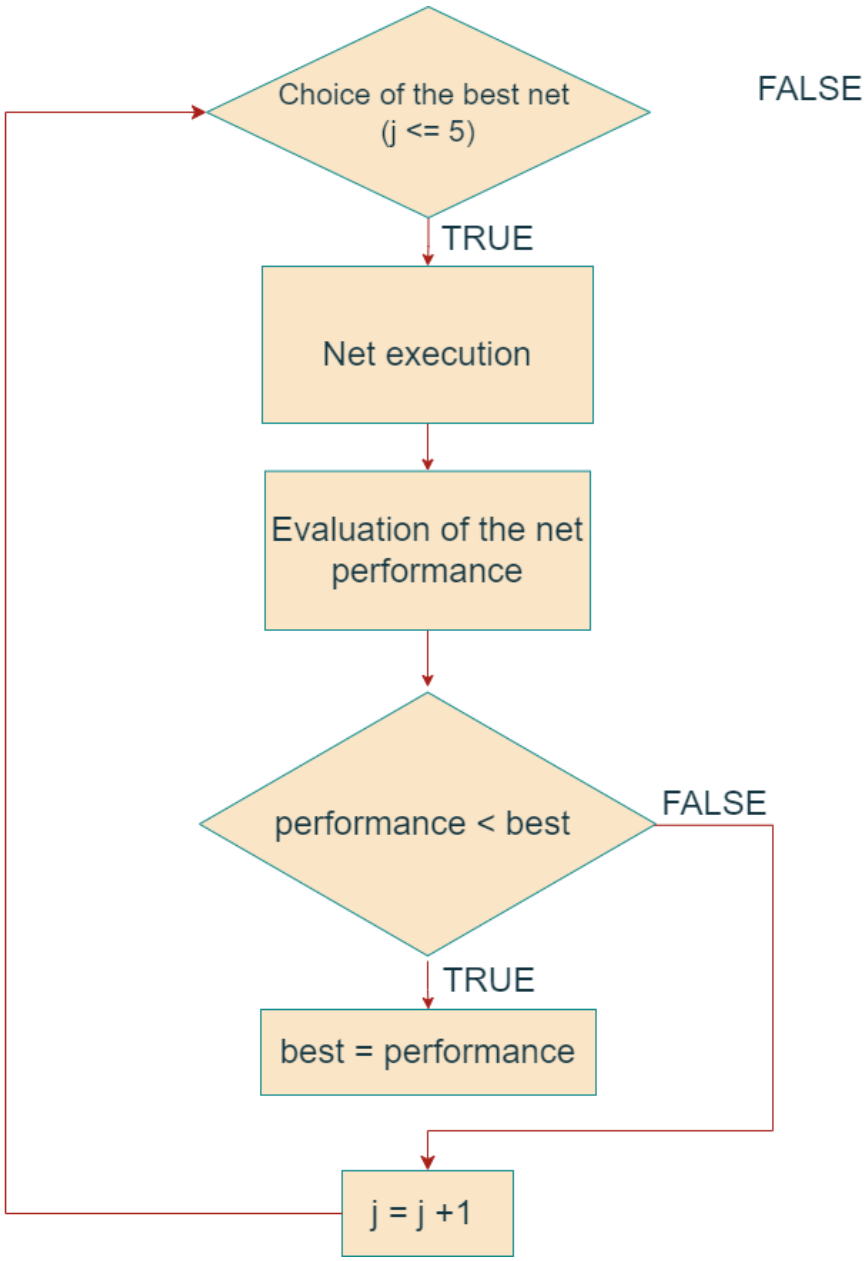


Figure 4.3: Flow chart of best net choice iterations

The algorithm's presentation through flow diagrams allows us to analyze each step of the process, starting with the data set management.

4.1 CHF Database

The NRC CHF database contains 24.579 CHF measurements in vertical water-cooled uniformly heated tubes compiled from various sources. The available data consist of measured boundary conditions (pressure P, mass flux G, inlet temperature T_{in}, and Critical Heat Flux CHF), geometrical parameters (test section diameter D, heated length L) and calculated parameters derived from measurements and water properties (outlet equilibrium quality X and inlet enthalpy ΔH_{in}. Due to the complexity of the database, the selection of the most influential parameters (Table 4.1 for range of values) is described in the following sections.

Variable	CHF [KW/m2̂]	P[Kpa]	G [Kg/m2̂/s]	X	D [mm]	L [m]
Min Value	50	100	8.2	-0.497	2	0.05
Max Value	16339.3	20000	7964	0.999	16	20

Table 4.1: Parameter spans of the NRC CHF database [7]

Figure 4.4 provides a plot matrix showing the relationship between any pair of inputs and input with output.

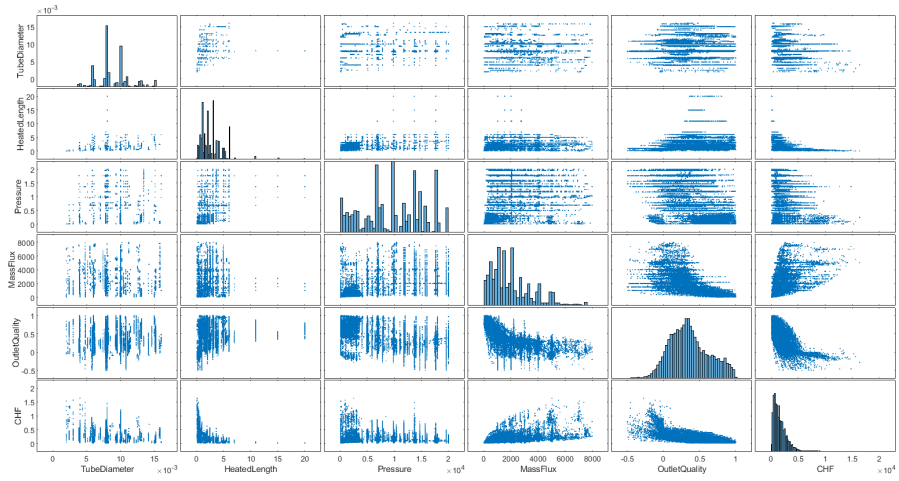


Figure 4.4: Scatter plot matrix of the NRC CHF database showing the relationship between pairs of variables [7].

4.1.1 Feature extraction and feature selection

Before developing the neural network model, it is essential to identify the parameters that are most relevant to the target - in this case CHF. Given the large data set of around 25,000 data points, **feature extraction** is particularly valuable in simplifying and structuring the data. This process allows us to capture **the most important characteristics** of the original data, allowing more efficient processing and analysis while preserving the key elements that influence CHF.

Pearson and Spearman correlation

Correlation coefficients are essential tools for identifying the most influential variables within our data set. To effectively assess the relationships between variables, it's important to understand the differences between Pearson and Spearman correlation coefficients. The Pearson correlation coefficient, also known as the linear correlation coefficient, quantifies the strength and direction of a linear relationship between two continuous variables. This coefficient, denoted r, ranges from -1 to 1, where -1 indicates a perfect negative correlation and 1 indicates a perfect positive correlation. A value of 0 indicates that there is no linear relationship between the variables. Given that our data set is normally distributed, using Pearson's correlation allows us to determine which variables have the most significant influence on the performance, specifically the Critical Heat Flux (CHF). The coefficient is given by the following formula [14]:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{4.1}$$

We apply this to our database with MATLAB and obtain the following values:

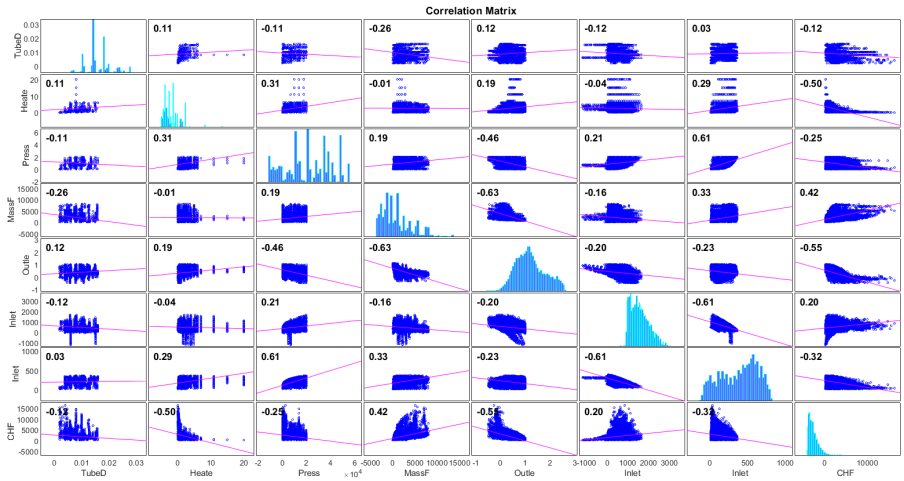


Figure 4.5: Pearson Correlation Matrix

For a better vision of the results here there is the table:

	TubeDiameter	HeatedLength	Pressure	MassFlux	OutletQuality	InletSubcooling	InletTemperature	CHF
TubeDiameter	1.00E+00	1.14E-01	-1.06E-01	-2.56E-01	1.16E-01	-1.19E-01	2.57E-02	2.57E-02
HeatedLength	1.14E-01	1.00E+00	3.09E-01	-7.58E-03	1.92E-01	-4.31E-02	2.92E-01	2.92E-01
Pressure	-1.06E-01	3.09E-01	1.00E+00	1.92E-01	-4.55E-01	2.10E-01	6.05E-01	6.05E-01
MassFlux	-2.56E-01	-7.58E-03	1.92E-01	1.00E+00	-6.26E-01	-1.55E-01	3.26E-01	3.26E-01
OutletQuality	1.16E-01	1.92E-01	-4.55E-01	-6.26E-01	1.00E+00	-2.00E-01	-2.30E-01	-2.30E-01
InletSubcooling	-1.19E-01	-4.31E-02	2.10E-01	-1.55E-01	-2.00E-01	1.00E+00	-6.05E-01	-6.05E-01
InletTemperature	2.57E-02	2.92E-01	6.05E-01	3.26E-01	-2.30E-01	-6.05E-01	1.00E+00	1.00E+00
CHF	-1.23E-01	-5.03E-01	-2.53E-01	4.24E-01	-5.55E-01	2.02E-01	-3.18E-01	-3.18E-01

Table 4.2: Pearson's Linear Correlation Coefficient for various parameters

As for the **Spearman correlation**, it measures the a monotonic relationship between two variables based on the **rank** of the data. Spearman correlation is often used for data that consists of outliers. To measure the Spearman correlation, the indicator used is the Spearman coefficient rs, also known as the rank coefficient, given by the formula below [14]:

$$r_s = 1 - \frac{6\sum d^2}{n(n^2-1)} \tag{4.2}$$

Also for this we do the analysis with MATLAB, providing the matrix and values for each input.

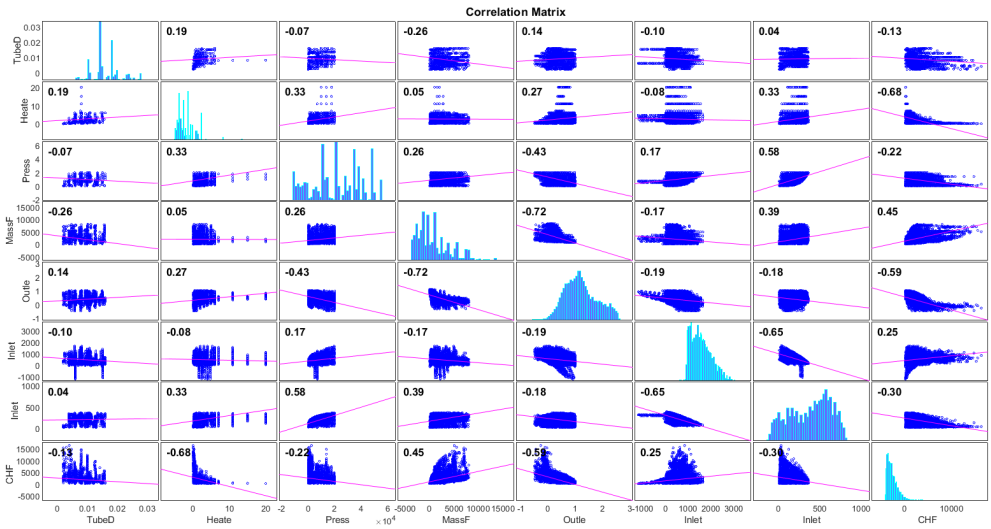


Figure 4.6: Spearman Correlation Matrix

	TubeDiameter	HeatedLength	Pressure	MassFlux	OutletQuality	InletSubcooling	InletTemperature	CHF
TubeDiameter	1.00E+00	1.90E-01	-6.59E-02	-2.64E-01	1.41E-01	-1.02E-01	4.44E-02	4.44E-02
HeatedLength	1.90E-01	1.00E+00	3.34E-01	5.44E-02	2.72E-01	-7.63E-02	3.25E-01	3.25E-01
Pressure	-6.59E-02	3.34E-01	1.00E+00	2.60E-01	-4.28E-01	1.73E-01	5.81E-01	5.81E-01
MassFlux	-2.64E-01	5.44E-02	2.60E-01	1.00E+00	-7.18E-01	-1.70E-01	3.95E-01	3.95E-01
OutletQuality	1.41E-01	2.72E-01	-4.28E-01	-7.18E-01	1.00E+00	-1.85E-01	-1.84E-01	-1.84E-01
InletSubcooling	-1.02E-01	-7.63E-02	1.73E-01	-1.70E-01	-1.85E-01	1.00E+00	-6.49E-01	-6.49E-01
InletTemperature	4.44E-02	3.25E-01	5.81E-01	3.95E-01	-1.84E-01	-6.49E-01	1.00E+00	1.00E+00
CHF	-1.28E-01	-6.81E-01	-2.24E-01	4.48E-01	-5.93E-01	2.49E-01	-2.96E-01	-2.96E-01

Table 4.3: Spearman Correlation Coefficients

4.1.2 Analysis of Linear Relationships and R^2 Coefficient Between Input Variables and CHF

We know that complex databases with multiple inputs are highly uncertain and it is important to use a rigorous method to obtain a realistic output distribution. The next step is to understand how each input affects the output. This is done by analyzing the linear relationship and evaluating the R^2 coefficient. In MATLAB we perform linear regression:

1. for each pair of inputs (input 1 vs input 2, input 1 vs input 3 etc...);
2. between each input and the output;
3. between chf and the whole group of inputs;
4. between chf and the most influential inputs.

For each model the code provides the model formula, the estimated coefficients (with their properties) and the model summary statistics. Coefficient properties include:

- *Estimate*: Coefficient estimate for each corresponding term in the model;
- *SE*: Standard Error of the coefficients;
- *tStat*: t-statistic for each coefficient to test the null hypothesis that the corresponding coefficient is zero against the alternative that it is different from zero, given the other predictors in the model. Note that $tStat = \frac{Estimate}{SE}$;
- *pValue*: p-value for the t-statistic of the hypothesis test that the corresponding coefficient is equal to zero or not

Instead, the summary statistics are:

- *Number of observations*: Number of rows without any NaN values;
- *Error degrees of freedom*: $n - p$, where n is the number of observations, and p is the number of coefficients in the model, including the intercept;
- *Root mean squared error*: estimates the standard deviation of the error distribution;
- *R-squared and Adjusted R-squared*: Coefficient of determination and adjusted coefficient of determination, respectively

Let's focus on part 3 and 4. We obtain the following results:

	Estimate	SE	tStat	pValue
(Intercept)	4.13E+03	5.68E+01	7.28E+01	0.00E+00
x1	-1.25E+04	2.45E+03	-5.10E+00	3.45E-07
x2	-1.60E+02	3.30E+00	-4.84E+01	0.00E+00
x3	-1.66E-01	3.14E-03	-5.29E+01	0.00E+00
x4	1.76E-01	5.66E-03	3.11E+01	5.12E-208
x5	-3.58E+03	3.50E+01	-1.02E+02	0.00E+00
x6	1.19E+00	5.27E-02	2.26E+01	6.36E-112
x7	8.73E-01	2.41E-01	3.62E+00	2.97E-04

Table 4.4: Linear regression model between the whole group of inputs and the CHF

Number of observations:	24579
Error degrees of freedom:	24571
Root Mean Squared Error:	874
R-squared:	0.699
Adjusted R-Squared:	0.698
F-statistic vs. constant model:	8.13e+03, p-value = 0

Table 4.5: Statistical Summary

	Estimate	SE	tSat	pValue
(Intercept)	5.38E+03	3.88E+01	1.39E+02	0.00E+00
x1	-3.79E+04	2.52E+03	-1.51E+01	5.83E-51
x2	-1.50E+02	3.46E+00	-4.32E+01	0.00E+00
x3	-1.53E-01	1.38E-03	-1.10E+02	0.00E+00
x4	6.44E-02	5.42E-03	1.19E+01	1.53E-32
x5	-4.19E+03	3.49E+01	-1.20E+02	0.00E+00

Table 4.6: Linear regression model between CHF and the most influencing inputs

Number of observations:	24579
Error degrees of freedom:	24573
Root Mean Squared Error:	923
R-squared:	0.664
Adjusted R-Squared:	0.664
F-statistic vs. constant model:	9.71e+03, p-value = 0

Table 4.7: Statistical Summary

The same analysis has been held with the rank of the whole matrix, thanks to **Spearman correlation** used previously. The results are the following;

	Estimate	SE	tStat	pValue
(Intercept)	6.21E+03	1.73E+02	3.59E+01	1.31E-275
x1	1.70E-01	2.82E-03	6.04E+01	0.00E+00
x2	-6.33E-01	3.67E-03	-1.73E+02	0.00E+00
x3	-4.54E-01	7.28E-03	-6.24E+01	0.00E+00
x4	5.85E-01	5.27E-03	1.11E+02	0.00E+00
x5	-5.59E-02	5.80E-03	-9.63E+00	6.29E-22
x6	5.81E-01	6.90E-03	8.43E+01	0.00E+00
x7	3.01E-01	8.87E-03	3.40E+01	1.28E-247

Table 4.8: Linear regression model (considering Rank of matrix) between the whole group of inputs and the CHF

4.2 Training, Validation and Test

Once the database has been created, the first step in training a neural network (NN) is to divide the entire data set into **training, validation and test sets**. They are independent of each other and we make this classification randomly.

- Training Set: this is the data set we use in our model to learn potential patterns and relationships. It is used to adjust the parameters. It is important to keep the division as unbiased as possible.
- Validation Set: it is used to evaluate and fine-tune a machine learning model during training, helping to assess the model's performance and make adjustments. It is also essential for
- Test Set: it is the set of data used to evaluate the final performance of a trained model.

There are several ways to split the data set, depending on several factors, including the use case, the amount of data, the quality of the data and the number of hyperparameters. **Random Sampling** is the most common approach to partitioning a data set and it is the method used in our model. This method involves a randomly assigning samples to training, validation and test according to pre-determined ratios. For class-balanced data sets, random sampling ensures that the partitioning is unbiased. In our case study, we set up two different models: one without test sets, with a split ratio of $[80, 20]$, and one with test, with a split ratio $[70/15/15]$, ensuring a balanced and representative sample for each subset.

4.2.1 Normalization of data

After dividing the data set, scaling the inputs and targets before training is often beneficial to keep them within a certain range. In MATLAB, the *mapminmax* function is used to scale inputs and targets to the range $[-1, 1]$, following the equation below [13].

$$x_{normalized} = 2\left(\frac{x - x_{min}}{x_{max} - x_{min}}\right) - 1. \tag{4.3}$$

Normalization is applied to both the training and validation sets before separation, and to the test set if defined.

4.3 Problem of overfitting

During the training phase, accuracy problems are common. They are typically related to the poor performance of the model, depending on its "simplicity" or "complexity" [8].

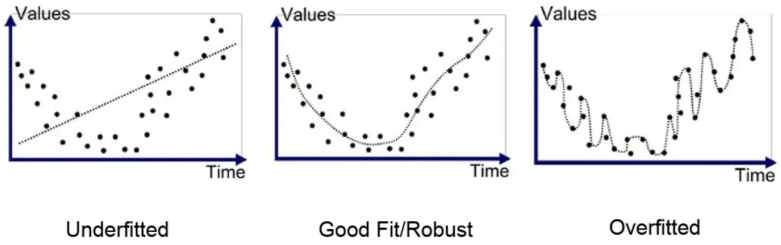


Figure 4.7: Graphs showing the difference between underfitting (on the left), good fit and overfitting (on the right) [8]

In an underfitting situation the line does not cover all the points on the graph. Instead, in the opposite situation, overfitting, the model is too complex and the line covers points that are outliers or noise. The best situation is the central graph. An ideal way to keep it is to set the number of the total neurons of the net (including input and output units) less than the number of training and validation data, useful for the training of the net.

$$n_i \cdot n_{h1} + n_{h1} + n_{h1} \cdot n_{h2} + n_{h2} + ... + n_{ho} \cdot n_{ho} + n_o < N_{train} + N_{validation}(+N_{test}) \quad (4.4)$$

4.4 Train and execution of the Neural Network

This is the "core" of our model. One of the ways to create a feed-forward network in MATLAB is to use the "feedforwardnet" function, specifying the number of hidden layers and the type of training.

```
net = feedforwardnet(n_hidden, 'trainscg');
```

Trainscg is a network training function that updates the weights and biases using the scaled conjugate gradient method. For the training we need to set several parameters:

- net.trainParam.min_grad: Maximum performance gradient;
- net.trainParam.max_Fails: Maximum number of validation failures;

- net.trainParam.show: epoch visualization;
- net.trainParam.epoch: Maximum number of iteration.

We also set the subdivision index and the transfer function between input, layers and output ("tansig" between input and the first layer hidden, "tansig" between adjacent layers hidden and "purelin" between the last layer and output).

```
net.layers{1}.transferFcn = 'tansig';
for i = 2:length(n_hidden)
    net.layers{i}.transferFcn = 'tansig';
end
net.layers{end}.transferFcn = 'purelin';
```

Then the execution starts and the following dashboard appears:

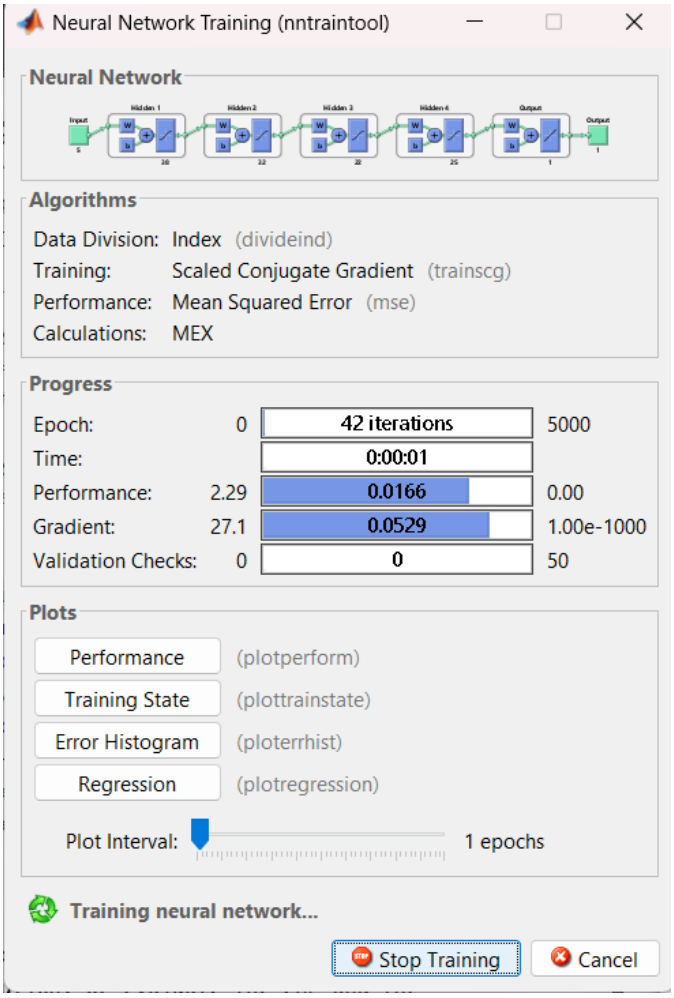


Figure 4.8: Neural network training dashboard

At the top is the layout of the neural network, showing the input layer, hidden layers, and output layer. Below this are the key algorithms used by the network, including the data set partitioning, the training function, and the error metric that determines performance and ultimately guides the selection of the optimal network.

The progress section reflects the parameters set by the user and tracks their evolution through the iterations, highlighting key indicators such as performance, gradient, and validation checks.

At the bottom, these aspects are visualized through various plots, providing a graphical representation of the network's training process.

4.4.1 Performance

The Plotperform function tracks the error against the time for training, validation and test performance of the TR training set returned by the train function.

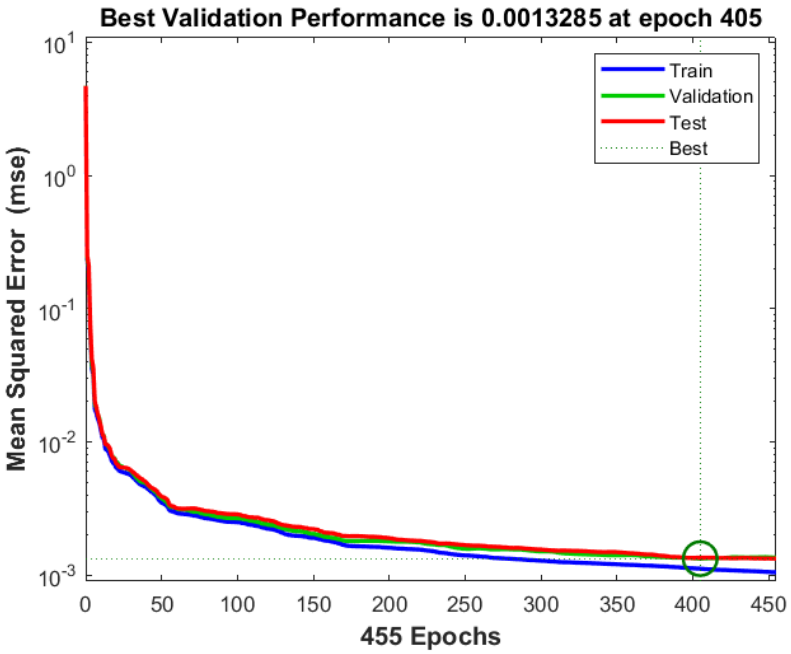


Figure 4.9: Plot of the performance of training of the neural network

Typically, the error decreases after a larger number of training periods, but it can start to increase on the validation data set as the network begins to overfit the training data. To improve accuracy and minimize bias errors, the best network is selected from five iterations based on the best performance on the validation set.

4.4.2 plottrainstate

This plot represents the training state of a neural network. There are two subsections: **gradient** is a value of the backpropagation gradient at each iteration in logarithmic scale. The other is the **validation checks**: there are iterations where the MSE has increased its value.

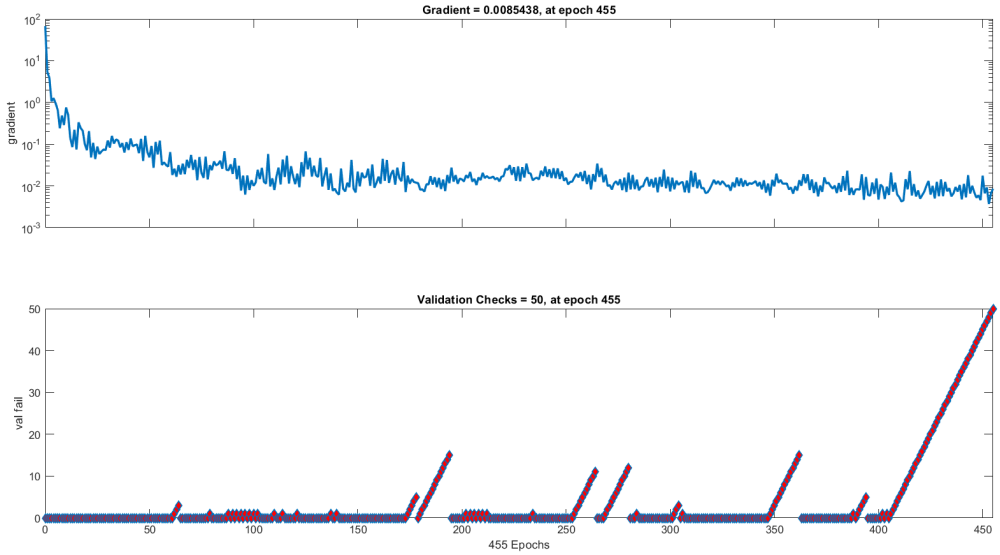


Figure 4.10: Evolution of gradient (top figure) and validation checks (bottom figure)

When the validation check reaches the threshold set when the net was created, the net stops training and execution and moves on to the next one.

4.4.3 Error Histogram

The error histogram is the histogram of the errors between target values and predicted values after training a feed-forward neural network.Since these error values indicate how the predicted values differ from the target values, they can be negative. The bins are the number of vertical bars on the graph. The entire error range is divided into 20 smaller bins here. The Y-axis represents the number of samples from data set which are in a particular bin. The zero error line corresponds to the zero error value on the error axis (i.e. X-axis).

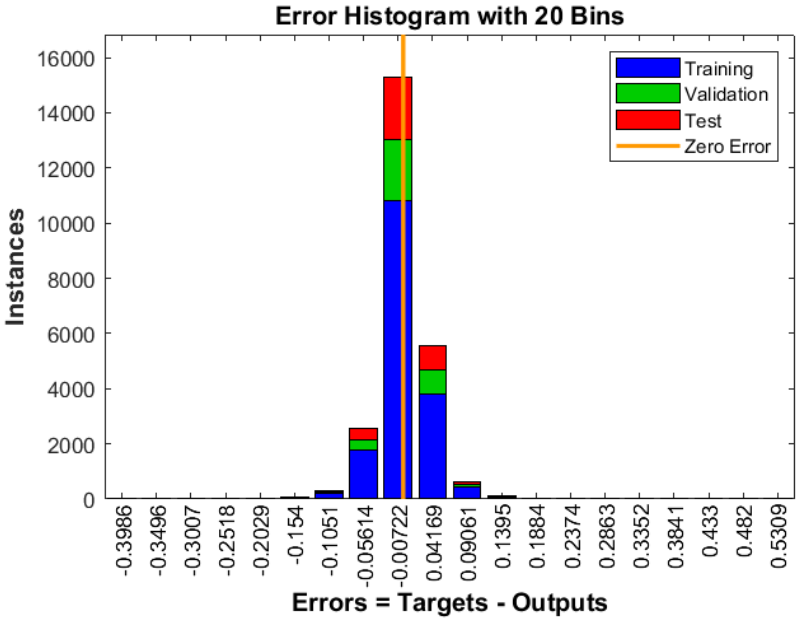


Figure 4.11: Plot of the error histogram

4.4.4 Plot regression

This plots the actual targets against the predicted outputs. The x-axis represents the targets (actual values), while y-axis represents the output values (predicted values). The diagonal is the line of best fit and the "coloured" line represents the line of regression with respect to the data points. The variable "R" is the correlation coefficient between the targets and the outputs of the neural network.

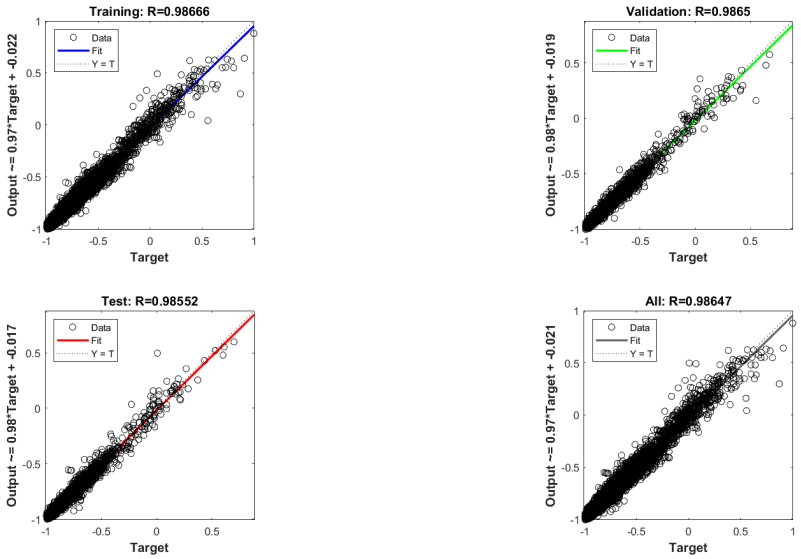


Figure 4.12: Plot of network regression

4.5 Post-processing

4.5.1 CHF LUTs Results

After execution, we evaluate the CHF and the corresponding metrics. First of all we put the result of the LUTs (table 4.9), since we are starting from it, and then we check if the NN model is better than it in predicting the CHF.

ALL	**Size of the data set**		**24579 samples**
	Evaluation	Mean P/M	1.032
		Std P/M	0.362
		RMSPE [%]	19.8
		MAPE [%]	36.3
		Q^2 error	0.063

Table 4.9: CHF LUTs prediction performance [7]

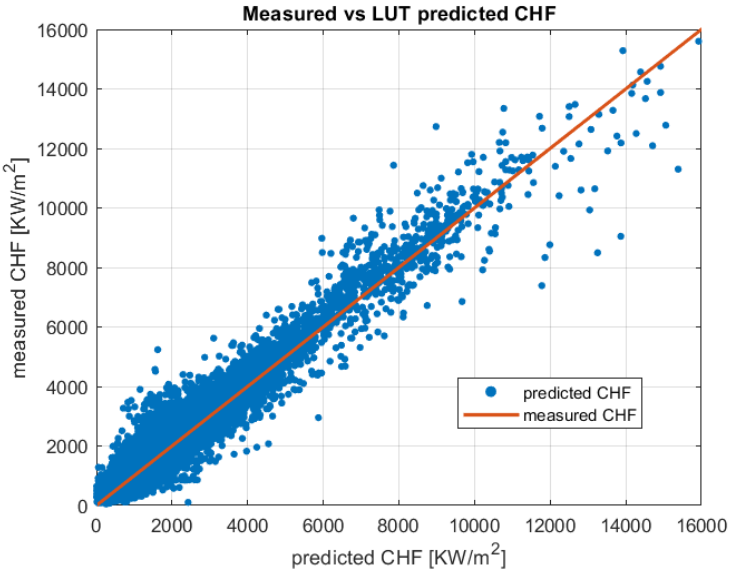
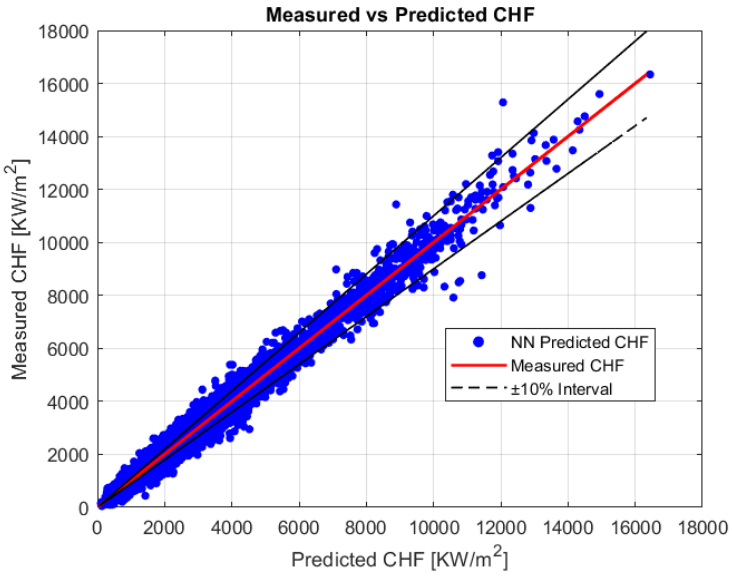
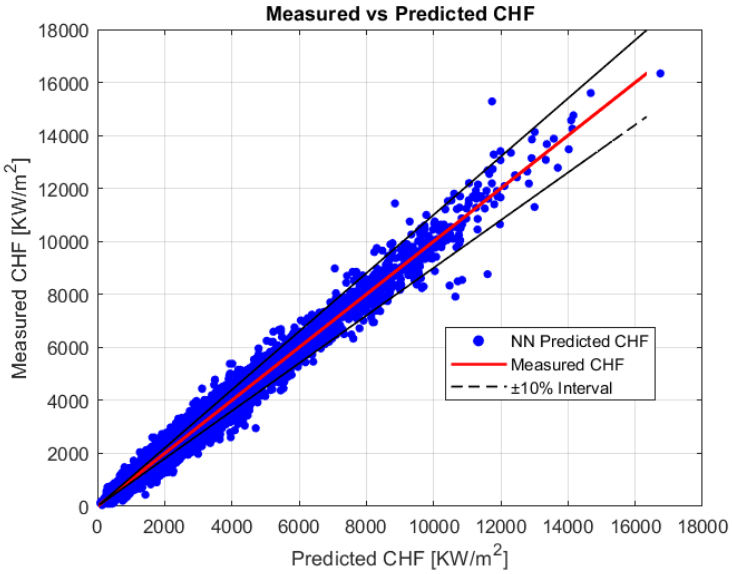


Figure 4.13: Measured vs LUTs predicted CHF [7]

4.5.2 NN results



(a) NN ensemble with training and validation sets



(b) NN Ensemble with entire set

In addition, we analyse the metrics for each individual network over all iterations, as well as the overall performance of the ensemble model [7]. It is therefore important to clearly define the statistical terms used before comparing NN performance with respect to the LUTs..

- The **mean** μ is defined as the expected value of a random variable Y, which is assumed to be continuous with probability density function (pdf) $p_y(y)$:

$$\mu = E[Y] = \int_{-\infty}^{+\infty} y p_y(y) dy \tag{4.5}$$

  If a pdf is unknown and the only thing available for Y is a set of N samples/meauserements y_i with $i = 1, ..., N$, then its mean can be estimated through:

$$\hat{\mu} = \frac{\sum_{i=1}^{N} y_i}{N} \tag{4.6}$$

- The **standard deviation** (σ) of a random variable is a metric of dispersion around its expected value and for the continous random variable Y defined as:

$$\sigma = \sqrt{E[(Y - \mu)^2]} = \sqrt{\int_{-\infty}^{+\infty} (y - \mu^2) p_y(y) dy} \tag{4.7}$$

  As for the mean, if all that is available for Y is a set of N samples/measurements y_i with $i = 1, ..., N$, then the standard deviation can be estimated through:

Other metrics are considered to measure the discrepancy between predictions and measurements.

- The **root mean square error (RMSE)** is the quadratic mean of the differences between the observed and predicted values. These deviations are called residuals when the calculations are performed on the sample of data used for estimation (and are therefore always relative to an estimate) and errors (or prediction errors) when they are calculated out of sample (aka on the full set, relative to a true value rather than an estimate).[7] It can be defined in three ways:

$$RMSE = \sqrt{E[(Y - Y^m)^2]} \tag{4.8}$$

$$R\hat{M}SE = \sqrt{\frac{\sum_{i=1}^{N} (y_i - y_i^m)^2}{N}} \tag{4.9}$$

$$R\hat{M}SE_p = 100\sqrt{\frac{\sum_{i=1}^{N} (\frac{y_i - y_i^m}{y_i^m})^2}{N}} \tag{4.10}$$

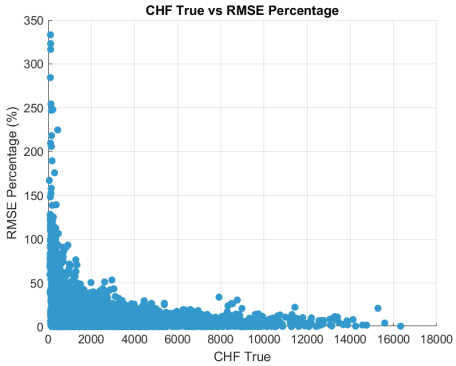
- The **mean absolute error (MAE)** is a measure of the error between paired observations expressing the same phenomenon. The MAE gives equal weight to all discrepancies, whereas the RMSE gives more weight to larger discrepancies [7].

$$MAE = E[|Y - Y^m|] \tag{4.11}$$

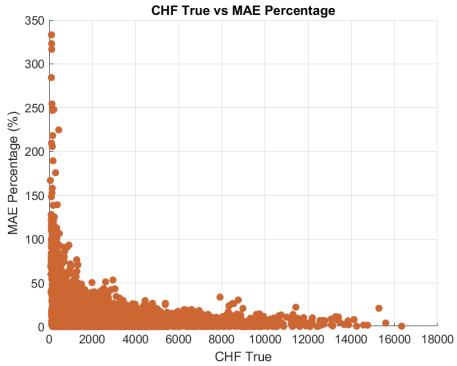
$$\hat{MAE} = \frac{\sum_{i=1}^{N} |y_i - Y_i^m|}{N} \tag{4.12}$$

$$\hat{MAE}_p = 100\frac{\sum_{i=1}^{N} |\frac{y_i - y_i^m}{y_i}|}{N} \tag{4.13}$$

  In this context, the RMSE and the MAE coincide. In the next figures we have the distribution of errors between the average ensemble prediction (that is the prediction we consider to be the final output of our model) and the one experimentally collected in the initial database. We are interested in the CHF, the output of the model. For the sake of completeness, figures showing the distribution of the errors as a function of all possible input variables are given in appendix A.

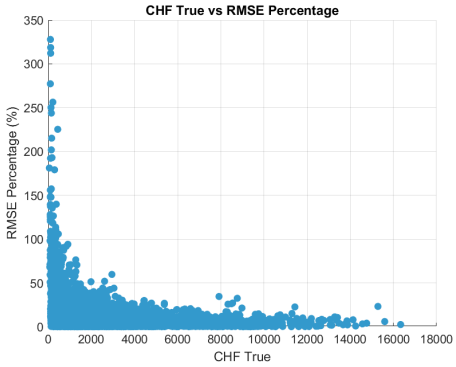


(a) RMSE [%]

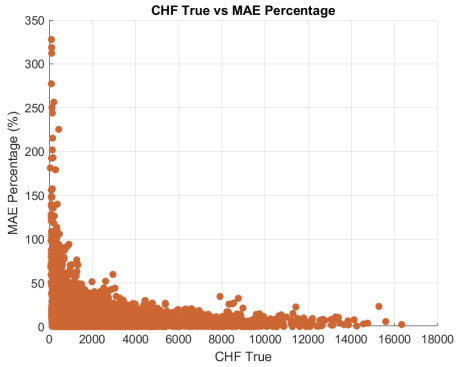


(b) MAE [%]

Figure 4.15: RMSE and MAE percentage error distribution for neural network with training and validation set



(a) RMSE [%]



(b) MAE [%]

Figure 4.16: RMSE and MAE percentage error distribution for neural network with rain, Validation and Test set

- The Q^2 **error (EQ^2)** measures discrepancies without taking into account the variance of the random variable. This means that small values of RMSE and MAE can be misleading if the random variable has small variations. The EQ^2 weights the square error discrepancies by the variance and essentially measures how much of the variability of the data are actually captured by the ML model. There may be cases where the RMSE is small, and thus the numerator of EQ^2 is small, but the variance in the denominator is also small, and thus the EQ^2 metric ends up with a large value. A value of 0 indicates that the ML model has perfect predictive capabilities, while a value greater than 1 indicates that the ML model is worse than a model that always uses the mean value as its predictor [7].

$$EQ^2 = \frac{E[(Y - Y^m)^2]}{E[(Y - \mu)^2]} \tag{4.14}$$

$$EQ^2 = \frac{\sum_{i=1}^{N}(y_i - y_i^m)^2}{\sum_{i=1}^{N}(y_i - \hat{\mu})^2} \tag{4.15}$$

- Another way to measure the performance of regression algorithm is the **normalized root mean square error (NRMSE)**: which is calculated in:

$$NRMSE = \frac{\sqrt{\sum_{i=0}^{N} \frac{(Y_i - \hat{Y}_i)^2}{N}}}{Y_{mean}} \tag{4.16}$$

  The NRMSE is a measurement of the absolute error. This means that the errors at high CHF will count more towards the NRMSE than the same proportional error at lower CHF.

- Last but not least, the $\frac{P}{M}$ **ratio** relates P, the predicted or estimated value, to M, the observed or actual value. In machine learning, this ratio is often used to assess how well a model's predictions match the observed data. This statistical metric allows the performance of the neural network to be compared with that of the LUTs.

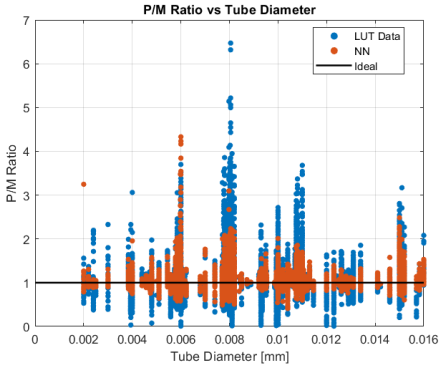
The two neural network models were simulated, the first with the data set split into training and validation data and the second with training, validation and test data. Once both simulations have been run, we record the performance metrics and collect them in tables (4.10 and 4.11). If we compare them with the 4.9 table, according to what has been described previously for each metric, we see that all of them are **better** than those of the LUTs model. Another comparison is made by analysing the respective $\frac{P}{M}$ ratio distribution among the input variables, as shown in the set of figures of 4.17.

ALL	**Size of the data set**		**24579 samples**
	Evaluation	Mean P/M	1.022
		Std P/M	0.161
		RMSPE [%]	16.219
		MAPE [%]	9.843
		NRMSE	0.108
		Q^2 error	0.015

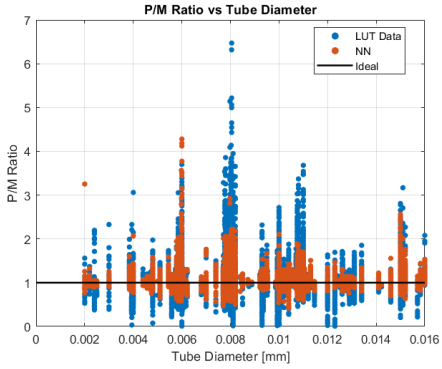
Table 4.10: NN with only training and Validation prediction performances

TEST	**Size of the data set**		**3687 samples**
	Evaluation	Mean P/M	1.024
		Std P/M	0.194
		RMSPE [%]	19.567
		MAPE [%]	11.085
		NRMSE	0.122
		Q^2 error	0.019
ALL	**Size of the data set**		**24579 samples**
	Evaluation	Mean P/M	1.023
		Std P/M	0.163
		RMSPE [%]	16.430
		MAPE [%]	10.003
		NRMSE	0.111
		Q^2 error	0.016

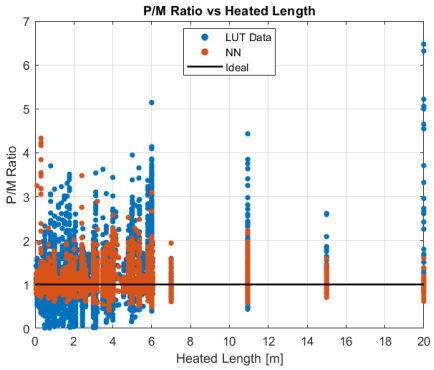
Table 4.11: NN with training, validation and Test prediction performances



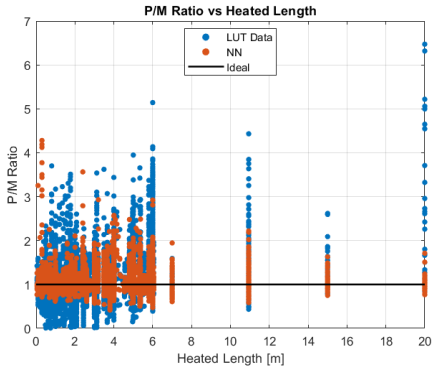
(a) P/M ratio vs Tube Diameter



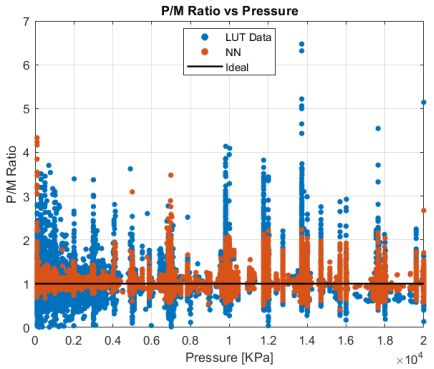
(b) P/M ratio vs Tube Diameter



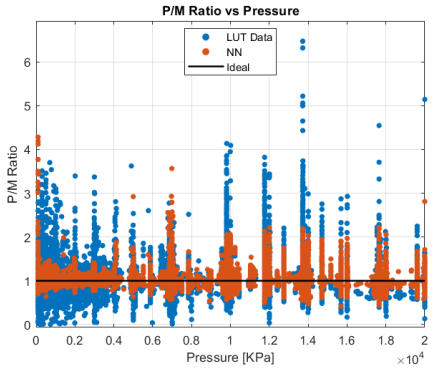
(c) P/M ratio vs Heated Length



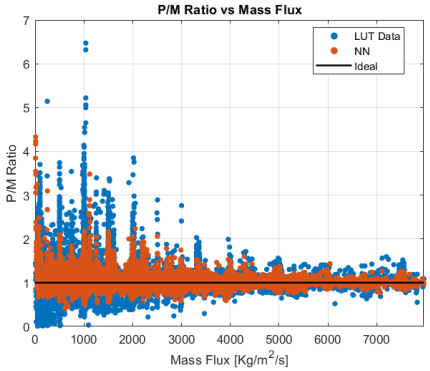
(d) P/M ratio vs Heated Length



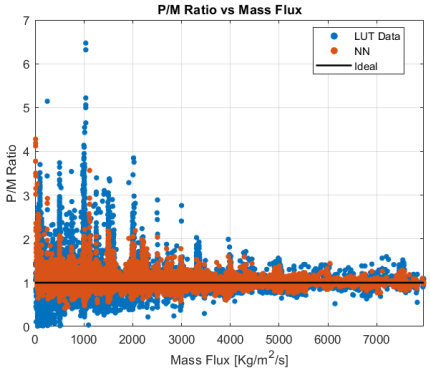
(e) P/M vs Pressure



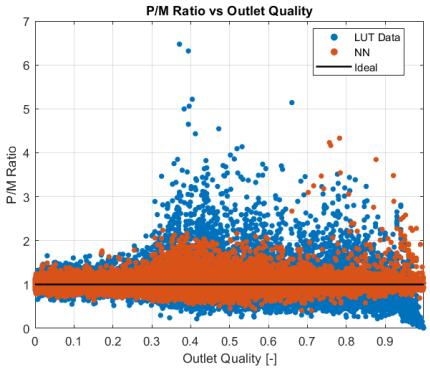
(f) P/M vs Pressure



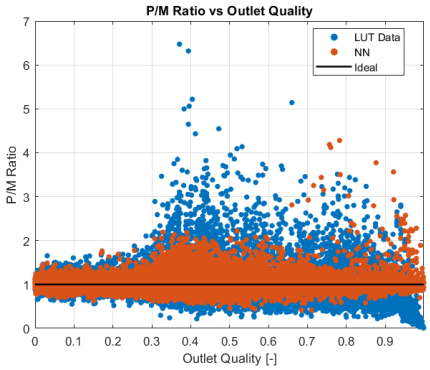
(g) P/M ratio vs Mass Flux



(h) P/M ratio vs Mass Flux



(i) P/M ratio vs Outlet Quality



(j) P/M ratio vs Outlet Quality

Figure 4.17: LUTs (blue) and NN (orange) Predicted versus Measured CHF scatter plots, versus selected independent parameters, on the left we have the NN model with only training and validation sets, on the right we have the "complete" model

4.5.3 Slice division of data set

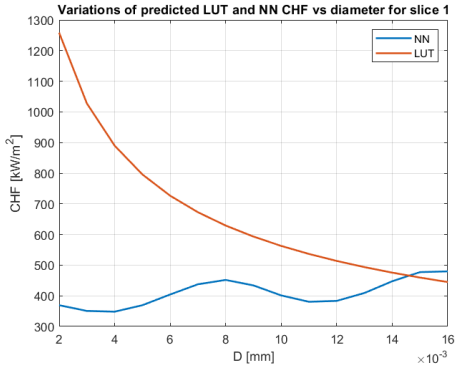
Another approach to data analysis is to divide the whole data set into smaller subsets. This allows for a more detailed extraction of information and a better understanding of how CHF varies when only one input parameter is changed at a time. As described in the NRC benchmark document [7], the data set is segmented by varying parameters such as D, L, G, P, and χ while holding all other variables reasonably constant.

From the identified data slices, two subsets per varying parameter have been selected and summarized in table below.

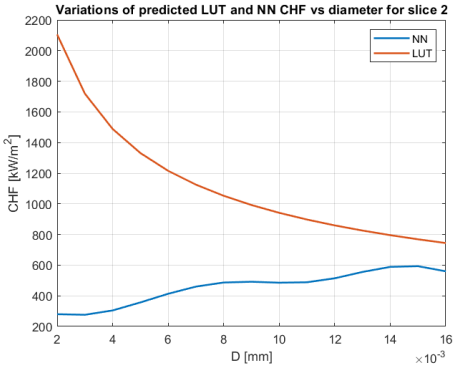
Table 4.12: Slice data sets [7]

# Slice	D [mm]	L [m]	P [KPa]	G $[Kg/(m^2s)]$	X [-]
1	0 - 16	6.00	14701	998.5	0.391
2	0 - 16	6.00	9807	1003.3	0.529
3	8.01	0 - 20	9806	1000	0.587
4	8.11	0 - 20	2009	752.2	0.756
5	8.00	0.998	0 - 20000	2006	0.14
6	13.40	3.658	0 - 20000	20040.2	0.378
7	8.00	1.57	12750	0 - 80000	0.144
8	10.00	4.966	16000	0 - 80000	0.343
9	8.14	1.943	9831	1519.5	-0.5 - 1.0
10	8.00	0.997	17650	2002.7	-0.5 - 1.0

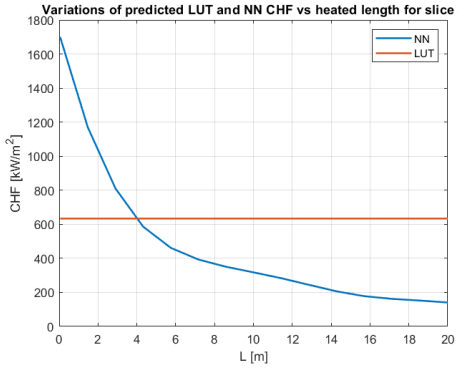
In the provided databases, the varying parameters are sliced into 15 equidistant values across the defined ranges. These sliced data sets allow the physical behaviour of predictive CHF models to be analysed across the parameter space, ensuring robustness and mitigating overfitting. This approach enhances the explanatory power of the models and increases confidence in their predictive ability [7]. The representation of each of these is shown in the figures below.



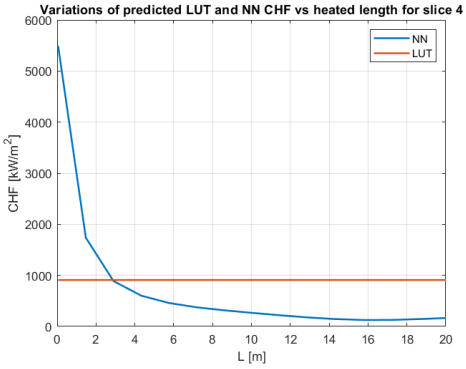
(a) The behavior of the model and LUTs as a function of diameter (slice 1).



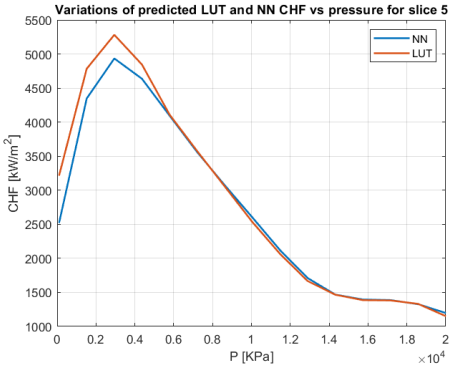
(b) The behavior of the model and LUTs as a function of diameter (slice 2)



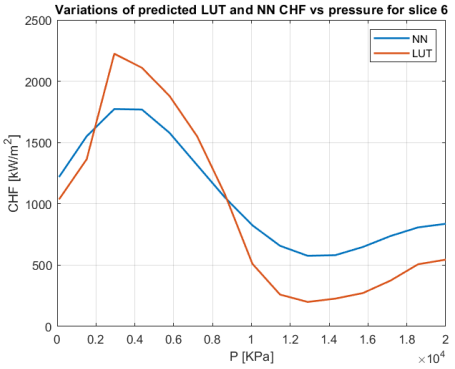
(c) The behavior of the model and LUTs as a function of heated length (slice 3)



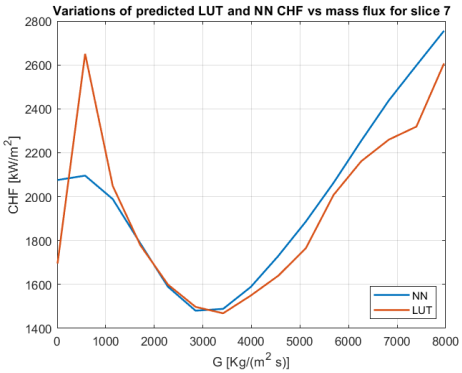
(d) The behavior of the model and LUTs as a function of heated length (slice 4)



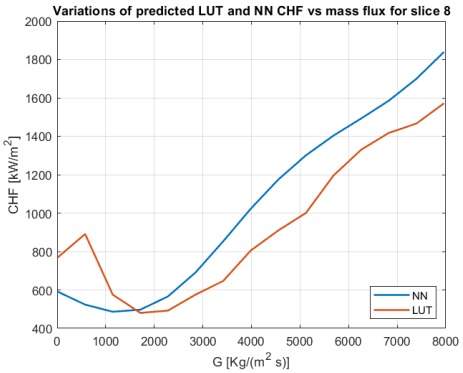
(e) The behavior of the model and LUTs as a function of pressure (slice 5)



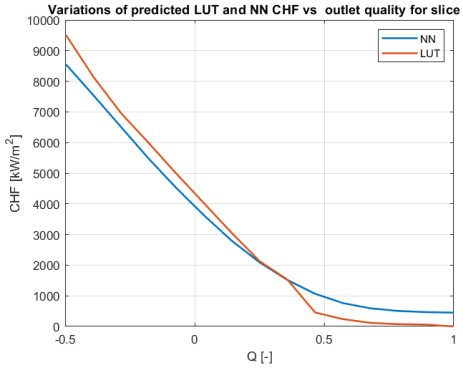
(f) The behavior of the model and LUTs as a function of pressure (slice 6)



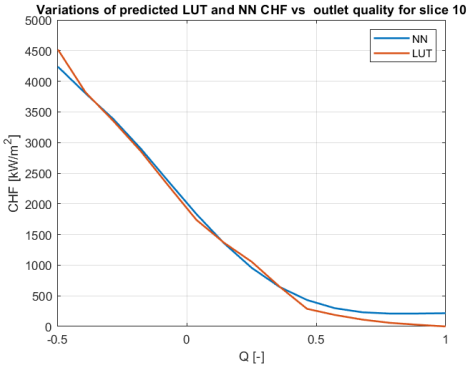
(g) The behavior of the model and LUTs as a function of mass flux (slice 7)



(h) The behavior of the model and LUTs as a function of mass flux (slice 8)



(i) The behavior of the model and LUTs as a function of outlet quality (slice 9)



(j) The behavior of the model and LUTs as a function of outlet quality (slice 10)

Figure 4.18: Comparison of the model and LUTs behavior as a function of different parameters.

Chapter 5

The split model: a way to improve the effectiveness of the neural network

This chapter looks at a method of improving the efficiency of the neural network in predicting CHF. One initial strategy is to use the same type of model as the original (an ensemble of networks), but to focus specifically on areas where accuracy decreases, allowing the network to specialize accordingly. One possible approach to achieve this is to analyze the points where the absolute value of P/M exceeds a certain threshold, i.e. where there is an overestimation and underestimation. Finally, two "specialized" models are developed: one for the network without test (i.e. only training and validation data) and one for the network with test, using the same partitioning of the "original" model.

5.1 Filtering and specialization of the Neural Network

To refine our results, we define different target thresholds to "filter" the data. This allows us to identify the regions where the network performs worse compared to the average accuracy. We highlight the points where accuracy is lower for different target values ($|\frac{P}{M}| > [1.1 - 1.5]$). This is the first step towards developing a specialized model. By analyzing the five different configurations and their respective identified points, the best compromise is found in the second configuration (as shown in 5.1, where the absolute ratio $\frac{P}{M}$ is set to be greater than 20% above the threshold. For completeness other configurations are shown in the appendix.

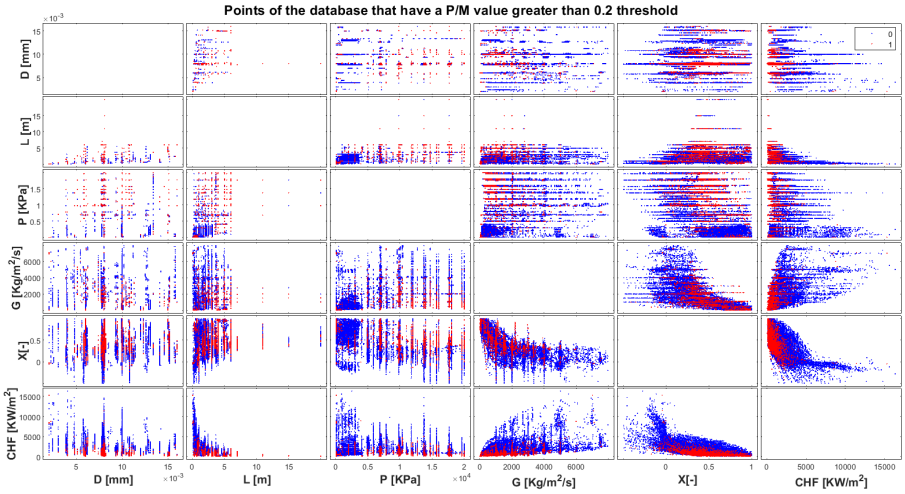


Figure 5.1: CHF database with points where the $\frac{P}{M}$ ratio is greater than the 20% of threshold value

This approach allow us to divide the initial data set into two blocks: one with the "filtered" data, where the model's performance deteriorates (in red), and the other with the remaining data (in blue).
Another filter has been carried out, finding values of mass flux $> 6000 \frac{Kg}{m^2 s}$ and the outlet quality > 0. We get about 2800 indices that represents the filtered data. The others are the remaining data.

The next objective is to develop two specialized models for the split data sets. For the first data set, an additional step is required: to avoid overfitting, the number of layers must be lower than the number of training and validation parameters, which requires a reduction in complexity. This issue does not arise for the second data set, where it is sufficient to maintain the original model structure.

5.1.1 The number of layer hidden

Choosing the number of layers, as described above, is a fundamental step, because if we set too few neutrons, it will result in **underfitting**. So there are these "rules of thumb":

- The number of hidden neurons should be between the size of the input layer and the size of the output layer;
- The number of hidden neurons should be $\frac{2}{3}$ the size of the input layer, plus the size of the output layer;

- The number of hidden neurons should not be less than twice the size of the input layer.

The second step is to configure the hidden layers and analyze their impact on the final results. For this purpose, a comparison has been made between three different configurations: one with five intermediate layers, one with four, and one with three. This approach allows us to determine the most effective structure.

Starting from the first configuration, we evaluate nine different parameter configurations, reducing the total number proportionally each time to identify the optimal one. The three main evaluation metrics used in this selection process are RMSE, MAE, and the R^2 coefficient, as shown in the results below.

```
23/10/24 10.50  MATLAB Command Window                1 of 1

Configurazione 1: RMSE Percentuale = 26.05%, MAE Percentuale = 26.05%,↙
R² = 0.9102
Configurazione 2: RMSE Percentuale = 24.96%, MAE Percentuale = 24.96%,↙
R² = 0.9167
Configurazione 3: RMSE Percentuale = 24.59%, MAE Percentuale = 24.59%,↙
R² = 0.9183
Configurazione 4: RMSE Percentuale = 24.42%, MAE Percentuale = 24.42%,↙
R² = 0.9178
Configurazione 5: RMSE Percentuale = 24.55%, MAE Percentuale = 24.55%,↙
R² = 0.9171
Configurazione 6: RMSE Percentuale = 24.22%, MAE Percentuale = 24.22%,↙
R² = 0.9199
Configurazione 7: RMSE Percentuale = 23.92%, MAE Percentuale = 23.92%,↙
R² = 0.9205
Configurazione 8: RMSE Percentuale = 24.02%, MAE Percentuale = 24.02%,↙
R² = 0.9204
Configurazione 9: RMSE Percentuale = 23.71%, MAE Percentuale = 23.71%,↙
R² = 0.9220
La configurazione con il RMSE percentuale più basso è la configurazione↙
9 con RMSE = 23.71%
La configurazione con il MAE percentuale più basso è la configurazione 9↙
con MAE Percentuale = 23.71%
La configurazione con il R² più alto è la configurazione 9 con R² =↙
0.9220
>>
```

Figure 5.2: Comparison of the different configurations based on the total number of neurons

Comparing this model with the original one, we see that the data set is about ten times smaller than the original. Consequently, a higher error would be expected. However, the RMSE remains within an acceptable range, indicating that the network's performance is still satisfactory. Moreover, the simplified model retains a good degree of generalization. According to figure 5.2 the best configuration is the last one with this layout: $[38, 32, 17, 25, 17]$. From it, we decrease the number of layer hidden and we do the same.

```
MATLAB Command Window                                    Page 1

Configurazione 1: RMSE Percentuale = 23.83%, MAE Percentuale = 23.83%,↙
R² = 0.9177
Configurazione 2: RMSE Percentuale = 23.30%, MAE Percentuale = 23.30%,↙
R² = 0.9206
Configurazione 3: RMSE Percentuale = 23.45%, MAE Percentuale = 23.45%,↙
R² = 0.9196
La configurazione con il RMSE percentuale più basso è la configurazione↙
2 con RMSE = 23.30%
La configurazione con il MAE percentuale più basso è la configurazione 2↙
con MAE Percentuale = 23.30%
La configurazione con il R² più alto è la configurazione 2 con R² =↙
0.9206
>>
```

Figure 5.3: Comparison of the different configurations based on the total number of neurons (4 layers)

In this case, the best configuration is: $[38, 32, 22, 25]$.
For what regards the last configuration with only three layers, the best one is $[40, 36, 32]$.

```
MATLAB Command Window                                    Page 1

Configurazione 1: RMSE Percentuale = 23.77%, MAE Percentuale = 23.77%,↙
R² = 0.9157
Configurazione 2: RMSE Percentuale = 23.43%, MAE Percentuale = 23.43%,↙
R² = 0.9163
Configurazione 3: RMSE Percentuale = 23.21%, MAE Percentuale = 23.21%,↙
R² = 0.9168
La configurazione con il RMSE percentuale più basso è la configurazione↙
3 con RMSE = 23.21%
La configurazione con il MAE percentuale più basso è la configurazione 3↙
con MAE Percentuale = 23.21%
La configurazione con il R² più alto è la configurazione 3 con R² =↙
0.9168
>>
```

Figure 5.4: Comparison of the different configurations based on the total number of neurons (3 layers)

So the final three configurations are:

Configuration	# of parameters	RMSEperc	MAEperc	R^2
[38,32,17,25,17]	2920	23.71%	23.71%	0.9220
[38,32,22,25]	2800	23.30%	23.30%	0.9206
[40,36,32]	2938	23.21%	23.21%	0.9168

The three configurations are very similar, but in the end, as a compromise between accuracy of "structure complexity" and the condition of the number of total parameters, the second is the best choice.

5.2 Split model with training and validation sets

The algorithm remains the same as previously described (including the size of data set). The main difference is the introduction of a split model. One model is dedicated to the "filtered" data points, using the previously selected hidden layer configuration, while the other processes the remaining data with the original configuration. Once both simulations are complete, the results are combined and compared with the previous model to assess any improvement in efficiency.

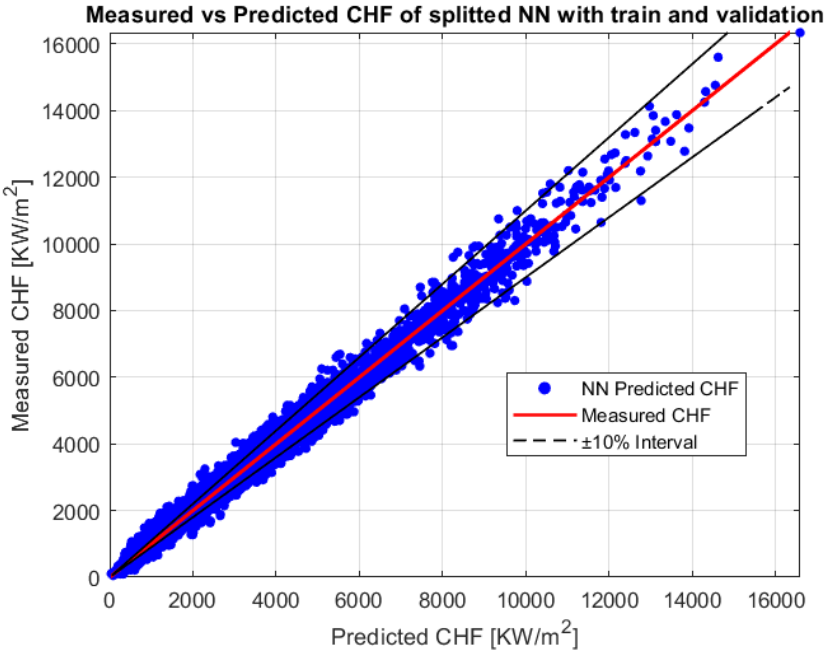


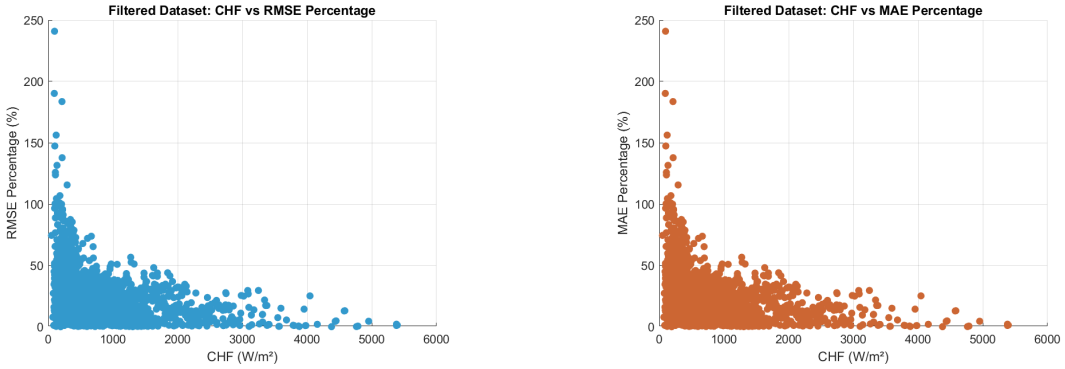
Figure 5.5: Measured vs predicted CHF for a NN with only train and validation data subdivision

The first thing we can see from the figure is that the majority of predicted CHFs are inside the "ideal" interval, which means that the improvement is made. The measure of this improvement is based on the statistical metrics discussed earlier, as noted in table 5.1.

ALL	**Size of the data set**		**24579 samples**
	Evaluation	Mean P/M	1.017
		Std P/M	0.126
		RMSPE [%]	12.746
		MAPE [%]	8.385
		NRMSE	0.092
		Q^2 error	0.011

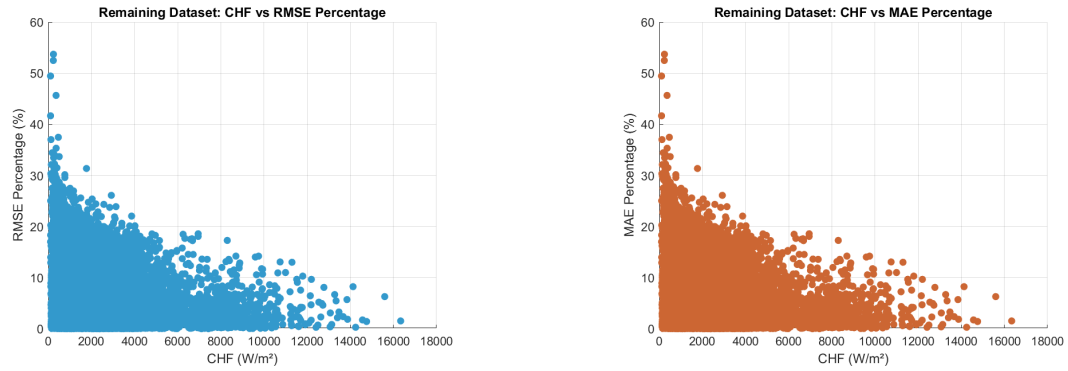
Table 5.1: split NN model with only training and Validation prediction performances

Regarding the distribution of RMSE and MAE percentage errors , we have a pair of plots for the filtered database (Figure 5.6) and one for the remaining (Figure 5.7), in order to show the difference in terms of accuracy between the two parts. It reflects the initial split due to the different value of the P/M ratio. In appendix C there are the other distribution of these errors versus the other input variables.



(a) RMSE [%] (b) MAE [%]

Figure 5.6: RMSE and MAE percentage error distribution for filtered points of database of the split network with training and Validation set



(a) RMSE [%] (b) MAE [%]

Figure 5.7: RMSE and MAE percentage error distribution for remaining points of database of the split network with training and Validation set

When comparing this model with the original NN network, the respective metrics are coupled to find the relative error difference between the former and the latter. The percentage difference has been set using the metrics of the first model as a reference. So the graph is a kind of demonstration of how these metrics change between the two models. We find that the split model is better than the previous one, as shown in the figure.

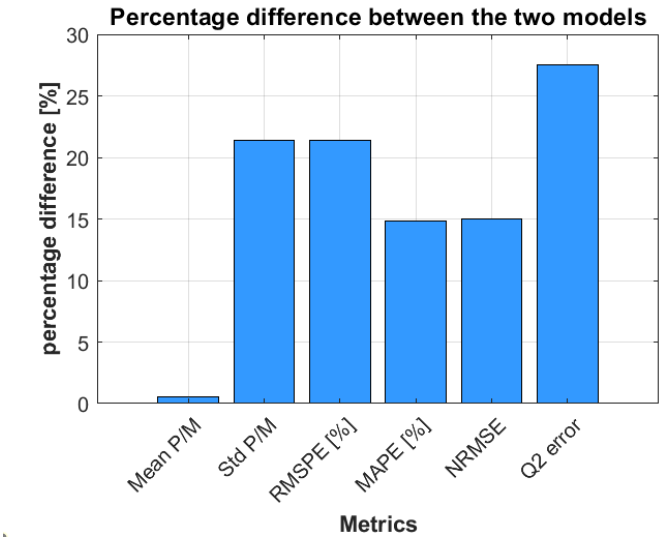


Figure 5.8: Relative error difference between the original NN model and the split one with training and Validation set

Another check was performed by comparing the $\frac{P}{M}$ ratio of the two models. In the next figures we have the distribution of this ratio as a function of some input variables.

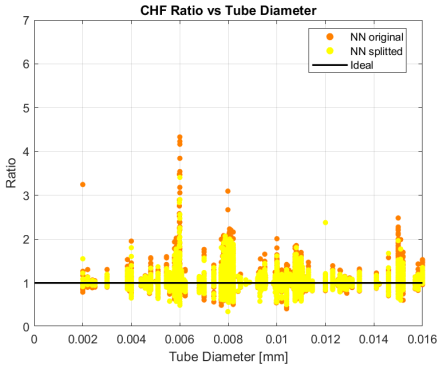


Figure 5.9: P/M Ratio vs Tube Diamete

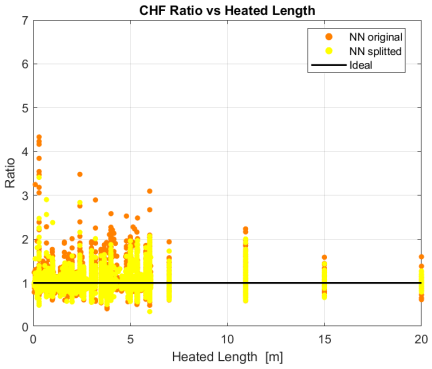


Figure 5.10: P/M Ratio vs Heated Length

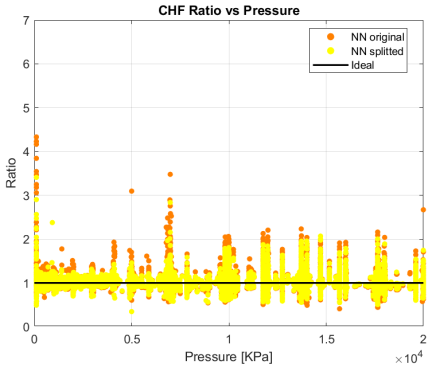


Figure 5.11: P/M Ratio vs Pressure

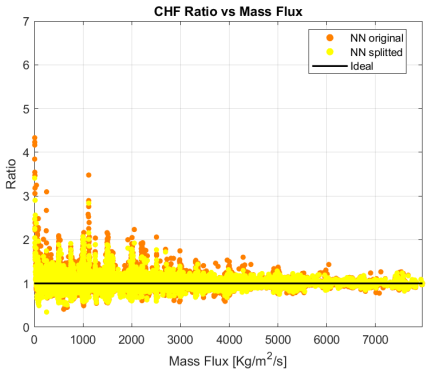


Figure 5.12: P/M Ratio vs Mass Flux

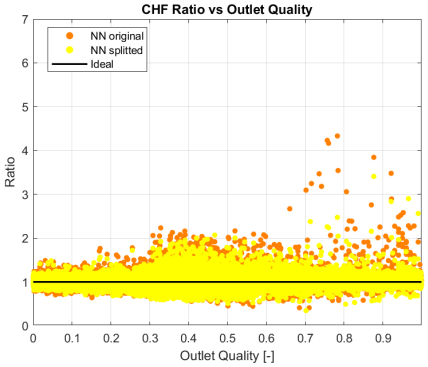


Figure 5.13: P/M Ratio vs Outlet Quality

Figure 5.14: Original NN (orange) and split NN (yellow) Predicted over Measured CHF scatter plots, vs selected independent parameters. NN model with only training and Validation

5.3 Split model with training. validation and test sets

For the model split by adding test data, we used the same procedure as for the original network with the separate test. That is, for the two separate models, the respective databases (the one with the filtered data and the one with the remaining data) were taken and the test data were randomly extracted using the same separation percentage as in the original model (hence 15% of the total). Once these data were saved, the network development continued, using the previous split model layout for the filtered data model without testing and the previous optimal structure for the rest.

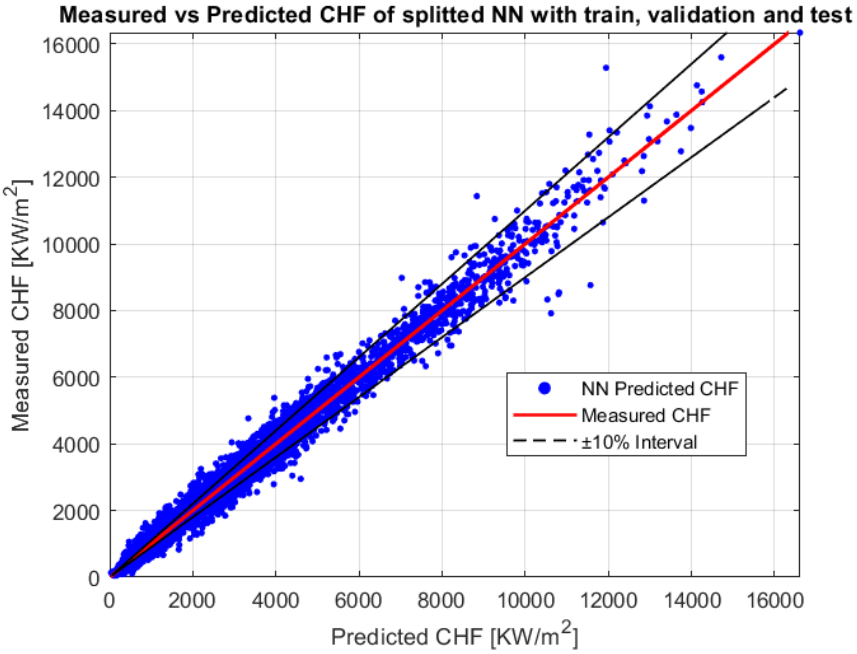


Figure 5.15: Measured vs predicted CHF for a NN with training validation and test subdivision

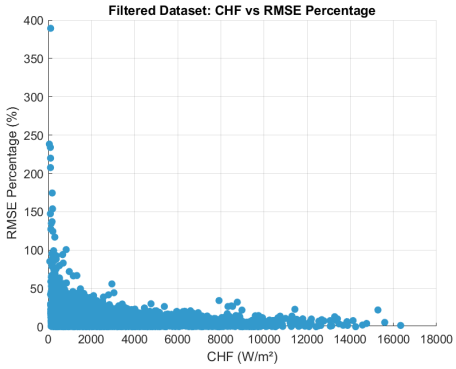
Also for this case, we can observe the same thing seen in previous figure.

In addition, once the results were obtained, the statistical metrics were combined and compared with the original model.

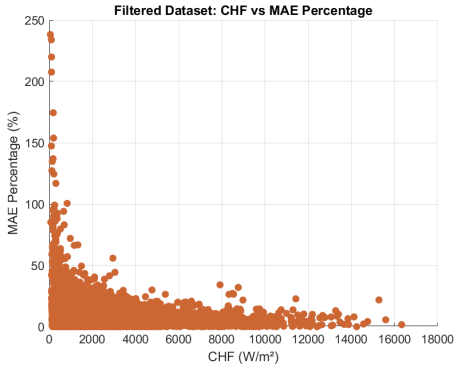
TEST	**Size of the data set**		**3687 samples**
	Evaluation	Mean P/M	1.020
		Std P/M	0.202
		RMSPE [%]	20.302
		MAPE [%]	11.916
		NRMSE	0.129
		Q^2 error	0.022
ALL	**Size of the data set**		**24579 samples**
	Evaluation	Mean P/M	1.012
		Std P/M	0.148
		RMSPE [%]	14.904
		MAPE [%]	9.339
		NRMSE	0.105
		Q^2 error	0.013

Table 5.2: split NN model with training, Validation and Test prediction performances

The distribution of the percentage errors follows the previous section instruction: first the filtered data set and then the remaining one.

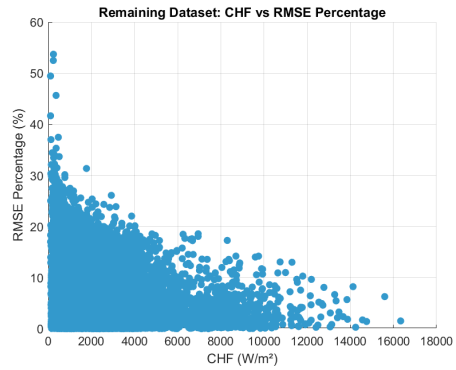


(a) RMSE [%]

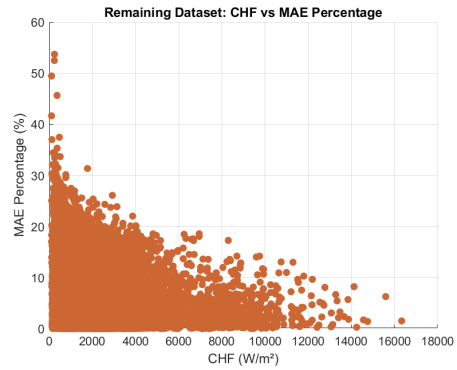


(b) MAE [%]

Figure 5.16: RMSE and MAE percentage error distribution for remaining points of database of the split network with training and Validation set



(a) RMSE [%]



(b) MAE [%]

Figure 5.17: RMSE and MAE percentage error distribution for remaining points of database of the split network with training and Validation set

Also for this case, we do the same comparison with the original NN model with training, validation and test and we get an increase of the overall performance with respect to the original one. So we can say that also for this model we have an improvement in the accuracy and efficiency of predicting CHF.

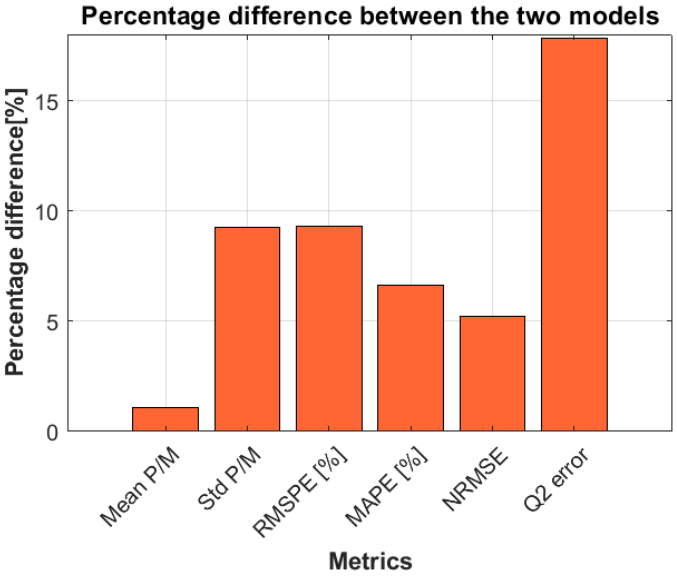


Figure 5.18: Relative error difference between the original NN model and the split one with training, Validation and Test

An additional check can be done by comparing the P/M ratio plots of the two models. We can see in all the scatter plots that the second one is closer to the ideal value compared to the original model.

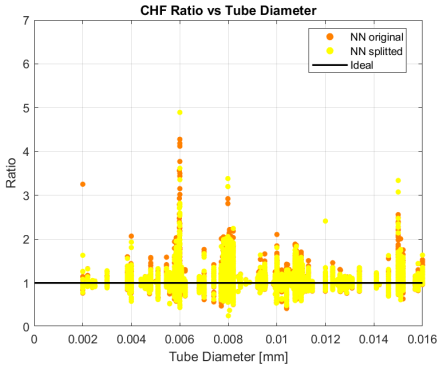


Figure 5.19: P/M Ratio vs Tube Diameter

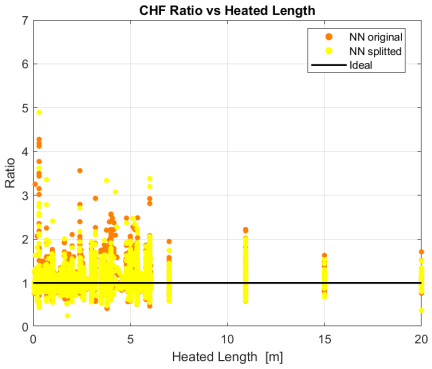


Figure 5.20: P/M Ratio vs Heated Length

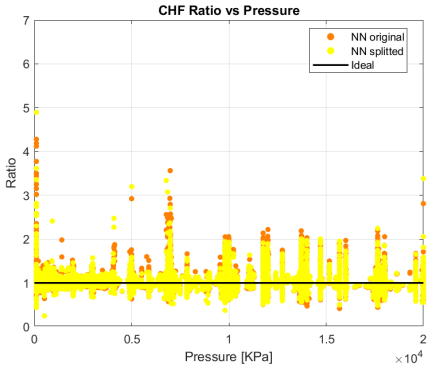


Figure 5.21: P/M Ratio vs Pressure

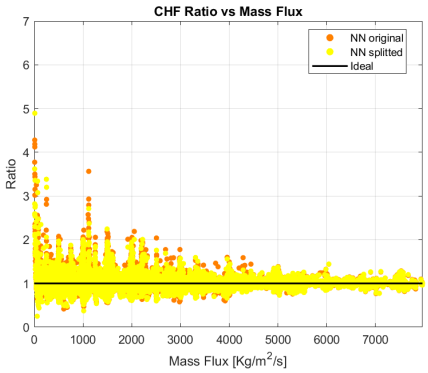


Figure 5.22: P/M Ratio vs Mass Flux

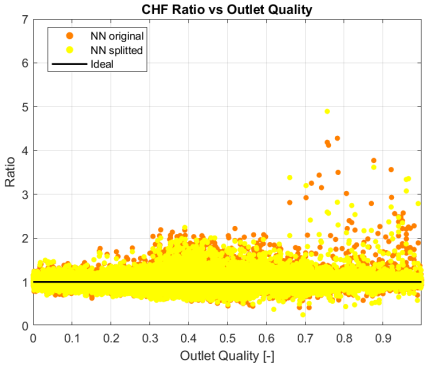


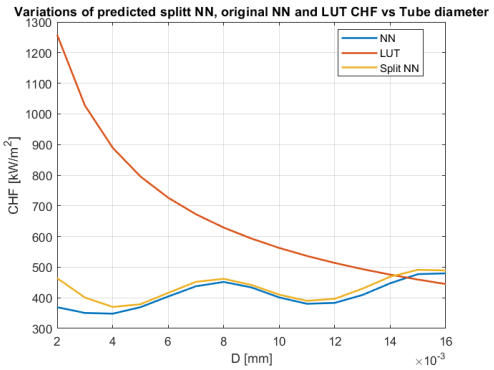
Figure 5.23: P/M Ratio vs Outlet Quality

Figure 5.24: Original NN (orange) and split NN (yellow) Predicted over Measured CHF scatter plots, vs selected independent parameters. NN model with training, Validation and Test

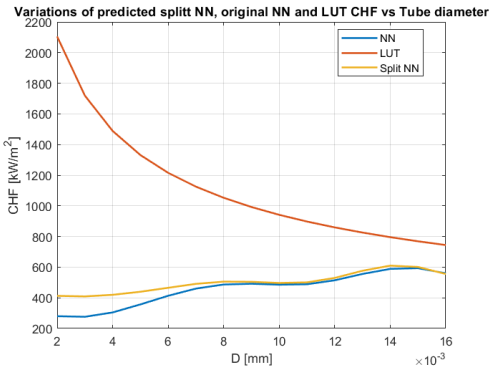
The results show that the choice of specialising the network, splitting it into two separate databases, one for the points where the $\frac{P}{M}$ ratio is worse respect than the threshold, and the remaining one, is a good move to improve the efficiency of the model for critical heat flux prediction.

5.4 Slice division of data set

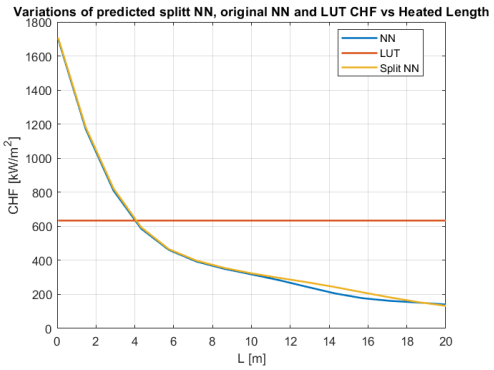
Since we are doing the slice subdivision for the original model, we can retrace the same graph by adding the latter mesh. On this way we can see how the distribution is with respect to the LUTs and the NN model previously performed.



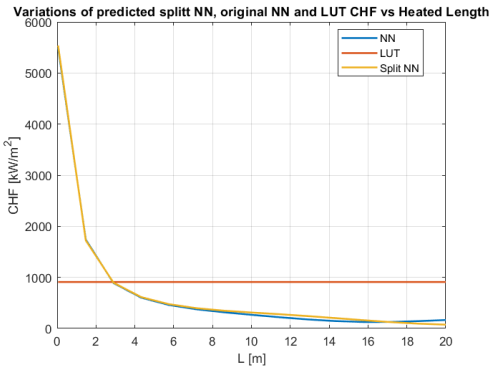
(a) The behaviour of the model and LUTs as a function of diameter (slice 1)



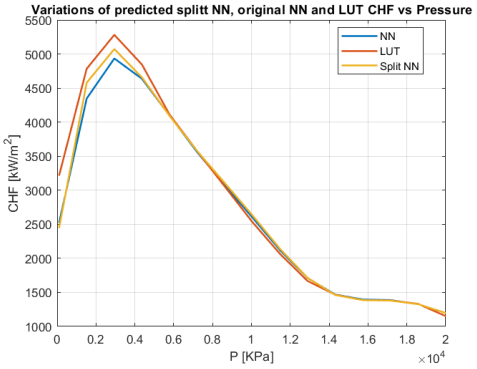
(b) The behaviour of the model and LUTs as a function of diameter (slice 2)



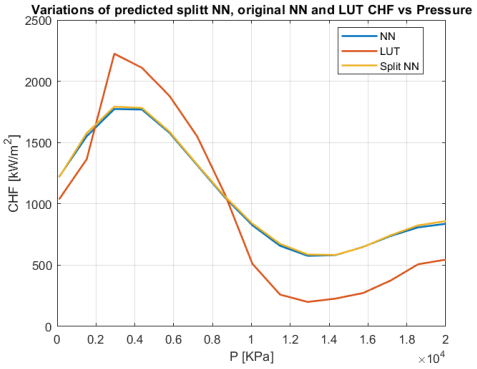
(c) The behaviour of the model and LUTs as a function of heated length (slice 3)



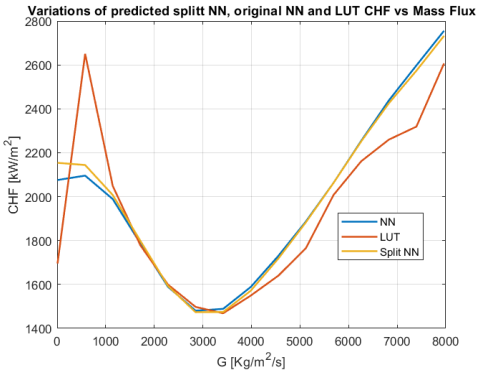
(d) The behaviour of the model and LUTs as a function of heated length (slice 4)



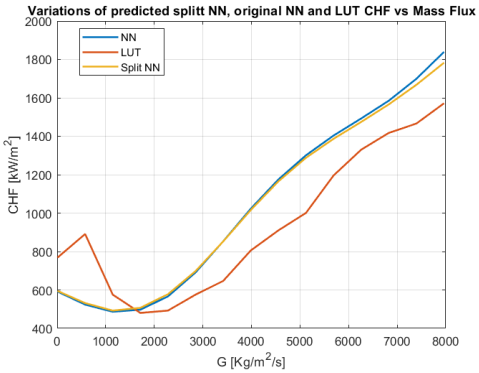
(e) The behaviour of the model and LUTs as a function of pressure (slice 5)



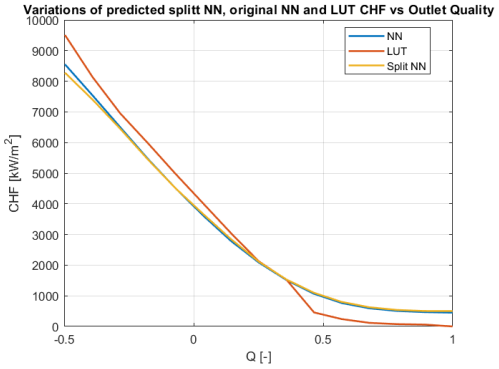
(f) The behaviour of the model and LUTs as a function of pressure (slice 6)



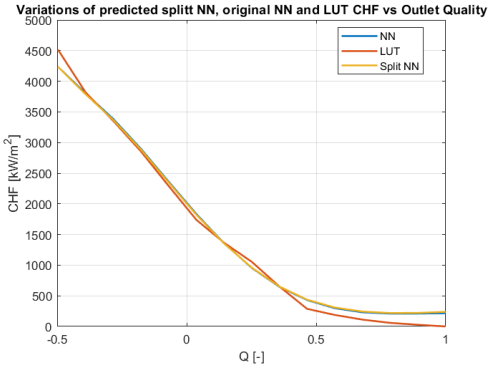
(g) The behaviour of the model and LUTs as a function of mass flux (slice 7)



(h) The behaviour of the model and LUTs as a function of mass flux (slice 8)



(i) The behaviour of the model and LUTs as a function of outlet quality (slice 9)



(j) The behaviour of the model and LUTs as a function of outlet quality (slice 10)

Figure 5.25: Comparison of the model and LUTs behaviour as a function of different parameters

Chapter 6

The Log-transformed Neural Network

This chapter dealts with another possible approach to improve the neural network model for better CHF prediction. Instead of splitting the data set, as discussed in Chapter 5, here a logarithmic function is adopted. Comparing the maximum and minimum values of each input variable and the output of the data set, we can see that the "distance" between these two values is quite large for heated length, pressure, mass flux and the CHF. Applying the logarithm transformation decreases the discrepancy between these extreme values in order to make the distribution wider, as shown in the figures below (6.1, 6.2, 6.3, 6.4). This leads to an improvement of numerical stability during the training phase of the net. Script implementation is quite the same as the previous model, the difference is when we record whole the data set where the logarithm is applied:

```
all_data_in = data(1:5,:); % Input features
all_data_out = data(end,:); % Output
all_data_in(2:4, :) = log(all_data_in(2:4, :));
all_data_out = log(all_data_out);
```

Of course, since the network is trained with log-transformed data, before applying the metrics performance, we must convert the data to the original scale using the exponential:

```
CHF_predictions_mean_den = exp(CHF_predictions_mean_den);
CHF_predictions_std_den = std(CHF_predictions_den,0,2);
all_out = exp(all_out);
train_and_valid_out = exp(train_and_valid_out);
test_out = exp(test_out);
```

Figures below show how the transformation reduces the distance between minimum and maximum values and makes the distribution more uniform.

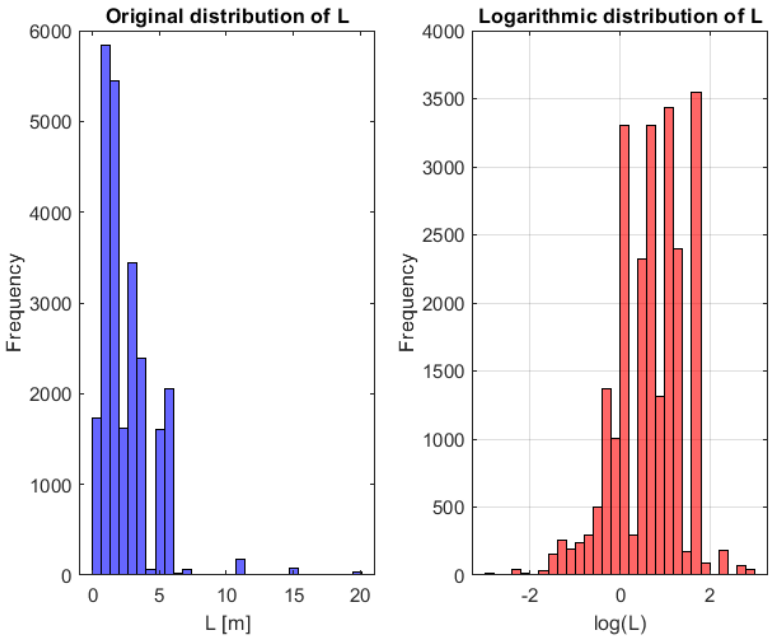


Figure 6.1: Distribution of heated length values

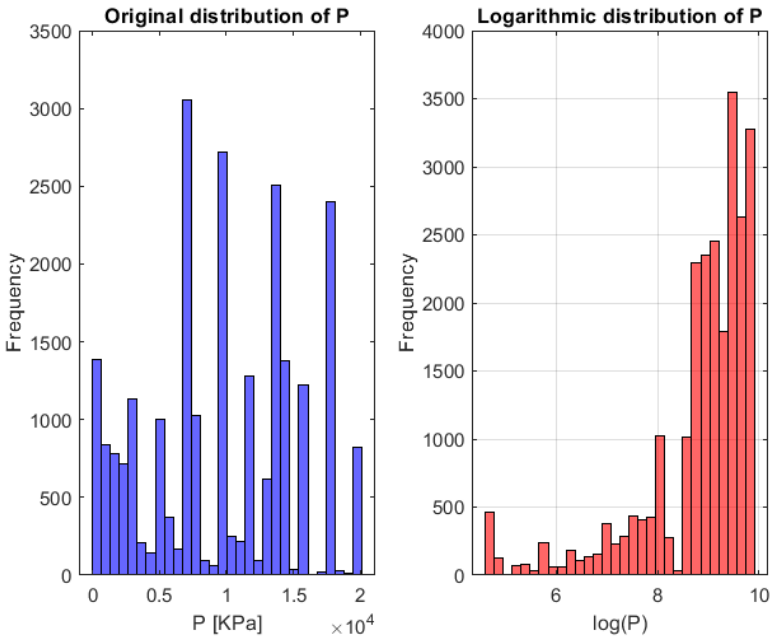


Figure 6.2: Distribution of pressure values

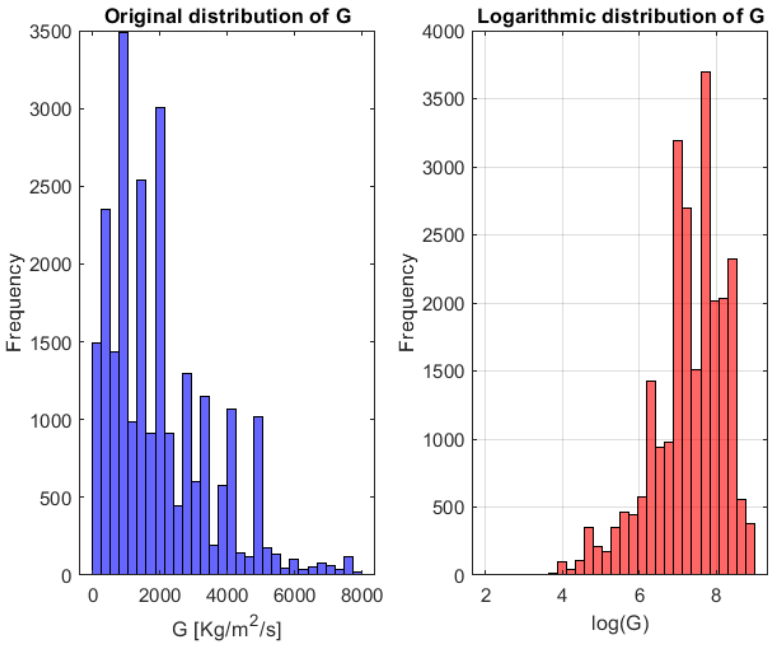


Figure 6.3: Distribution of mass flux values

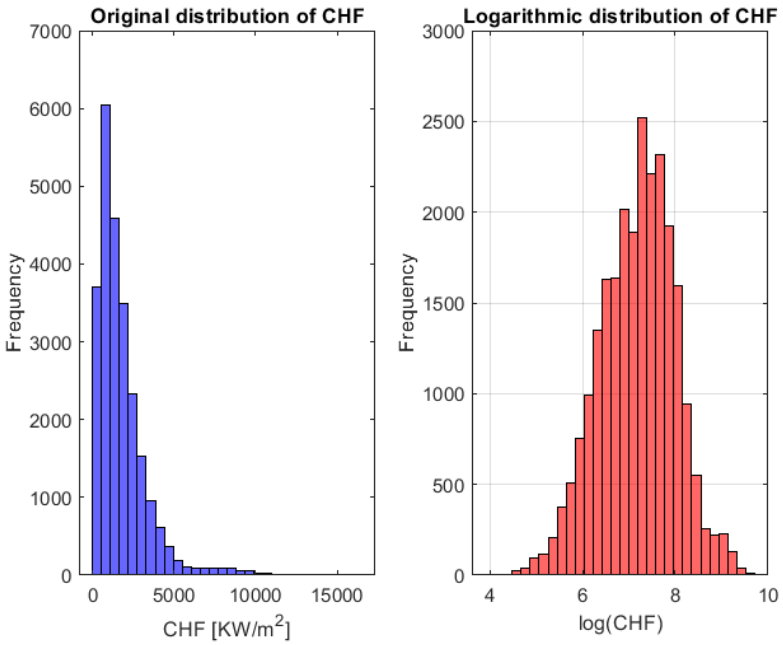


Figure 6.4: Distribution of CHF values

6.1 Results

The predictive performance of the log transformation is evaluated using the same metrics as the previous models. For this model, we only evaluate the case of the model with training, validation and test sets. If we have an improvement in performance, it is certain that we have the same for the model with only training and validation. As described in the section of results for chapter 4 and 5, also here we record the comparison between the measured vs predicted CHF (figure 6.5) and metrics performance (table 6.1)

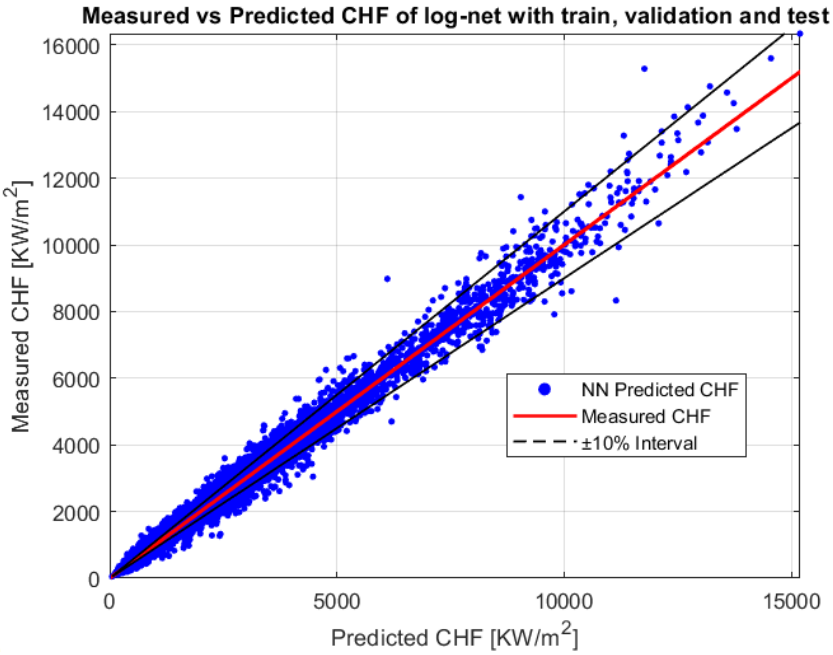
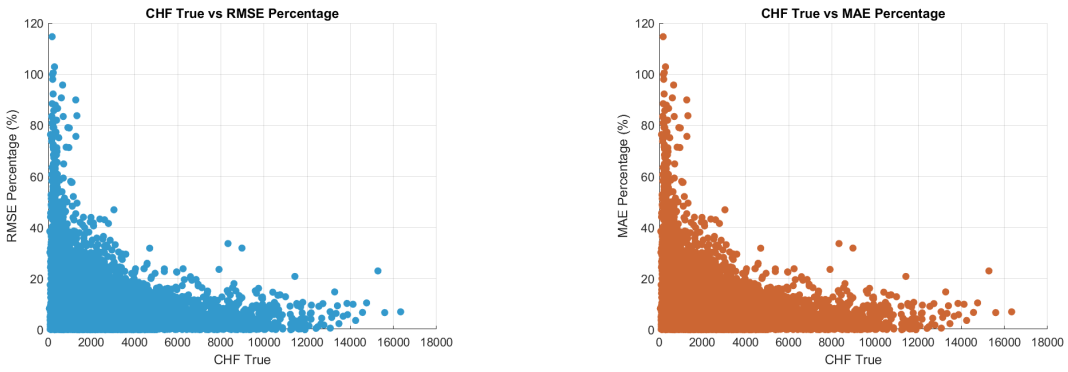


Figure 6.5: Measured vs predicted CHF for the log-net with train validation and test subdivision

TEST	**Size of the data set**		**3073 samples**
	Evaluation	Mean P/M	1.005
		Std P/M	0.117
		RMSPE [%]	11.734
		MAPE [%]	7.725
		NRMSE	0.106
		Q^2 error	0.014
ALL	**Size of the data set**		**24579 samples**
	Evaluation	Mean P/M	1.006
		Std P/M	0.113
		RMSPE [%]	11.268
		MAPE [%]	7.382
		NRMSE	0.099
		Q^2 error	0.0126

Table 6.1: Performance Evaluation for TEST and ALL data sets

The distribution of percentage errors, like before, is represented in 6.6, while the percentage difference between this model and the original one is represented in figure 6.7. The others are in Appendix D.



(a) RMSE [%] (b) MAE [%]

Figure 6.6: RMSE and MAE percentage error distribution for neural network with Train, Validation and Test set

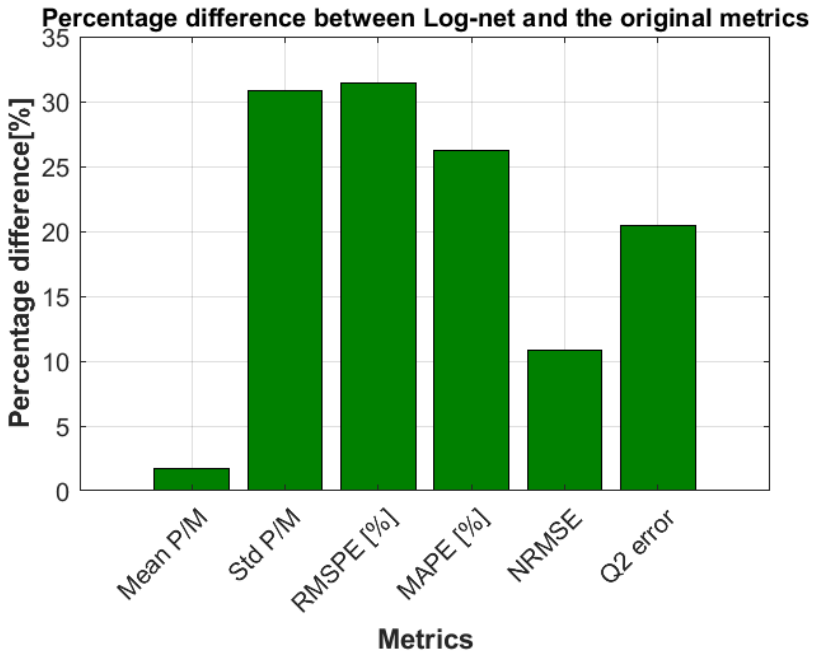


Figure 6.7: Relative error difference between the original NN model and the log-transformed one with training, validation and test sets

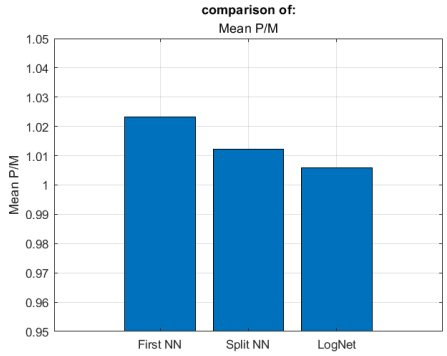
According to these results, we notice that the log-net has better accuracy compared to the original, confirming a better predictive ability. In the next section a final comparison is made between the three network models, in order to demonstrate that this model is the best one.

6.2 Final comparison between the three models

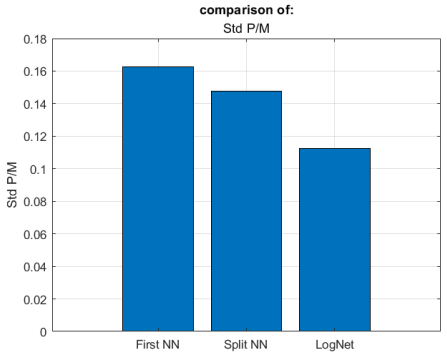
In this section, we make a final comparison between the three models and assess how the changes in the setup improve the performance of the CHF predictions. The key of the presentation is to analyze the performance of metrics: the **lognet model** shows a better CHF prediction with respect to the first model and the split one. This is drawn also by the figures below.

- **Mean and std $\frac{P}{M}$ ratio (figures 6.8a and 6.8b)**: the lognet model predictions are closer to the expected values, infact it shows the smallest value for both the two metrics. It means that the prediction is so consistent and more efficient with respect to other two models.
- **Root Mean Square Percentage Error (RMSE) and Mean Absolute Percentage Error (MAPE) (6.8c and 6.8d)**: according to the definition of these metrics, the lognet model overperform respect to the other, so this means that we have a small discrepancy between the predicted CHF and its expected value.
- **Normalized Root Mean Square Error (NRMSE) and Q^2 error (6.8e and 6.8f)**: The lognet model have a better predictive power.

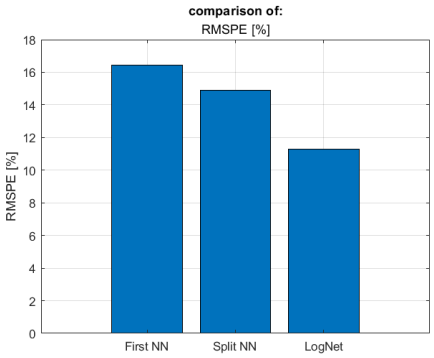
Overall, we can say that modifying the layout of the network, splitting it into two databases: one for the points that have a worse prediction and the others, or using the logarithm, reducing the distance between the maximum and minimum range of each variable, improve the CHF prediction. However, the latter is the best choice because it is computationally simpler, since we have only one model to simulate instead of two, and it is more efficient, as reflected in all the figures analysed before.



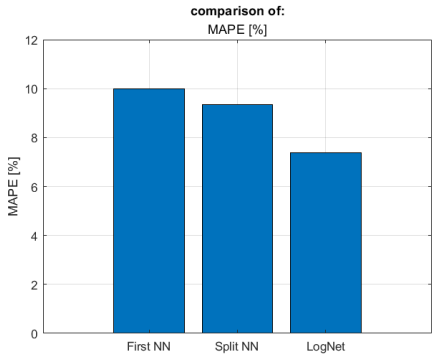
(a) Mean P/M ratio comparison



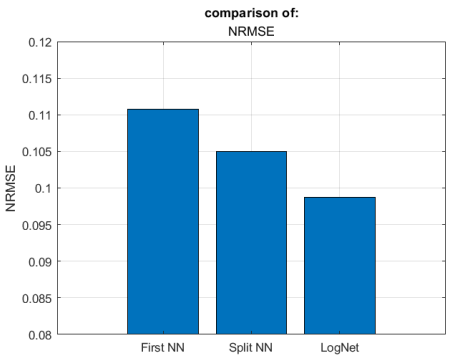
(b) Std P/M ratio comparison



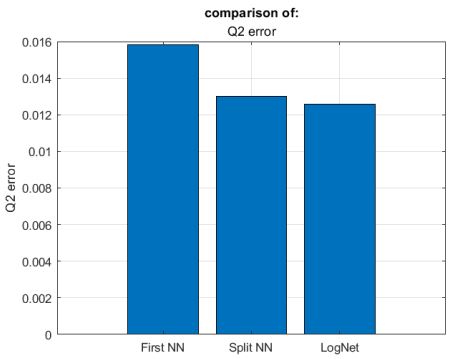
(c) RMSPE comparison



(d) MAPE comparison



(e) NRMSE comparison



(f) Q^2 error comparison

Figure 6.8: Network models final metrics comparison

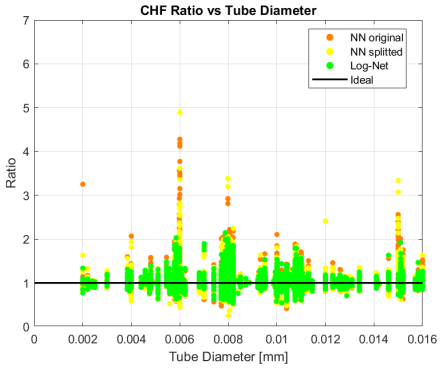


Figure 6.9: P/M Ratio vs Tube Diameter

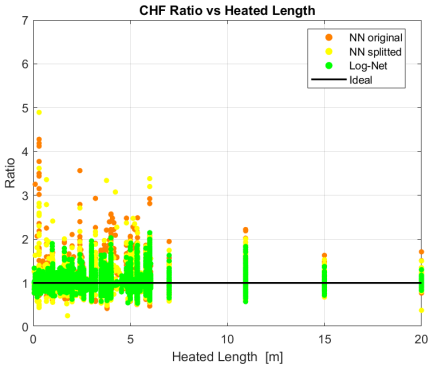


Figure 6.10: P/M Ratio vs Heated Length

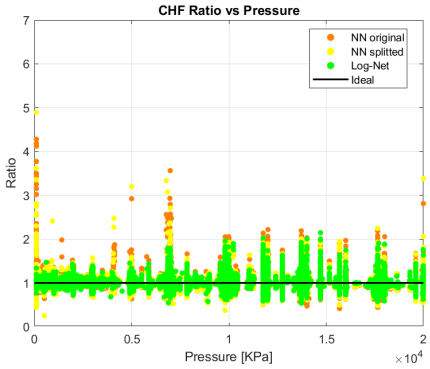


Figure 6.11: P/M Ratio vs Pressure

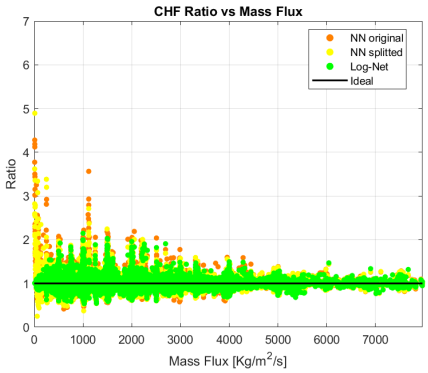


Figure 6.12: P/M Ratio vs Mass Flux

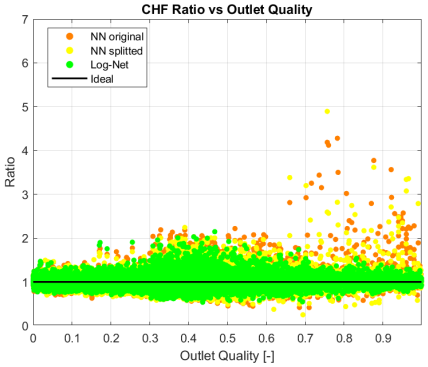


Figure 6.13: P/M Ratio vs Outlet Quality

Figure 6.14: Original NN (orange), split NN (yellow) and log-net (green) Predicted over Measured CHF scatter plots, vs selected independent parameters. NN model with Train, Validation and Test

Chapter 7

Conclusion and perspective

The construction of artificial intelligence models for the prediction of Critical Heat Flux (CHF) is a major advance in nuclear reactor safety and optimization. In this work the construction and analysis of artificial neural networks has been carried out, comparing the results with those of traditional models, in accordance with the Task Force provided by the EGMUP as mentioned in section 3.3. In the first part of this work, the main mechanisms of the CHF inside a reactor are described, from a physical and empirical point of view, through the study of Look-Up Tables (LUTs). Then, based on this knowledge, an artificial intelligence model was constructed by selecting the most influential parameters (chapter 4).
The first results show that it is possible to obtain greater efficiency with respect to the empirical LUTs, reducing error discrepancies and obtaining an adaptable model for specific operating conditions. Several tests, optimizations, analysis and comparisons have been carried out in order to obtain a potential tool for CHF prediction.
In addition, there are possible ways to make the model more efficient: **specializing network** (Chapter 5), splitting the data set into two parts, according to statistical metrics such as the P/M ratio, or applying **a logarithmic transformation of the input** (Chapter 6), where the discrepancy between the maximum and minimum values is so large that it is reduced and the data made numerically stable for the training of the network.
The comparison between these three models, with the results in section 6.2, was based on the performance of several metrics: Root Mean Squared Percentage Error (RMSPE), Mean Absolute Percentage Error (MAPE), Normalized Root Mean Squared Error (NRMSE), EQ^2 error and the P/M ratio with its mean value and standard deviation. Of these we chose as the key metrics the P/M ratio, because it assesses how well a model's predictions match the observed data.
From this comparison we can see that the log-transformed neural network is the best choice to improve the first model. It is also simpler from computational point

of view, because we only have one model and not two separate ones.
The overall results show how the machine learning can overcome the limitations of previous empirical methodologies, being a potential tool for the future research to solve security and other issues. Anyway, new actions will be taken for the future, according to the timeline and plans of the Task Force[7]:

- Improve the accuracy and robustness of the model using advance uncertainty quantification and Data Augmentation techniques;
- Adapting of the model to "complex" geometries with the aim of applying the physics learned from a given database to a different case (for example CHF in subchannels of a rod bundle);
- Model interpretability and explainability to make the AI adaptable in any industrial field;
- Fuel bundle benchmark: As a final step of the whole Task Force, a large CHF database will be provided to allow a realistic application to nuclear power plant safety analysis.

This work, despite its limitations, is proposed as a preliminary but but hopefully valuable approach to artificial intelligence research to make an important contribution to the safety and operability of nuclear power plants.

Bibliography

[1] Mujid S. Kazimi Neil E. Todreas. *Nuclear Systems I - Thermal Hydraulic Fundamentals*. Massachusetts Institute of Technology, 1989.

[2] W. Liu, H. Nariai, and F. Inasaka. Prediction of critical heat flux for subcooled flow boiling. *International Journal of Heat and Mass Transfer, vol 43, pp. 3371-3390*, 2000.

[3] Cristina Bertani. Flow boiling. Course slides, 2023. Lecture slides, Politecnico di Torino.

[4] Francesco Taurisano. The perceptron. `http://www.ftlaboratory.com/`, 2017.

[5] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi, Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Elsevier Ltd*, 2018.

[6] Yue Jin, Koroush Shirvan, and Xingang Zhao. Constructing a new chf look-up table based on the domain knowledge informed machine learning methodology. *The 19th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-19)*, 2022.

[7] Jean-Marie Le Corre, Gregory Delipei, Xu Wu, and Xingang Zhao. Benchmark on artificial intelligence and machine learning for scientific computing in nuclear engineering. Technical report, NEA, 2022.

[8] Anup Bhande. What is underfitting and overfitting in machine learning and how to deal with it. https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76, 2018.

[9] John R. Lamarsch and Anthony J. Baratta. *Introduction to Nuclear Engineering*. Pearson, 2014.

[10] D.C. Groeneveld, J.Q. Shan, A.Z. Vasic, L.K.H. Leung, A. Durmayaz, J. Yang, S.C. Cheng, and A. Tanase. The 2006 chf look-up table. *ScienceDirect*, 2007.

[11] Theodore L. Bergman, Adrienne S. Lavine, Frank P. Incropera, and David P. Dewitt. *Introduction to heat transfer - Sixth edition.* John Wiley and Sons, Inc., 2011.

[12] S.Y. Ahmad. Axial distribution of bulk temperature and void fraction in a heated channel with inlet subcooling. *International Journal of Heat and Mass Transfer, volume 92, series C, Number 4*, 1970.

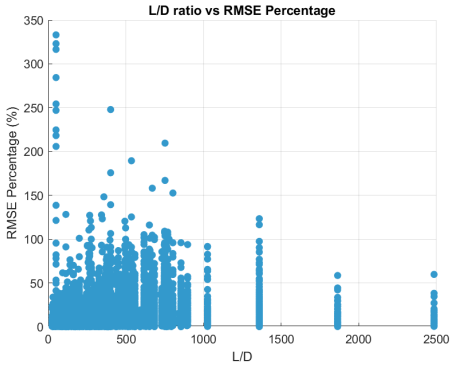
[13] Emil H. Grosfilley. Investigation of machine learning regression techniques to predict critical heat flux. *Uppsala University*, 2022.

[14] Pearson and spearman correlations: A guide to understanding and applying correlation methods. https://datascientest.com/en/pearson-and-spearman-correlations-a-guide-to-understanding-and-applying-correlation-methods, 2024.

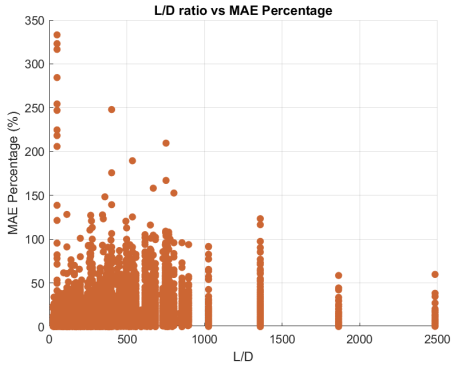
Appendix A

RMSE e MAE for the original model

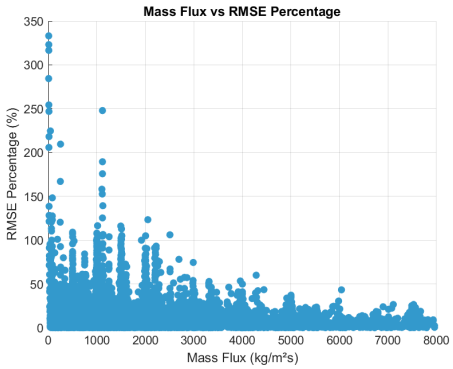
A.1 Model with training and validation sets



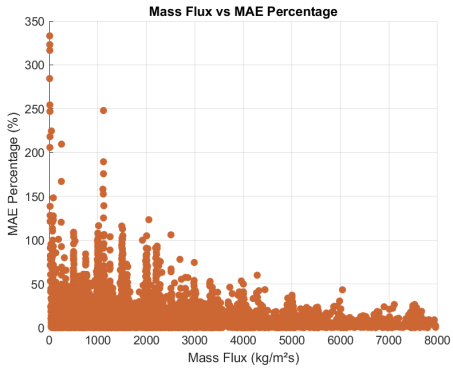
(a) RMSE vs L/D ratio



(b) MAE vs L/D ratio

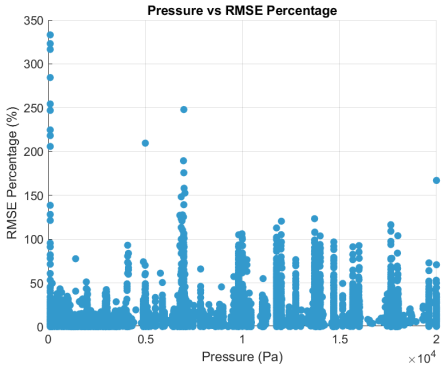


(c) RMSE vs Mass flux

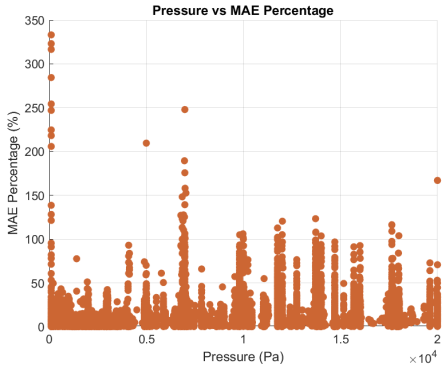


(d) MAE vs Mass flux

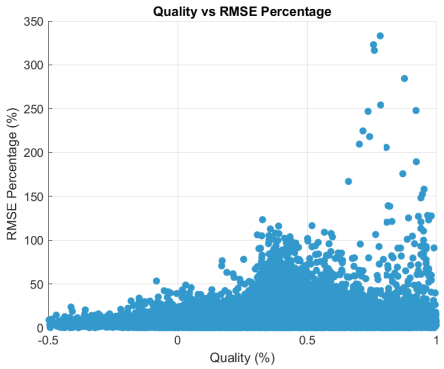
Figure A.1: RMSE and MAE percentage error distribution for neural network with training and validation set (Part 1)



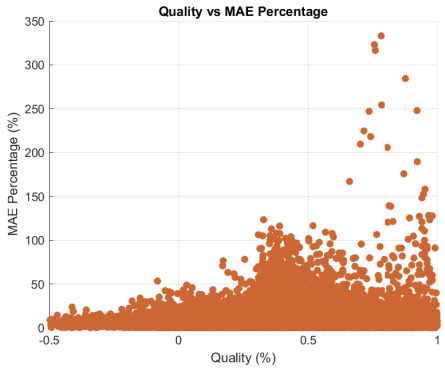
(e) RMSE vs Pressure



(f) MAE vs Pressure



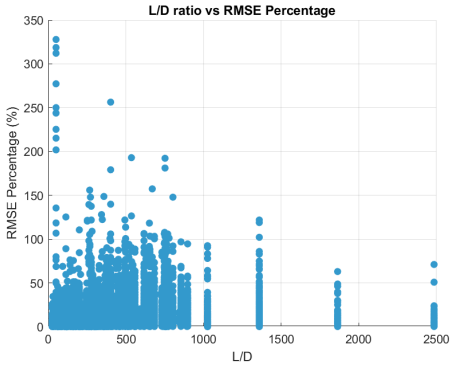
(g) RMSE vs Quality



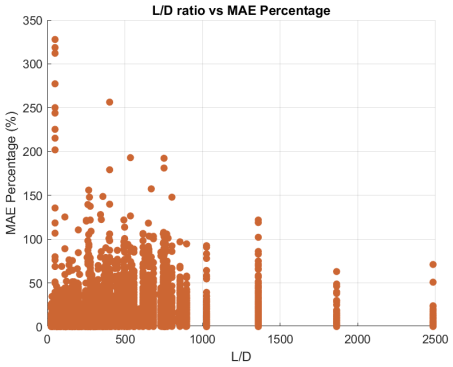
(h) MAE vs Quality

Figure A.1: RMSE and MAE percentage error distribution for neural network with training and validation set (Part 2)

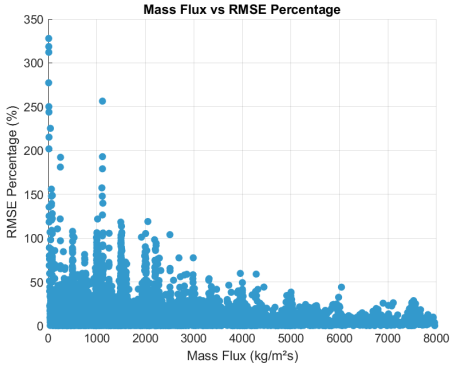
A.2 Model with entire set



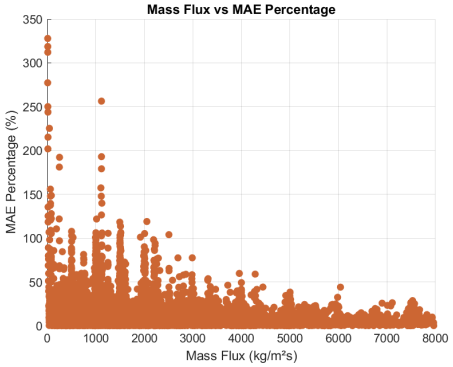
(a) RMSE vs L/D ratio



(b) MAE vs L/D ratio

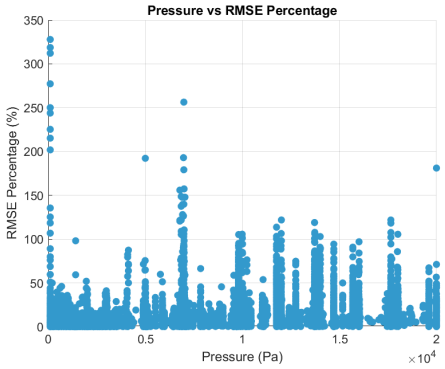


(c) RMSE vs Mass flux

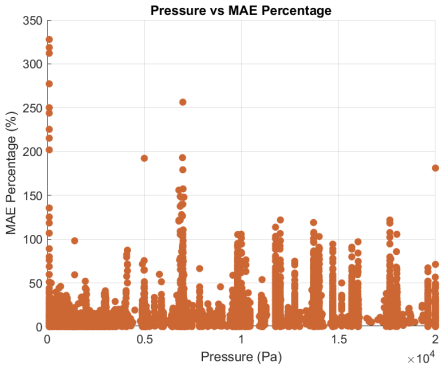


(d) MAE vs Mass flux

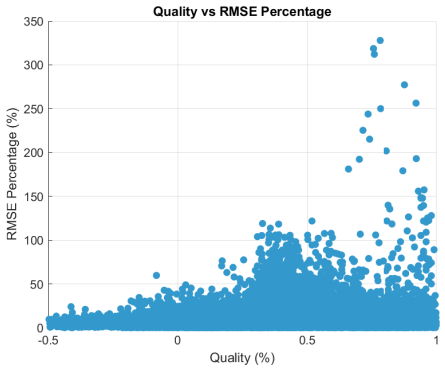
Figure A.2: RMSE and MAE percentage error distribution for neural network with training, validation and Test set (Part 1)



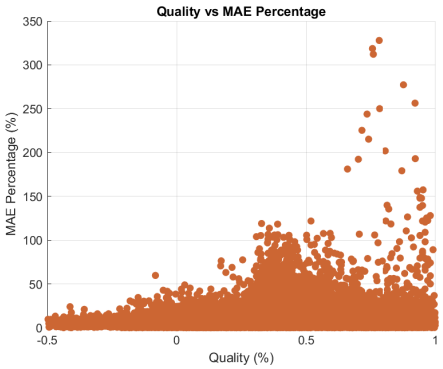
(e) RMSE vs Pressure



(f) MAE vs Pressure



(g) RMSE vs Quality



(h) MAE vs Quality

Figure A.2: RMSE and MAE percentage error distribution for neural network with training, validation and Test set (Part 2)

Appendix B

Configurations for P/M ratio threshold

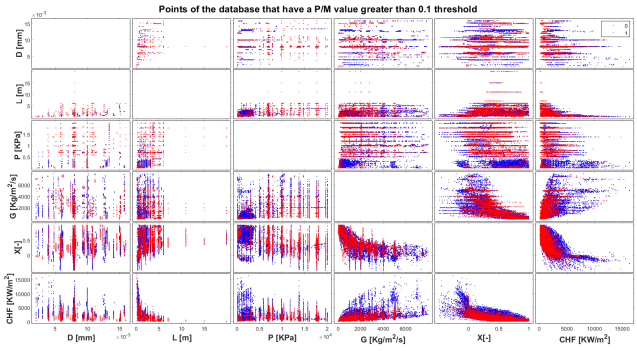


Figure B.1: CHF database with points where the $\frac{P}{M}$ ratio is greater than the 10% of threshold value

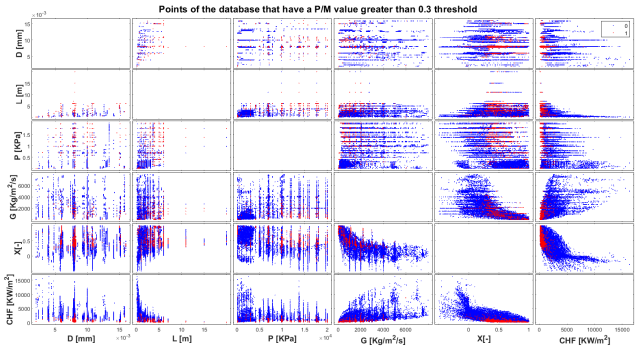


Figure B.2: CHF database with points where the $\frac{P}{M}$ ratio is greater than the 30% of threshold value

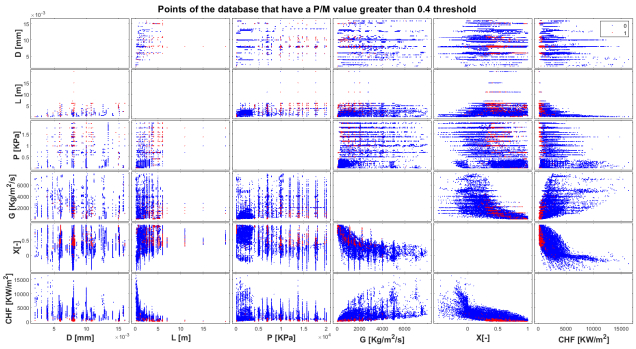


Figure B.3: CHF database with points where the $\frac{P}{M}$ ratio is greater than the 40% of threshold value

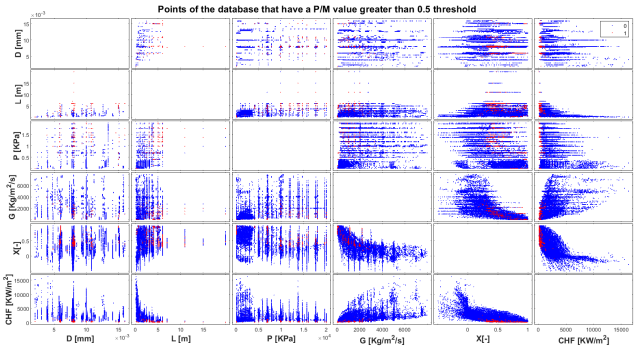
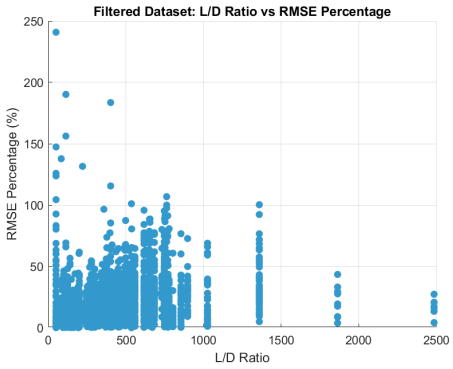


Figure B.4: CHF database with points where the $\frac{P}{M}$ ratio is greater than the 50% of threshold value

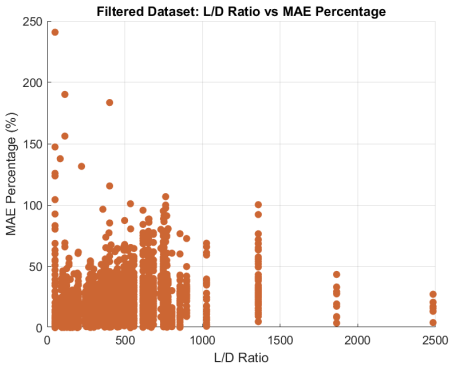
Appendix C

RMSE e MAE for split model

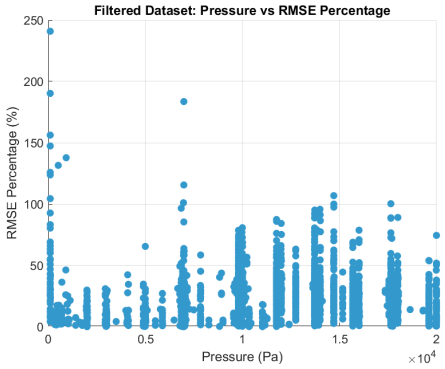
C.1 Split model with training and validation set - filtered points



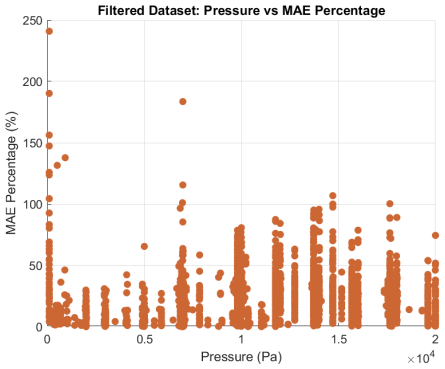
(a) RMSE vs L/D ratio



(b) MAE vs L/D ratio

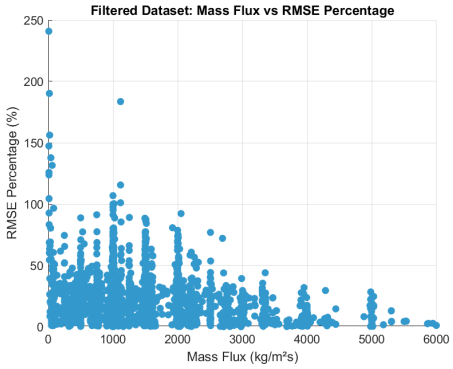


(c) RMSE vs Pressure

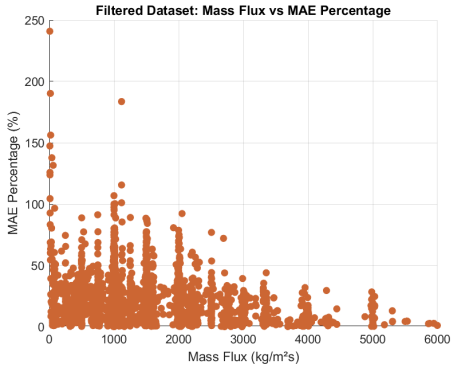


(d) MAE vs Pressure

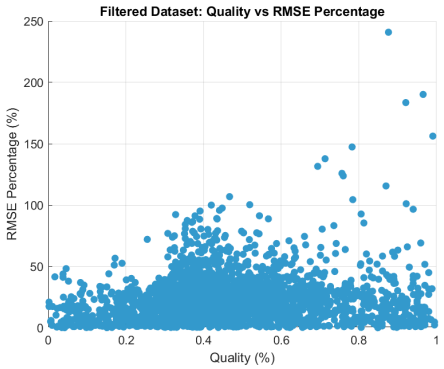
Figure C.1: RMSE and MAE percentage error distribution for neural network with training and validation set (Part 1)



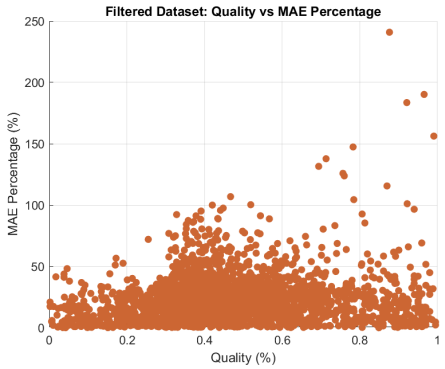
(e) RMSE vs Mass flux



(f) MAE vs Mass flux



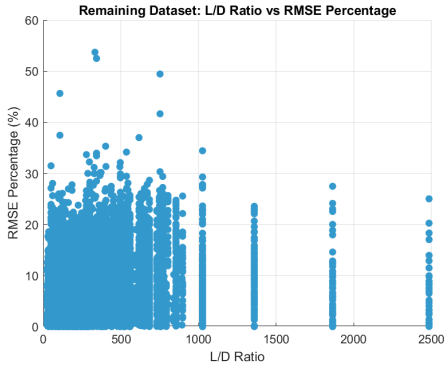
(g) RMSE vs Quality



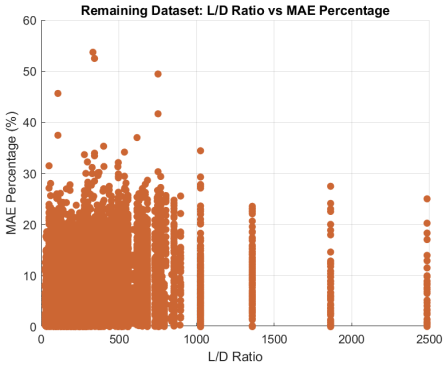
(h) MAE vs Quality

Figure C.1: RMSE and MAE percentage error distribution for neural network with training and validation set (Part 2)

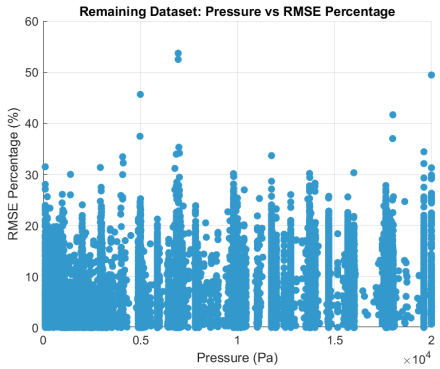
C.2 Split model with training and validation set - remaining points



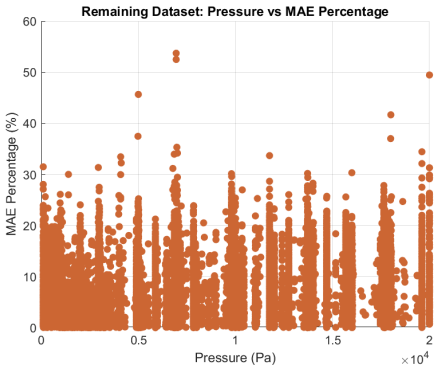
(a) RMSE vs L/D ratio



(b) MAE vs L/D ratio

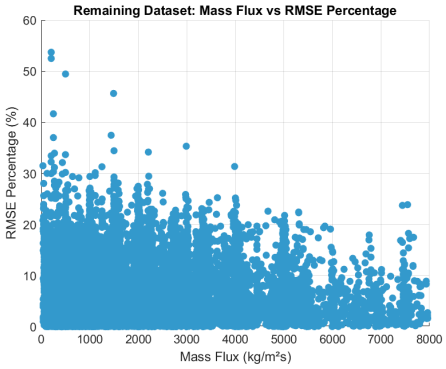


(c) RMSE vs Pressure

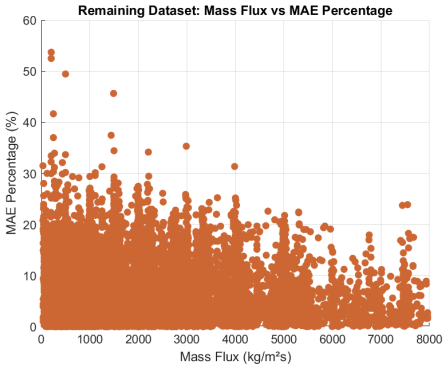


(d) MAE vs Pressure

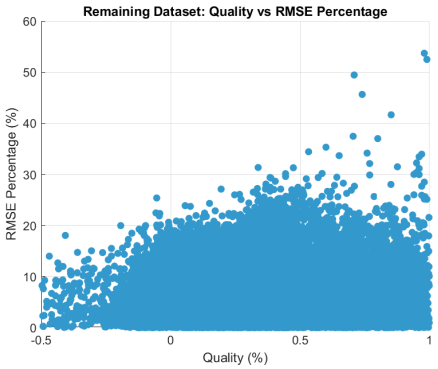
Figure C.2: RMSE and MAE percentage error distribution for neural network with training and validation set (Part 1)



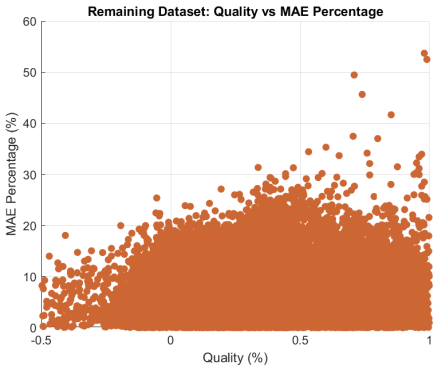
(e) RMSE vs Mass flux



(f) MAE vs Mass flux



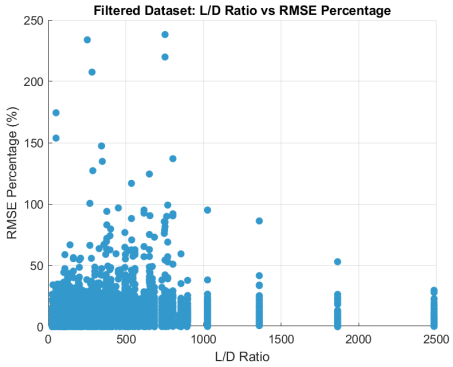
(g) RMSE vs Quality



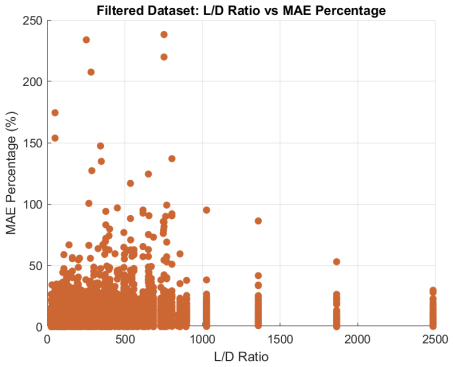
(h) MAE vs Quality

Figure C.2: RMSE and MAE percentage error distribution for neural network with training and Validation set (Part 2)

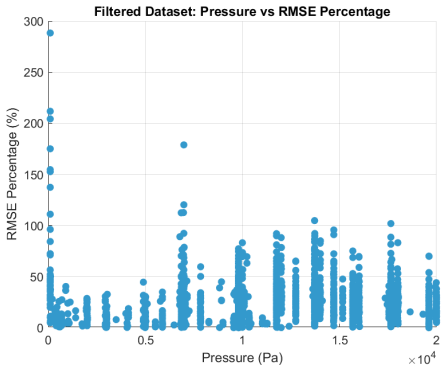
C.3 Split model with training, validation and test set - filtered points



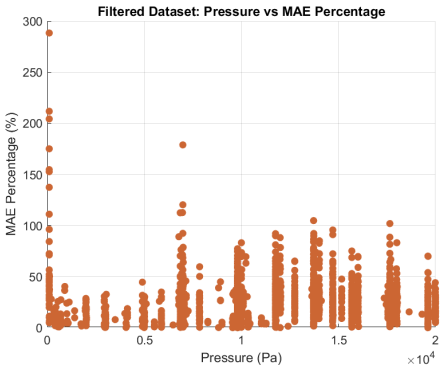
(a) RMSE vs L/D ratio



(b) MAE vs L/D ratio

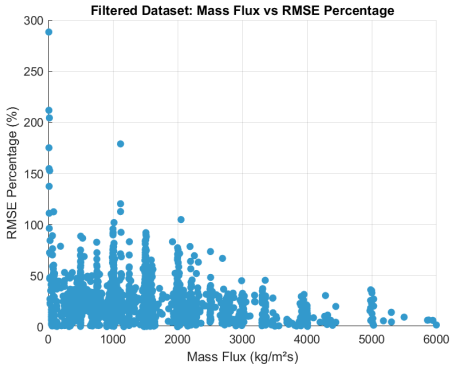


(c) RMSE vs Pressure

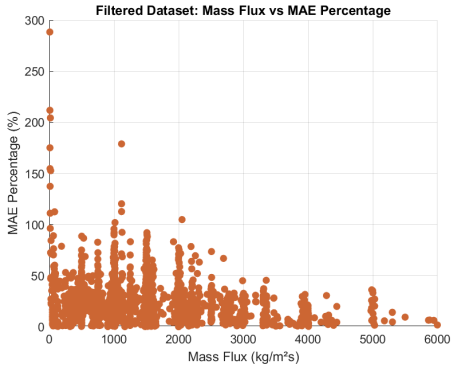


(d) MAE vs Pressure

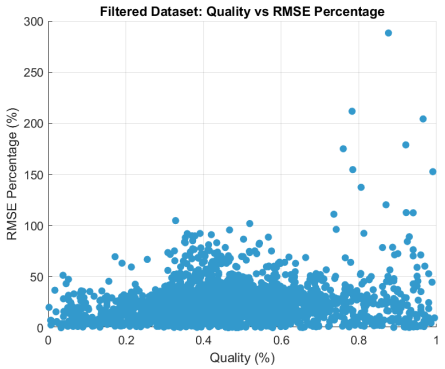
Figure C.3: RMSE and MAE percentage error distribution for neural network with training and Validation set (Part 1)



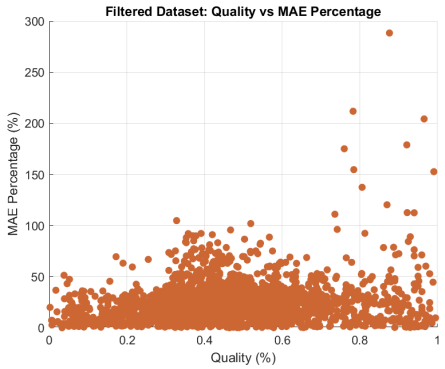
(e) RMSE vs Mass flux



(f) MAE vs Mass flux



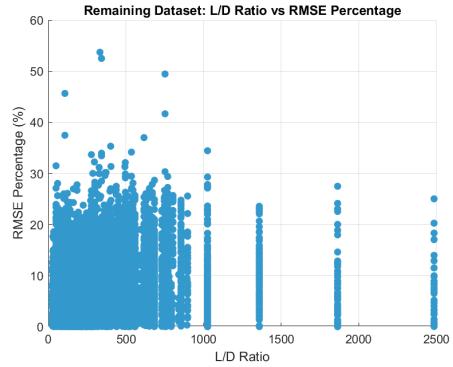
(g) RMSE vs Quality



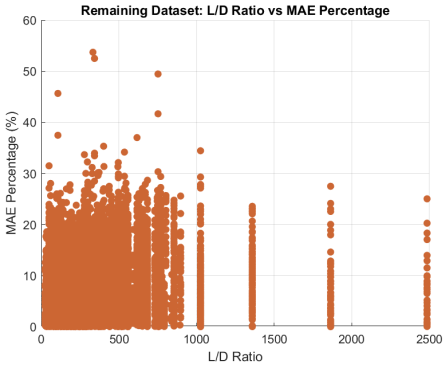
(h) MAE vs Quality

Figure C.3: RMSE and MAE percentage error distribution for neural network with training and Validation set (Part 2)

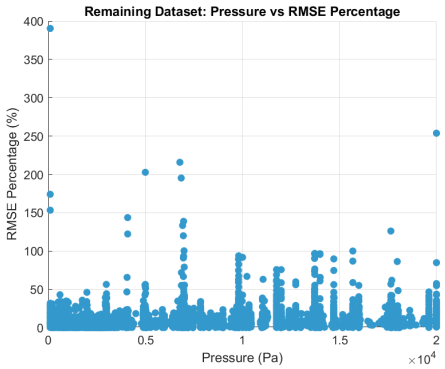
C.4 Split model with training, validation and test set - remaining points



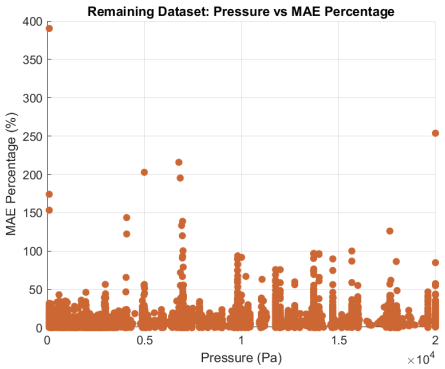
(a) RMSE vs L/D ratio



(b) MAE vs L/D ratio

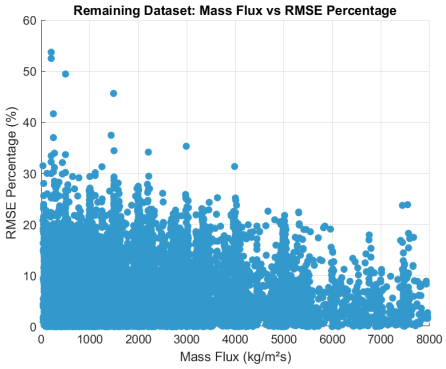


(c) RMSE vs Pressure

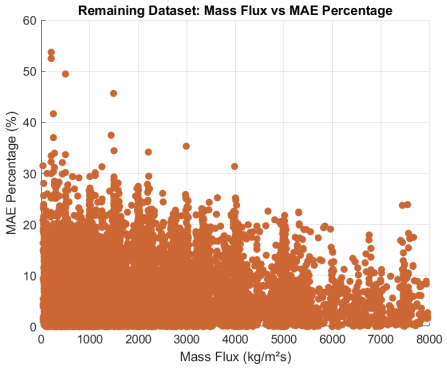


(d) MAE vs Pressure

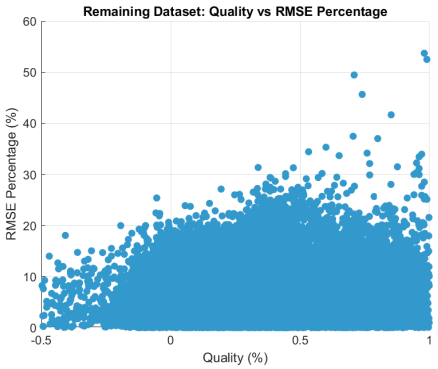
Figure C.4: RMSE and MAE percentage error distribution for neural network with training and Validation set (Part 1)



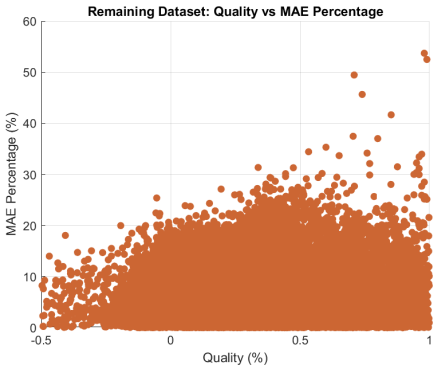
(e) RMSE vs Mass flux



(f) MAE vs Mass flux



(g) RMSE vs Quality

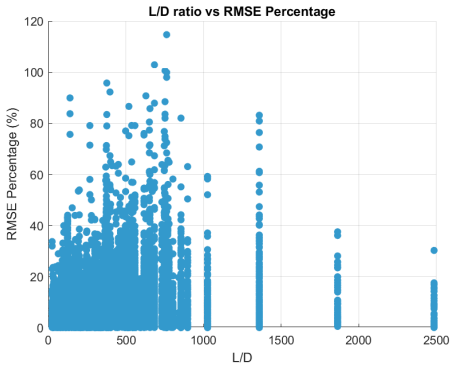


(h) MAE vs Quality

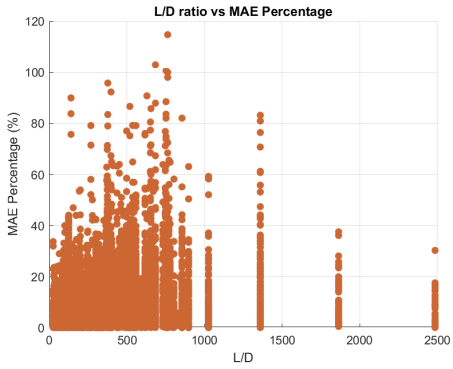
Figure C.4: RMSE and MAE percentage error distribution for neural network with training and Validation set (Part 2)

Appendix D

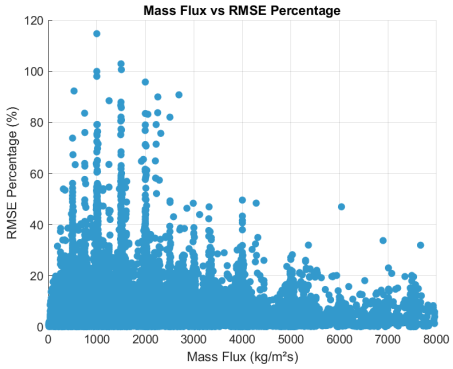
RMSE e MAE for the log-net model



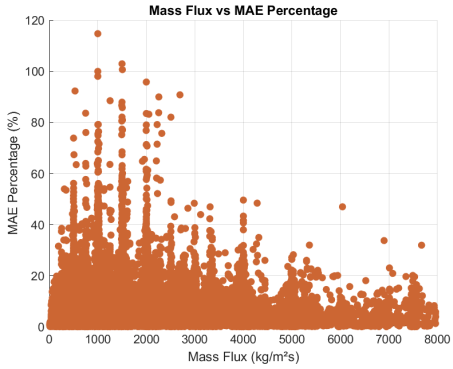
(a) RMSE vs L/D ratio



(b) MAE vs L/D ratio

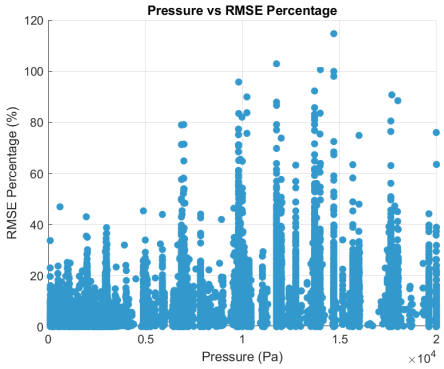


(c) RMSE vs Mass flux

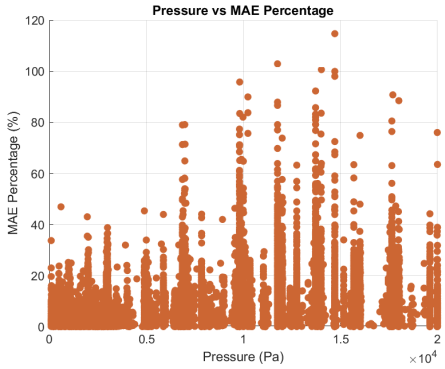


(d) MAE vs Mass flux

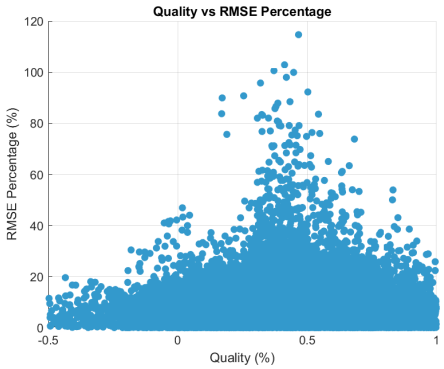
Figure D.1: RMSE and MAE percentage error distribution for log-net with training, Validation and Test set (Part 1)



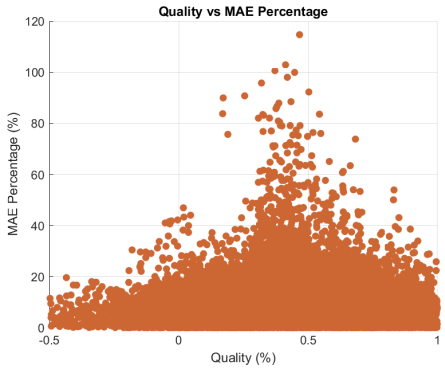
(e) RMSE vs Pressure



(f) MAE vs Pressure



(g) RMSE vs Quality



(h) MAE vs Quality

Figure D.1: RMSE and MAE percentage error distribution for log-net with training, Validation and Test set (Part 2)