2019•2020

Faculteit Industriële ingenieurswetenschappen

master in de industriële wetenschappen: chemie

Masterthesis

Energy minimization of distillation columns by neural networks

PROMOTOR :

Prof. dr. ir. Mumin Enis LEBLEBICI

PROMOTOR :

ir. Min WU

Jordy Ottenburgs

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: chemie

Gezamenlijke opleiding UHasselt en KU Leuven

UHASSELT KU LEUVEN

UHASSELT KU LEUVEN

2019•2020

Faculteit Industriële ingenieurswetenschappen

master in de industriële wetenschappen: chemie

Masterthesis

Energy minimization of distillation columns by neural networks

PROMOTOR :

Prof. dr. ir. Mumin Enis LEBLEBICI

PROMOTOR :

ir. Min WU

Jordy Ottenburgs

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: chemie

UHASSELT KU LEUVEN

Deze masterproef werd geschreven tijdens de COVID-19 crisis in 2020. Deze wereldwijde gezondheidscrisis heeft mogelijk een impact gehad op de opdracht, de onderzoekshandelingen en de onderzoeksresultaten.

PREFACE

As a master's student at the engineering faculty of UHasselt in cooperation with KU Leuven, I was granted the opportunity to perform research at 'Centrum voor Industriële Procestechnologie' (CIPT) in prospect of my master's dissertation. CIPT, also previously known as LAB4U, performs research with the intention to develop new technologies which can be applied to and for industrial processes. This includes a range of subjects like the improvement of the production of pharmaceuticals and reactors running on solar power [1].

The research took place from the 28th of February until the 11th of May and focuses on the application of neural network structures in the prediction of distillation column settings and specifications.

I would like to take this opportunity to thank a few people who contributed to this dissertation.

- First, I would like to thank my internal promotor, ir. Min Wu for guiding me through this research project and helping me overcome the problems that emerged during coding and the design of logical structures.
- Prof. dr. ir. Mumin Enis Leblebici, for his continuous support and guidance during my research project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

AI	Artificial intelligence
ANN	Artificial neural network
B	Bottom flow rate
BO	Bayesian optimization
BR	Bayesian regularization
BTX	Distillation of benzene, toluene and three xylene isomers
CIPT	Centrum voor industriële procestechnologie
CPU	Central processing unit
D	Distillate flow rate
DFM	Driving force method
EPR	Electronic paramagnetic resonance
GPU	Graphical processing unit
HPLC	High performance liquid chromatography
L/D	Reflux ratio
LM	Levenberg-Marquardt
Logsig	Logarithmic sigmoid transfer function
MSE	Mean square error
Q_C	Condenser duty
Q_R	Reboiler duty
RMSE	Root mean square error
SMOGN	Synthetic Minority Oversampling Technique for Regression in combination with Gaussian Noise
Tansig	Hyperbolic tangent transfer function
X_D	Distillate composition
X_B	Bottom stream composition

ABSTRACT

Distillation is one of the most commonly used operation units in the chemical industry. Minimizing energy consumption is an important aspect of designing distillation columns. This energy minimization leads to a reduction in CO_2 emission. Designing distillation columns with the goal of energy minimization depends on graphical interpretations or requires the use of specialized software. When a lot of these simulations need to be performed, the time needed to perform all simulations can accumulate. This problem can be solved by applying an artificial neural network (ANN) solution. The goal of this research is to create an ANN that is both accurate and efficient in predicting the optimal settings of a distillation column and feed conditions.

To comply to this goal, first an ANN solution is created and trained using data collected from performing simulations in Aspen. Additionally, the ANN also needs to be optimized to ensure a good performance. The optimization procedure is performed by altering the hyperparameters of the ANN such as the size of the network and the training algorithm. After finding the optimal network configuration, results are obtained by performing simulations using the ANN.

The network is able to predict the required energy with a high accuracy, confirmed by performing an accuracy check. To ensure an accurate prediction of the optimum conditions and settings, the neural network needs to be trained using a balanced data set and the reflux ratio needs to be added to the output features of the network.

ABSTRACT IN DUTCH

Destillatie is een van de meest gebruikte systemen in de chemische industrie. Een belangrijk aspect bij het ontwerpen van destillatiekolommen is het optimaliseren van energieverbruik. Het minimaliseren van de energie leidt tot een verminderde CO_2 uitstoot. Het ontwerpen van destillatiekolommen met oog hierop gebeurt vaak via gespecialiseerde softwarepakketten of wordt uitgevoerd door grafische interpretaties en berekeningen. Wanneer er een groot aantal simulaties worden uitgevoerd met deze software kan de tijd die nodig is om al deze simulaties uit te voeren zeer hoog oplopen. Dit probleem is niet aanwezig indien men hiervoor gebruik zou maken van neurale netwerken. Het doel van dit onderzoek is om een neuraal netwerk te construeren dat zowel nauwkeurig als efficiënt is in het voorspellen van de optimale instellingen van een destillatiekolom.

Om aan dit doel te voldoen wordt er een grote hoeveelheid data verzameld in Aspen om het netwerk te trainen. Hiernaast moet ook de configuratie van dit netwerk worden geoptimaliseerd. Dit gebeurt door verschillende *hyperparameters* aan te passen zoals de grootte van het netwerk en het trainalgoritme. Na het aanpassen van deze parameters worden resultaten verkregen door het uitvoeren van simulaties.

Het netwerk blijkt de nodige energie met een zeer hoge nauwkeurigheid te kunnen voorspellen. Om een correcte voorspelling te kunnen doen moet het netwerk getraind worden met een gebalanceerde dataset. Daarnaast moet de refluxverhouding toegevoegd worden aan de *output features* van het netwerk.

CHAPTER 1: INTRODUCTION

1. CONTEXT

'Centrum voor industriële procestechnnologie' (CIPT), also previously known as LAB4U, performs research with the intention to develop new technologies which can be applied to and for industrial processes. These technologies include a range of subjects like the production of pharmaceuticals and reactors running on solar power [1]. One subject in particular that is researched is the use of artificial intelligence in chemical industry applications. In the latest decade there has been an increase in the use of artificial neural networks (ANN) for multiple applications, an evolution that has been caused by major improvements in the development of the central processing unit (CPU), the graphical processing unit (GPU) and other computational components. The application of neural networks has already proven to be useful in chemical engineering applications. One application in particular that will be studied in this dissertation is the utilization of artificial neural networks in the design and modification of a distillation column.

Distillation is one of the most commonly used unit operations in chemical industry. The main goal of distillation is to purify product streams using a difference in boiling points. Accordingly, the product streams that enter the distillation column need to be heated and energy has to be supplied to the tower. During recent years, a lot of attention has been payed towards the sustainability of the industry and the use of renewable energy. One important aspect that needs to be considered to be able to comply with the emission standards is energy minimization. Reducing energy use leads to a reduction in CO_2 emission [2]. Thus, it is important that next to the required specifications of the output product, the column uses as little energy as possible.

2. PROBLEM DEFINITION

Distillation cases generally consist of two main categories: simulation and design cases. The first case study consists of a design problem. Given a fixed feed composition and a fixed number of column stages, what is the optimal design of the column concerning the reflux ratio, distillate composition and the feed stage? An attempt is made to shape a distillation column to a product stream with a specified composition, temperature and pressure. The column will be designed with the objective of energy minimization. The second case study will serve as an extension to the first case study and will in addition focus on optimizing the total number of stages, feed temperature given a specific feed composition.

Chemical processes are often described as systems of complex mathematical equations. These equations need to account for the multiple interactions that arise during the process, including the interactions between the components and the multiple phases [3]. The design of a distillation column is usually performed either through the use of specialized software or

depends on graphical interpretations. Performing a lot of simulations using specialized software may cost a lot of time. Each simulation takes a few seconds, while a neural network can perform tens of thousands of simulations in a fraction of a second. Additionally, tests to simulate and optimize settings are usually performed in a lab scale environment and are susceptible to errors made due to a scaling-up factor [2]. That is where the neural network vows to be an efficient tool. Artificial neural networks (ANN) are generally known for their ability to relatively fast detect non-linear relations between input features and the respective output features. The ANN will be used to provide a fast and simple interpretation to approximate the new column settings/specifications.

In this dissertation, an attempt will be made to apply ANN structures to the design of distillation columns.

3. RESEARCH OBJECTIVE

In order to be able to determine if the constructed ANN solution provides an efficient solution, it is useful to compare it to how well other methods perform in predicting the optimal column design. That is why a research article is considered, where a new method is used to optimize the design with the intention of energy minimization. The ANN prediction will afterwards be compared to the prediction of this method to see if it is efficient and if it is able to provide a solution that requires less energy.

The main objective of this research study is to find an efficient and accurate ANN solution that can be applied to a design problem. An additional goal is to get a better solution (less energy required) from the ANN than in the case of using the conventional method from the research article. To be able to comply to this objective, the ANN needs to be optimized to return a minimal error on the predicted settings/specifications. This will be accomplished by tuning the hyperparameters of the network.

4. LAYOUT

The second chapter consists of a theoretical background on neural networks and the analysis of the degrees of freedom. Additionally, the materials and methods will be discussed. The third chapter contains the results of the creation of the first neural network which is used to predict the optimum feed stage and reflux ratio for a fixed feed composition and a fixed number of column stages. In the fourth chapter, this model is expanded to also provide the optimum feed temperature and number of stages for different feed compositions. The final chapter consists of a conclusion and an outlook to the future work that needs to be done to further optimize the prediction by the neural network models.

CHAPTER 2: THEORETICAL BACKGROUND

1. ARTIFICIAL INTELLIGENCE

The last few decades, a lot of progress is being made towards the development of new techniques that will be used to further simplify everything surrounding us. A lot of these techniques will be used in engineering applications to optimize processes and reduce costs.

One of the most prominent techniques on the rise is the use of artificial intelligence (AI). This terminology might sound futuristic, but existing applications have already proven otherwise. A lot of applications of artificial intelligence are already present in our everyday life.

1.1. Artificial intelligence, machine learning and deep learning

The first big moment of artificial intelligence originates from the 1950s, when Alan Turing developed the Turing Test to test a machines intelligence against a human one. Over a short period of time, simple programs, like artificial checkers players were developed. Little did they know they would spark an interest leading to the revolution of today's automation.

Artificial intelligence has become known as the addition of human intelligence to machines. The thought behind it is to learn machines to mimic the thinking process of a human brain. Later on, in the 1980s, the use of machine learning started to flourish. Machine learning is based on using input data to 'train' artificial intelligence in order to make future decisions based on new data.

In the last decade, deep learning has found its way into practical use cases. Deep Learning is a technique based on the functionality of the neurons in our brain. The created neural networks will allow for the model to recognise patterns in data and draw conclusions from them. The theory behind deep learning will be explained in 1.4 [4], [5].

1.2. Supervised and unsupervised learning

Machine learning can be operated using two different techniques: supervised and unsupervised learning.

If a supervised machine learning algorithm is used, the data inserted into the algorithm is labelled. This includes algorithms like linear regression and object recognition in photo's. Two examples are supplied to elaborate on the operation of the algorithm [4], [5].

- The first example is a classification problem. A possible scenario of a classification problem is the evaluation of the malignancy of tumors. Based on the model developed by the input data, new data will be classified as 'malignant' or 'not malignant'.
- Another type of problem is regression. Imagine wanting to predict the price of real estate. Prices of recently sold houses in a neighborhood are available along with properties like habitable area, number of bedrooms and number of bathrooms. This data will be used to train a model which in the end will be used to estimate the real estate price of a house.

Unsupervised learning is used in cases where data is supplied unlabeled. A case where the application of unsupervised learning is convenient is the use of e-mail spam filtering. A clustering technique will be used to group similar samples together based on their properties and form classes: spam and non-spam e-mails.

1.3. Linear regression and cost function

In the previous paragraph, the difference between supervised and unsupervised learning algorithms was explained. In the case of supervised learning regression problems, the model uses linear regression to predict a value. In order to achieve a good correlation between the expected and calculated value, a cost function is used to assess the model.

A possible cost function is represented by the mean square (MSE) error equation:

$$J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2 \cdot m} \cdot \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 \quad (1)$$

With:

- m = number of data entries
- $h_\theta(x^{(i)}) = \theta_0 + \theta_1 \cdot x_1^{(i)} + \theta_2 \cdot x_2^{(i)} + \cdots + \theta_n \cdot x_3^{(i)}$ = predicted value
- $y^{(i)}$ = actual value

A lower cost indicates a better model. The cost function needs to be minimized by iteratively altering the coefficients or weights, which is called backpropagation. There are a number of ways to do this. The first method consists of using gradient descent to find the optimum coefficients.

The iterative formula for the coefficient is represented by:

$$\theta_j := \theta_j - \alpha \cdot \frac{1}{m} \cdot \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j{}^{(i)} \quad (2)$$

With:

- $x_0{}^{(i)} = 1$
- α = learning rate

The coefficient will be recalculated until the cost remains the same or increases. The cost can be approached as a three-dimensional figure of which the global minimum needs to be achieved by altering the position on the graph. Such graphical interpretation is represented in Figure 1. When applying this method, careful consideration has to payed to the learning rate. This factor influences the performance of the algorithm. A learning rate too small or large might cause the algorithm to get stuck at a local minimum, causing the algorithm to never reach an optimal solution.

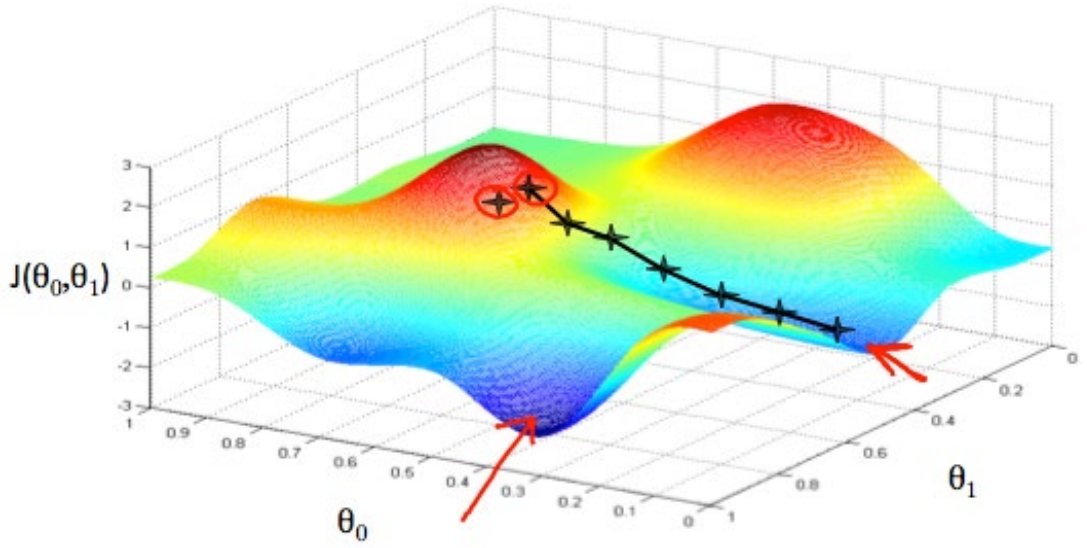


Figure 1: Graphical interpretation of gradient descent [5]

Another method that can be used to determine the optimal coefficients is the use of the normal equation, a method that solves θ analytically using the following vector expression.

$$\theta = (X^T \cdot X)^{-1} \cdot X^T \cdot y \tag{3}$$

In case of a large number of features (>10^6), gradient descent is preferred. When working with a smaller set of features, the use of the normal equation is suggested as it will converge faster [5], [6].

1.4. Artificial neural networks and deep learning

Artificial neural networks are algorithms based on the functioning of a human brain, more specifically on the functionality of neurons. Our brain consists of billions of neurons, which are all interconnected with each other. Each neuron is connected to approximately another 1000 neurons [7]. An artificial neural network will attempt to replicate this structure using nodes (neurons) which are connected through layers. The goal of this artificial neural network is to find a connection between the input and the output of the training data.

The node will function according to Figure 2.

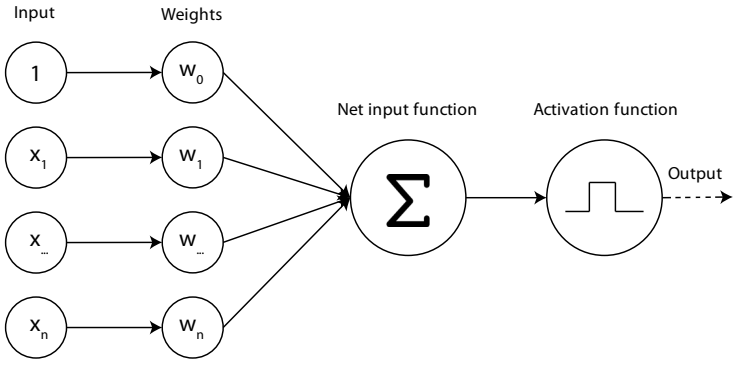


Figure 2: Diagram of node functionality

The input data will be entered through the input nodes. A weight will be assigned to the inputs depending on the significance of each input factor, a more significant factor is assigned a larger weight coefficient. The first input factor is considered the bias. As can be seen in Figure 2, the input is first multiplied by the weight coefficient. Afterwards, all factors will be enumerated leading to one final sum. This sum will be passed to the activation function, which will decide if the node is activated or not.

Activation or transfer functions exist in several forms and are picked depending on the kind of data they need to transfer. The following activation functions mentioned in Figure 3 are the most frequently used ones and can be chosen for use in a hidden layer.

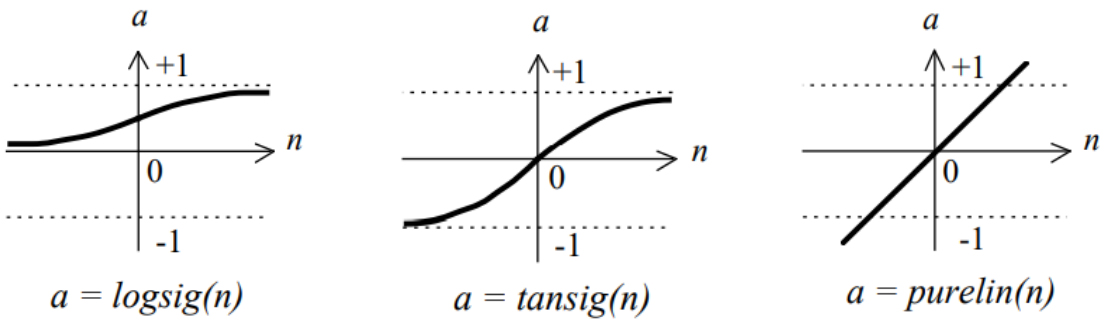


Figure 3: Commonly used activation functions [8, p.2-5]

In case of the use of linear regression, the hidden layer that passes data to the output layer will make use of the linear activation function.

Figure 4 shows an artificial neural network, which consists of a number of layers: the input layer, the output layer and a variety of hidden layers - also referred to as the depth of the structure. The output of one layer will be used as the input for the consequent layer. Deep learning networks are networks with a depth larger than three layers. Multiple hidden layers lead to the ability to process more complex input features [9].

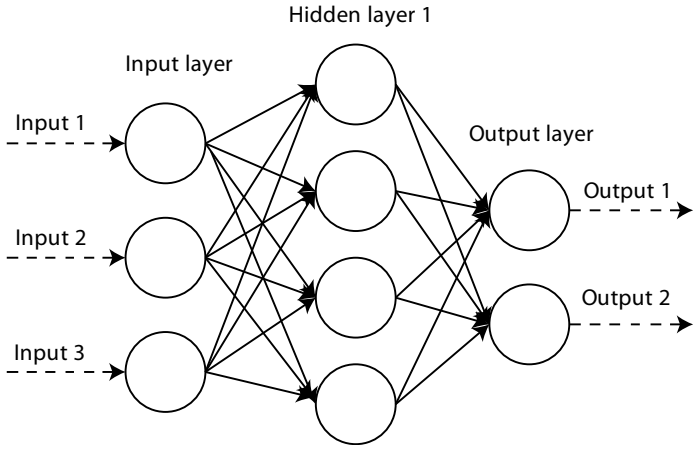


Figure 4: Artificial neural network structure

An algorithm often used to train deep learning networks is the method of backpropagation. The algorithm will iteratively try to fit the calculated data to the labelled data using loss minimalization.

An important aspect of constructing artificial neural networks is to choose the appropriate amount of nodes and layers, which is called hyperparameter optimization. Choosing too much hidden layers and nodes can cause overfitting of the data while not choosing enough layers might lead to underfitting [10]. Other parameters included in the domain of hyperparameters are network parameters like the type of transfer function and the training algorithm used to train the network with the training data.

1.5. General applications of deep learning

Artificial intelligence and deep learning are already applied in a broad range of domains. A lot of research is being done towards application in more complex situations. Several applications focus on making our lives easier while AI and deep learning also feature in a lot of industry applications.

A subject that has frequently appeared in the news lately, is the use of autonomous vehicles. Several multinational manufacturers are in progress of research towards employment on public roads, while some of them already have employed autonomous vehicles on the road. Furthermore, research being done in the medical sector generally focuses on the use of artificial

intelligence to make diagnoses and to predict what treatment methods are more likely to succeed on a patient based on historical data.

An example of an artificial intelligence application in the economical domain is the use of personalized advertisements. By analyzing web browsing data, a model will predict what kind of products potential customers are interested in.

1.6. Applications of deep learning in chemical engineering

Intensive research is being done towards the implementation of deep learning in several domains of chemical engineering. The most important reasons are to optimize processes, reduce production losses and thus reduce costs and maximize profits.

ORGANIC CHEMISTRY

- Coley et al. developed a supervised deep learning approach to predict the end products of organic reactions given the reactants and solvents. The goal of this research is to have a reaction evaluation, to be able to predict the formation of by-products during a chemical reaction and possibly calculate the reaction yield in given circumstances.

  The constructed model consists of four main steps. The first step is to input the reactants and solvents. Next, the reaction sites of a molecule are determined using the detection of functional groups within the molecule. The third step consists of forming possible end-products using the chosen reactive sites. Afterwards, products that can be produced are ranked according to their likelihood to form. In the final step the reaction products are displayed in the output. The products predicted by the model were 85% accurate and all calculated within a time of 100 ms [7], [8] .

- One year earlier, Gao et al. constructed an ANN to forecast optimal conditions for organic reactions. The model is able to predict up to one catalyst, two solvents, two reagents and the temperature starting from the end-product [13].

COMPUTATIONAL CHEMISTRY

- Moghadassi et al. applied deep learning to the prediction of liquid viscosity. Because of product plant design, the viscosity of a liquid is very important to know during a production process as it influence the choice for pipeline diameters. Data on viscosity is not always available for specific conditions and might only be able to be determined empirically. In many cases it is convenient to use a neural network model to predict the viscosity. The predictions of the applied model resulted in an average error of 2.53% in comparison to data available in literature [14].

- Furthermore, attempts have been made to apply ANN structures to determine lipophilicity of chemicals [15], ionic activity coefficients and solubilities of electrolytes [16].

BIOCHEMISTRY

Artificial neural networks have also been a great tool in the analysis and understanding of the functionality of proteins.

- The use of ANNs has been employed in predicting the stability of RNA/DNA hybrid duplexes, which are used for the design of chemotherapeutic drugs. A parameter used to assess the stability of the duplex is the melting temperature. The ANN predicted a melting temperature for a given duplex with a maximum standard error of 3.506 K [17].
- Dombi et al. reported the use of ANNs for the analysis of protein transmembrane helical regions. A troublesome part of protein analysis is the characterization of the tertiary structure of the protein. To gain a better understanding of protein structures, a neural network structure was applied to determine the secondary structure of proteins. Although secondary structures are more properly documented, the applied ANN can still turn out to be of use [18].
- Empirical methods to determine the chemical shift point of a protein often do not take into account the 3D-structure of a protein and interactions with other molecules. Through the use of artificial neural networks, Meiler at al. attempted to include these interactions in the prediction of the chemical shift of proteins [19].
- Another example of an ANN application in biochemical engineering is the prediction of peptide liquid chromatography elution times in an effort to improve the confidence level of peptide identification [20].

FOOD INDUSTRY

- In the tea industry, characterization of different tea kinds is desirable to classify health effects. Characterization is based on chemical composition and antioxidant activity. Antioxidant activity levels are usually measured using either high-performance liquid chromatography (HPLC) or electronic paramagnetic resonance (EPR). As this methods are rather expensive and time consuming, using a model structure to predict the antioxidant activity is preferred. An ANN structure was used to predict the antioxidant activity in the tea and was trained using a backpropagation algorithm. Relative errors were reportedly less than 0.4% [21].

PROCESS CONTROL

- In 1991, Himmelblau et al. developed an artificial neural network structure to detect faults in production processes. An example where this model is of use is the diagnosis of defects in a heat exchanger. The model was trained using simulated data (temperatures and heat transfer coefficient) with noise applied to the data and was able to correctly detect when defects were applied (0.02 of original value) [22]. In 1993, Suewatanaku et al. expanded the model by including flow rates, pressure drops in the tube and shell size to study fouling [23].
- Barton et al. used ANNs to predict polymer quality in industrial reaction units. Controlling reactor settings was difficult because of the large time between the changing of the controls and the measurement of samples as a result of the change in controls. The arrival of measurements was too late to accurately control the reactor settings. The model turned out to be accurate in the prediction of polymer quality over the entire reactor operating region [23].
- MacMurray et al. reported the use of ANNs for model predictive control (MPC). The goal of this model is to predict behavior of a chemical process, in order for process parameters to be manipulated at the right time to achieve the desired product properties [23].

2. DISTILLATION

Distillation is a chemical unit operation which is supported by a difference in boiling temperature between multiple components. This type of operation is used to separate components and purify product streams. The component with the lowest boiling point is the most volatile one, leading to a faster displacement to the vapor phase in opposition to the least volatile component.

When working with design specifications and settings, attention has to be payed towards the number of variables specified for a specific case. Only a limited amount of variables can be altered. The analysis of the degree of freedom makes sure the column will not be over specified. It assumes a constant molar overflow and a constant pressure within the column. For a normal distillation column the degree of freedom is calculated accordingly.

Table 1 and Table 2 display the accounted degrees of freedom for each specified distillation input feature and the feed respectively. For example, the degrees of freedom linked to the feed composition equals C-1 with C being the number of components in the feed stream. The summation of all degrees of freedom results in a total of C+6 degrees of freedom. In most cases however, the feed parameters are set fixed.

Table 1: Available degrees of freedom of distillation column [24]

Variables	Degree of freedom
Feed flow F	1
Feed composition	C-1
Feed temperature	1
Recovery component A in distillate	1
Recovery component B in bottom	1
L/D or V/B or heating duty	1
Saturated liquid reflux or T_{reflux}	1
Optimum feed plate location	1
	C+6

Table 2: Degrees of freedom set by specified feed [24]

Variables	Degree of freedom
Feed composition	C-1
Feed flow	1
Feed temperature	1
Feed pressure	1
	C+2

The corresponding degrees of freedom of the feed equals C+2. According to [24], an alternative to feed pressure is the reflux temperature or enthalpy.

If the degrees of freedom of the feed are subtracted from the total degree of freedom of the distillation column, a remaining number of four variables remain to be specified. In a distillation problem case, two situations might occur. One situation where it is necessary to design a new column for a process (design problem) or a separation process of which some calculations need to be made (simulation problem). Each problem requires a different approach [25] .

The cases in Table 3 describe some examples that can be applied to design a new column starting from a fixed feed state.

Table 3: Specifications and calculated variables for distillation design problems [24]

	Specified variables		Calculated variables
A	Mole fraction light component in distillate (x_D)	A	Distillate flow rate and Bottom flow rate
	Mole fraction heavy component in bottom (x_B)		Heating and cooling duty (Q_R and Q_c)
	Reflux ratio		Number of stages and optimum feed plate
	Use of optimum feed plate		Column diameter
B	Fractional recovery of components in distillate	B	x_B, x_D, D, B
	Fractional recovery of components in bottom		Q_R and Q_C
	Reflux ratio		Number of stages and optimum feed plate
	Optimum feed plate		Column diameter
C	Distillate flow rate or Bottom flow rate	C	B or D
	x_D or x_B		x_B or x_D
	Reflux ratio, L/D		Number of stages and optimum feed plate
	Optimum feed plate		Column diameter
D	x_D	D	D, B
	x_B		Q_R, Q_C
	Reflux ratio, L/D		Number of stages and optimum feed plate
	Optimum feed plate		Column diameter

The specifications in Table 4 can be utilized in case of a simulation problem. Column specifications like the feed stage and number of stages are already specified, only the settings of the column can be changed.

Table 4: Specifications and calculated variables for distillation simulation problems [24]

	Specified variables		Calculated variables
A	Number of plates	A	Reflux ratio
	Feed plate		B and D
	x_D, x_B		Q_c and Q_R
	Column diameter (constraint)		Check $V < V_{max}$
B	Number of plates	B	x_B (or x_D)
	Feed plate		B and D
	L/D, x_D (or x_B)		Q_c and Q_R
	Column diameter (constraint)		Check $V < V_{max}$
C	Number of plates	C	Reflux ratio
	Feed plate		B and D
	x_D (or x_B)		Q_c and Q_R
	Column diameter		X_B or x_D
D	Number of plates	D	Reflux ratio
	Feed plate		B and D
	Q_R, x_D (or x_B)		Q_c , x_B (or x_D)
	Column diameter (constraint)		Check $V < V_{max}$

3. MATERIALS AND METHOD

In this section, the materials and methods used to conduct the experiments will be discussed.

3.1. Materials

The following software packages are used to perform experiments for the research in this dissertation.

- MATLAB® R2019b (9.7.0.1190202)
- MATLAB® Deep Learning Toolbox version 13.0
- MATLAB® Statistics and Machine Learning Toolbox version 11.5
- Aspen Plus® version 10.0

3.2. Creating an artificial neural network structure

The construction of a proper artificial neural network structure consists of a number of steps. The first step consists of simulating a considerable amount of data entries through the use of Aspen Plus®. During the next step, the data is pre-processed to be able to start the construction of the neural network. Input and output features for the neural network are selected and the model is constructed. A large amount of possibilities exist to shape the structure of a neural network., common options for the hyperparameters are explored. These hyperparameters are varied in order to obtain a neural network with a good accuracy. Finally, the neural network is used to predict the parameters that are needed to complete each case study. The proposed methodology is shown in Figure 5.

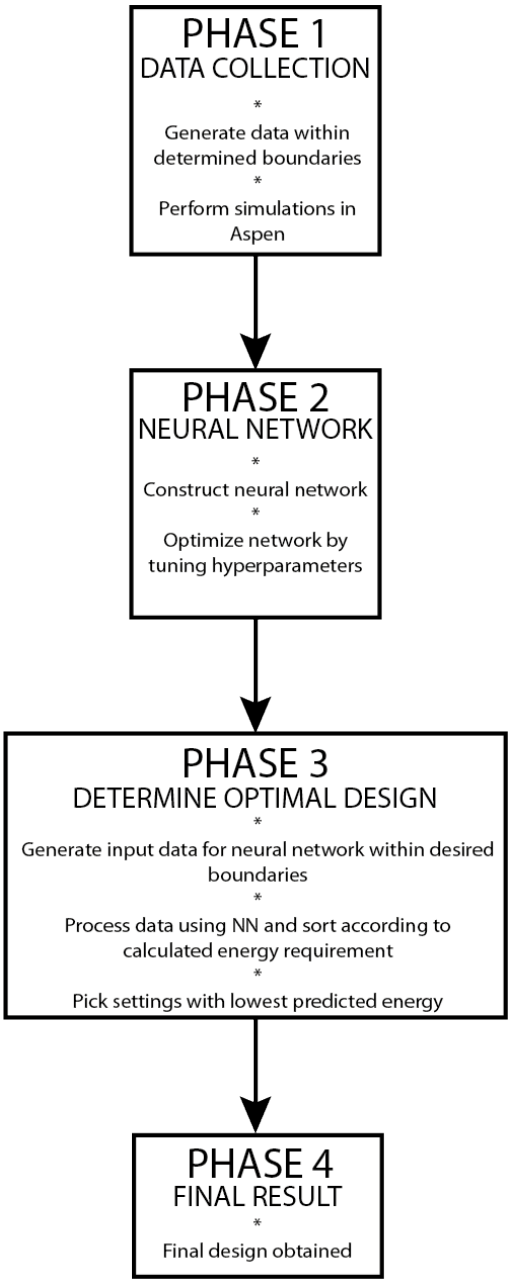


Figure 5: Proposed methodology

3.2.1. Simulation

As a first step all the necessary data is collected in Aspen Plus. For the neural network to be able to predict the desired features accurately, a lot of data needs to be collected. A choice for a sample size that is too small might result in the problem of underfitting. Underfitting may cause the network to provide inaccurate predictions.

3.2.2. Extracting data

Manually performing simulations and recording the data would be rather time consuming. To address this issue, MATLAB is used in combination with Aspen Plus. Creating a COM interface within MATLAB already proved to be an efficient way to control Aspen Plus from MATLAB [26].

Random data is generated using MATLAB within predetermined boundaries for the input features. Accordingly, this data is inserted into Aspen using the COM interface and simulations are performed. After each simulation, the data of interest is saved into a matrix and saved as a .csv file to be processed later on. The code to extract the data was constructed using information from Satola et al. [26] and Abril et al. [27].

3.2.3. Pre-processing data

An important step in the process is to pre-process data as this will mainly determine the speed at which the network will learn. This is caused by the structure of the activation function. An activation function often used in the hidden layers of a neural network is the sigmoid function, which does not function well when using net inputs with a value greater than three. Gradients will be very small and network training will be slow [8]. For this purpose the data is 'feature scaled' using the following standardization equation.

$$x_n^{(i)} = \frac{x^{(i)} - \mu}{\sigma} \tag{4}$$

With:

- $x_n^{(i)}$ = new value
- $x^{(i)}$ = old value
- μ = mean of all feature entries
- σ = standard deviation of all feature entries

Alternatively, methods like rescaling (min-max normalization) and mean normalization can also be used to feature as a function for feature scaling.

3.2.4. Creating a neural network structure

After the pre-processing of the data, a neural network structure is created using the 'Deep learning toolbox' of MATLAB®. This toolbox provides a graphical user interface that can be used to create a shallow neural network with a single layer. The input and output features are chosen depending on the goal of each case study.

A considerable advantage of the Deep learning toolbox is that it is able to generate a training code. This training code can be edited to change various parameters of the neural network, which will be used to carry out the hyperparameter optimization. An example of this code is supplied with Appendix A. A few examples of parameters that can be varied are:

- number of hidden nodes,
- number of hidden layers,
- cost function,
- activation function,
- training algorithm,
- division of data.

An important note is that the algorithm must return the same MSE for every training iteration at the same settings. At default settings the initial weights are randomized each iteration and a random data division is utilized. To prevent this from happening the following measures need to be taken.

To prevent from starting at different weights each iteration, the random seeds need to be initialized at the start of each training loop. A random seed specifies the starting point of the generation of a random number [28]. To compare different configurations of the network, the seeds are initialized at zero for every loop to start with the same initial weights.

Another measure that needs to be taken is the fixed division of data. As previously mentioned, the data set is divided into a training set, a test set and a validation set with a default ratio of 70/15/15. The validation set is used to make training less time intensive and correct for overfitting. The validation passes are defined, which will lead to early stopping. This indicates that training will stop if the MSE on the validation test set fails to decrease after a defined number of epochs. Each iteration will need to use the same division of data sets. This is taken care of by using the division by indices function in the Deep learning toolbox. Alternatively, the default division function can be used that divides the data set randomly. By initializing the seed at every training iteration, a fixed division of the data is ensured.

3.2.5. Hyperparameter optimization

Next, to make sure the created neural network is efficient and performs well for all the output features, an optimization step has to be performed. The hyperparameters that must be optimized include the number of hidden layers, hidden layer size, the activation function and the training algorithm. While these are all varied, the performance and regression of each

output feature is recorded, together with the total performance. Based on these parameters the optimal network configuration is determined.

To optimize the hidden layer size and the number of hidden layers, a training loop is performed. For each iteration the performance for each feature together with the performance curve and regression for each feature is recorded. An attempt is made to find a model with a minimal MSE and a maximal regression fit without over or underfitting the data. As training algorithms are very computationally intensive with high network sizes, the layer size is limited to 25 nodes while the number of hidden layers is limited to two. A Bayesian Optimization algorithm will be used to check larger configurations for a better performance. This is an algorithm that is often used to evaluate configurations that take a long time to assess.

Another parameter that is varied is the training algorithm. The following backpropagation algorithms are the most commonly used:

- Levenberg-Marquardt: trainlm,
- Bayesian regularization: trainbr,
- Scaled conjugate gradient backpropagation: trainscg.

The Levenberg-Marquardt (LM) training algorithm is the default training algorithm for regression problems in the 'Deep learning toolbox'. This algorithm supports the division of the data set into a test, training and validation data set. The validation data set is used to prevent the model from overfitting. It stops the algorithm from training when the validation performance fails to improve or when a maximum amount of training loops or 'epochs' have been performed. The default validation checks that are performed are set to six while the maximum of epochs the algorithm performs is limited to 1,000. The main goal of this algorithm is to minimize the mean square error.

The second algorithm is the Bayesian regularization (BR) backpropagation training algorithm. This algorithm is based on the Levenberg-Marquardt algorithm, but adds generalization. Unlike the LM training algorithm, this algorithm does not work with a validation subset. The goal of the algorithm is to minimize the weighted sum of squared errors and squared weights in an effort to prevent overfitting of the data set. As overfitting generally occurs when not enough training data is present, this method is often used for smaller data sets. As both the LM and BR algorithms are based on the mean square error, only the MSE cost function can be used in combination with these two algorithms [8], [29].

This two training algorithms are prompted in order to find the optimal neural network, next to the two transfer functions for the first hidden layer: 'tansig' and 'logsig'. In the final step of determining the optimal network configuration, consideration has to be paid to the choice of the model. A model with a good performance does not automatically indicate it is a better model. Larger sizes of neural network often lead to overfitting of the data, which indicates that the model performs well for the data in the training set but performs very poor on generalization. Generalization indicates how well a model performs for data it has never seen before.

The decision that a model is overfitting can be deducted from the training performance curve. This curve shows the development of the MSE of the training, test and validation set over each training epoch. Two examples of such training curves are displayed in Figure 6 and Figure 7. At the left, a figure is displayed where the model suffers from overfitting. When the model does not show any signs of overfitting, the figure will look like the one on the right. Additionally, the performance of the test set can be used to judge the overfitting properties of the network.

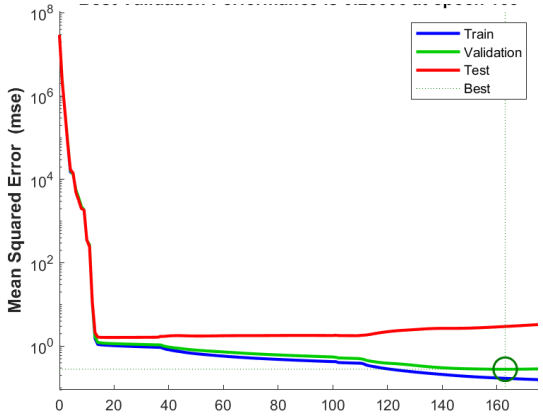


Figure 6: Performance curve overfitting

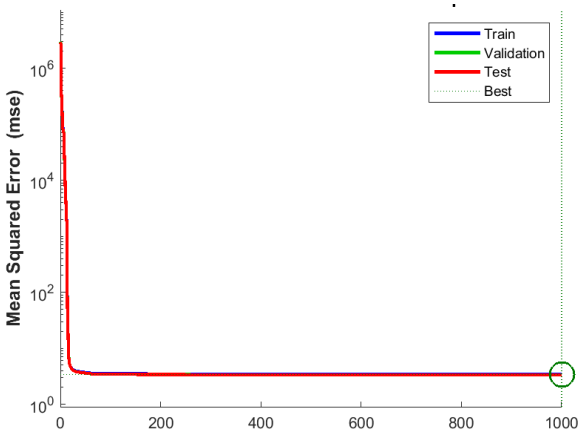


Figure 7: Performance curve no overfitting

An additional method to optimize the hyperparameters will be used later on in this dissertation and is called a Bayesian Optimization algorithm. This algorithm will perform random searches according to a defined objective function. An example of a possible objective function is the RMSE of the test set. The code is modified from the code published by [30].

3.2.6. Application of neural network

Furthermore, a considerable amount of input data is generated within the same boundaries established in the first phase with the intention to process them using the neural network. As a final step, the prediction of the neural networks are used in Aspen to check the findings of the neural network for their accuracy. This is achieved by using an adapted version of the code used for the collection of data.

CHAPTER 3: CASE STUDY ONE – CREATION OF A PREDICTION MODEL

To be able to determine if a proposed ANN solution can provide an efficient solution, it is compared to the use of a more conventional method. In 2014, Faris et al. published a paper concerning the design of energy efficient distillation column systems [2]. Towards the prediction of an energy optimum design, a method called 'driving force method' (DFM) was used and was compared against the use of a shortcut design of a distillation column. This method proved to be more energy efficient than the shortcut design and resulted in an economization of approximately nine percent. The DFM depends on graphical interpretations to determine at which settings there is a minimum of energy required to separate the components.

The paper described two use cases . Both cases were systems of two distillation columns and the DFM was applied to the final column in the system . A direct and an indirect system approach to a BTX distillation were studied– BTX refers to a distillation of benzene, toluene and three xylene isomers. In this dissertation, the direct approach will be used as the foundation of the first case study. An ANN solution will be applied to this problem and the result will be compared to the result obtained by using the DFM and shortcut design.

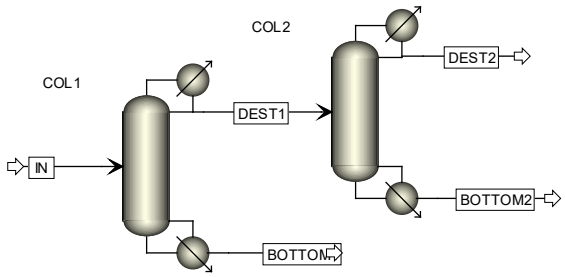


Figure 8: Indirect approach

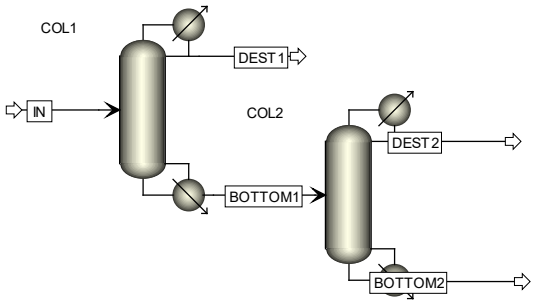


Figure 9: Direct approach

The proposed methodology of the first case study is shown in Figure 5.

1. DATA

The case study starts with a representation of the data. The data in Table 1 was derived from the research paper [2] for the feed stream, with a temperature of 75 °C and a pressure of 2 atm.

Table 5: Feed specification [2]

Streams	Feed	
Components	kmol/hr	X
benzene	50.00	0.50
toluene	30.00	0.30
p-xylene	20.00	0.20
total	100.00	1.00

The research paper also discloses results for the specifications of the shortcut design and the driving force method. But as no column pressure was specified, the column pressure was set as the same pressure as the feed – 2 atm. The results for both methods were recalculated using Aspen Plus and are displayed in Table 6.

Table 6: Results of shortcut design and driving force method

		Shortcut design	Driving force method
No. of stages, Ns		21	21
No. of feed location, NF		11	13
Reflux ratio		1.60	1.41
Composition	benzene	0.9900	0.9900
at top	toluene	0.0100	0.0100
	p-xylene	0.0000	0.0000
Composition	benzene	0.0100	0.0415
at bottom	toluene	0.5900	0.5714
	p-xylene	0.4000	0.3871
Energy condenser (kW)		1064.37	953.88
Energy reboiler (kW)		1315.35	1201.64
Total energy (kW)		2379.72	2155.52

2. CREATING LOGICAL STRUCTURES

Before moving on to the simulation of the data, a decision needed to be made for the selection of the input and output data that was used in the artificial neural network as well as the data that was used for simulations in Aspen.

2.1. Logical structure for Aspen data collection

First, the network features need to be collected using Aspen. A criterium for this step is that the simulations happen as efficient as possible. Just like in the case of the model selection for the neural network, a few models have been tested for the data collection. The models are based on a specific set of parameters. A limited amount of combinations are available for the specification of the column in Aspen. As determined in the analysis of the degrees of freedom, four column parameters can be specified. Of these four, the number of stages and feed stage are already chosen. Two parameters remain to be specified.

Aspen allows for a range of parameters to be chosen to fill the remaining two slots when using the 'Radfrac' model. Next to this parameters, the reflux ratio was also taken into account as it is used to compare the results against the one obtained by the conventional method. The reflux ratio features either on the input side or the output side of the data collection model in Aspen.

The available parameters in Aspen are: distillate to feed ratio, bottoms to feed ratio, distillate rate, boilup rate, reflux rate, reflux ratio, reboiler duty and condenser duty. The reasoning behind the choice of the final model is explained in Table 7.

Table 7: Model selection process Aspen

Model	Model structure	Result
1	Input - Feed composition - Feed conditions - Number of stages - Feed stage - Condenser duty - Reboiler duty Aspen Output - Reflux ratio - Distillate flow rate benzene - Distillate flow rate toluene - Distillate flow rate p-xylene	Both model 1 and model 2 were rejected because they needed to much iterations to converge. Acquiring 10,000 simulations would take more time.
2	Input - Feed composition - Feed conditions - Number of stages - Feed stage - Condenser duty - D:F ratio Aspen Output - Reboiler duty - Condenser duty - Distillate flow rate benzene - Distillate flow rate toluene - Distillate flow rate p-xylene	
3	Input - Feed composition - Feed conditions - Number of stages - Feed stage - Condenser duty - D:F ratio Aspen Output - Reboiler duty - Reflux ratio - Distillate flow rate benzene - Distillate flow rate toluene - Distillate flow rate p-xylene	Model 3 was accepted as almost every simulation converges.

The model that will be used for the following steps is model 3.

2.2. Logical structure for neural network

During the course of this experiment, a variety of structures for the ANN have been investigated. The selection for the appropriate model was performed experimentally and was considered a process of 'trial and error'. The reasoning behind the choice of the final model is discussed in Table 8.

Table 8: Model selection process neural network

Model	Model structure	Result
1	Input: - Reboiler duty, - Condenser duty → ANN → Output: - Distillate flow rate benzene, - Distillate flow rate toluene, - Distillate flow rate p-xylene, - Reflux ratio, - Feed stage	Both model 1 and model 2 were rejected because of their inadequate performance to predict the accurate feed stage. The correlation between the predicted and the real feed stage was not high enough. This can be observed in Figure 10.
2	Input: - Distillate flow rate benzene, - Distillate flow rate toluene, - Distillate flow rate p-xylene, - Reboiler duty, - Reflux ratio → ANN → Output: - Condenser duty, - Feed stage	
3	Input: - Distillate flow rate benzene, - Distillate flow rate toluene, - Distillate flow rate p-xylene, - Feed stage, - Reflux ratio → ANN → Output: - Condenser duty, - Reboiler duty	Model 3 did not face the issue present in model 1 and model 2 as the feed stage is now on the input side of the network. However, performance was still poor as the network predicted a positive value for the condenser duty when using the specifications of the shortcut design. This model was rejected on the basis of its poor overall performance.
4	Input: - Distillate flow rate benzene, - Distillate flow rate toluene, - Distillate flow rate p-xylene, - Feed stage, - Reboiler duty → ANN → Output: - Condenser duty, - Reflux ratio	The final model which was tested is model 4. The predictions of this model resulted in satisfactory results for the MSE and the correlation between the predicted and real values of both output features.

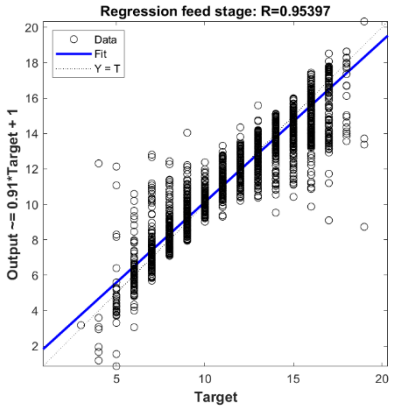


Figure 10: Correlation predicted and real feed stage - even distribution

The model that was used for the following steps is model 4.

3. ASPEN SIMULATIONS

To ease the simulation procedure, an Aspen 'template' file was created. The specifications that were used are described in the following paragraphs.

Properties – Specification

The components were entered in the 'specifications' tab of properties. In this case, the components that need to be entered are benzene, toluene and p-xylene.

Properties – Methods

In the methods section, the appropriate property model was selected. The property model was chosen according to the property model selection flow chart mentioned in the Aspen User Guide [31].

The feed mixture only contains three (nonpolar) components. Accordingly, the property model that can be selected are 'Peng Robinson', 'Redlich-Kwong-Soave' and 'Lee is 'Peng Robinson' (PENGROB). For the sake of this experiment, 'Peng Robinson' was selected as the property model.

Simulation – Flowsheet

Next, the block model was specified. For the sake of this experiment, the rigorous 'RadFrac' model was chosen. The RADFRAC column allows for the following parameters to be specified: feed stage, number of stages, condenser duty, distillate to feed ratio.

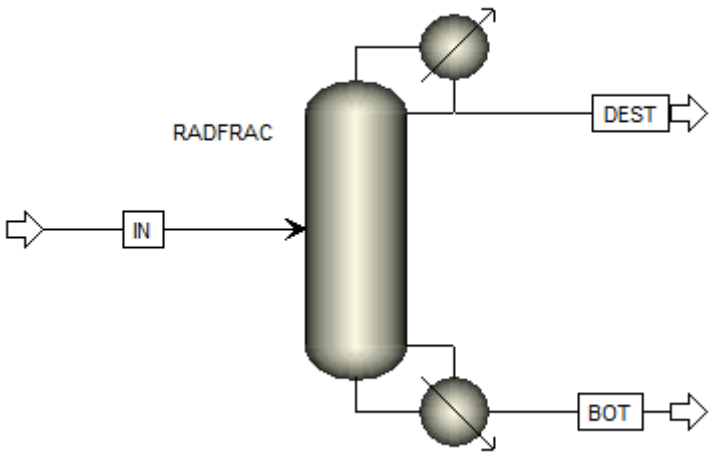


Figure 11: Flowsheet simulation

Simulation – Feed stream

The inlet stream was defined accordingly to Table 9. The corresponding degrees of freedom to the specification of the feed stream is five (C+2).

Table 9: Feed specification in Aspen Plus

Specificiation	Value
Temperature	75 °C
Pressure	2 atm
Total flow rate	100 kmol/hr
$X_{BENZENE}$	0.50
$X_{TOLUENE}$	0.30
$X_{P\text{-}XYLENE}$	0.20

Simulation – Column specifications

As a final step, the column settings were specified. The goal of this experiment is to find the optimal column settings for a minimized duty. To reach this goal several parameters were.

Four parameters remained to be specified:

- the total amount of stages was set fixed at 21,
- feed stage (varied),
- condenser duty (varied),
- distillate to feed ratio (varied).

4. GENERATING INPUT DATA FOR ASPEN

11,000 data entries were prepared both in a Gaussian distribution and an even distribution. The intention is to compare both distributions at the end of the case study according to their performance to predict the optimal settings. The boundaries were chosen as following:

- Feed stage: 3 – 19
- Condenser duty: -3000 – -500
- Distillate to feed ratio: 0.2 – 0.8

The distributions for each of the input features are displayed in Figure 12 and Figure 13.

a. Gaussian distribution

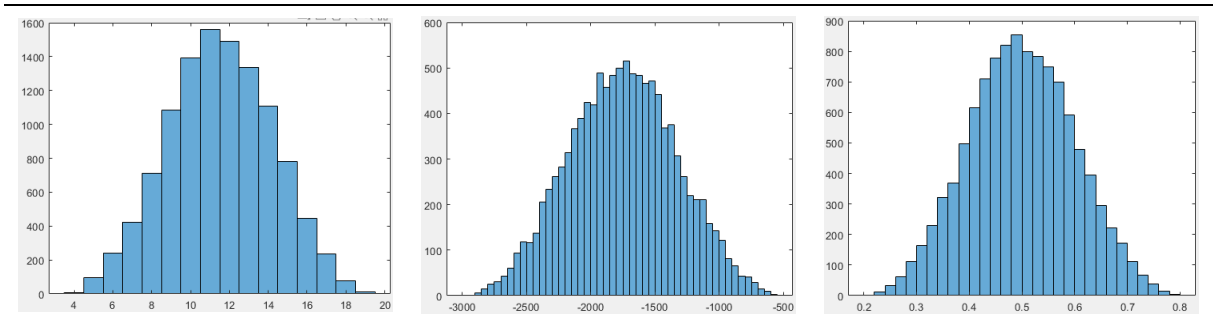


Figure 12: Gaussian distribution of aspen input parameters

b. Even distribution

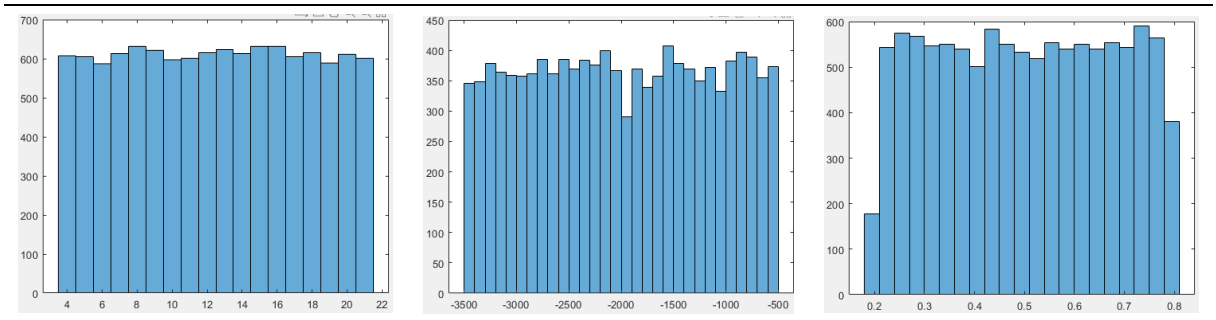


Figure 13: Even distribution of aspen input parameters

5. EXTRACTING DATA AND PRE-PROCESSING

The data entries were entered in the column specifications in Aspen. Simulations are run and non-converging data entries are removed. Afterwards the data was pre-processed. As discussed earlier, the data needed to be normalized in order for the network to be trained efficiently. For this purpose, mean normalization will be used.

6. HYPERPARAMETER OPTIMIZATION

A network configuration needs to be found with a minimal error on the predicted results. To find a network that is able to accurately predict the desired result, it needs to be optimized by tuning the hyperparameters. The hyperparameter optimization was performed in three steps.

6.1. Neural network size

An iterative training loop was performed and resulted in the following optimal network configurations for each input distribution.

Table 10: Comparison of network performance - network size - even and gaussian distribution

Distribution in Aspen	even	Gaussian
Network configuration	20x23	17x11
Total MSE	0.000552	0.000450
MSE Reflux ratio	0.000038	0.000007
R^2 Reflux ratio	1	1
MSE Reboiler duty	0.001066	0.000894
R^2 Reboiler duty	1	1
Test MSE	0.000597	0.000482

As can be seen in Figure 14 and Figure 15, no overfitting occurred for both configurations.

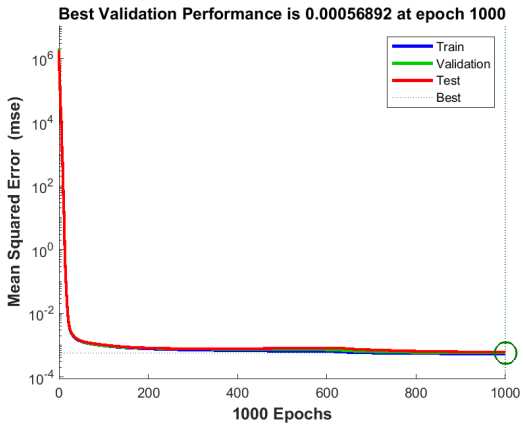


Figure 14: Performance curve - even distribution

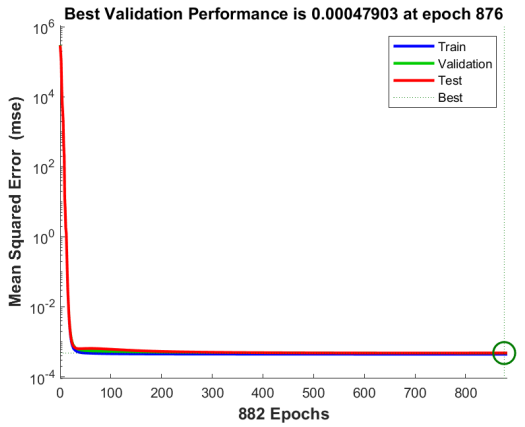


Figure 15: Performance curve - Gaussian distribution

The results were acquired by using the settings for the training displayed in Table 11.

Table 11: Training settings

Training algorithm	Levenberg-Marquardt
Activation function	Tangent sigmoid
Validation passes	6

6.2. Activation function

The second step consisted of varying the activation function. The default activation function in the 'deep learning toolbox' is 'tansig', a tangent sigmoid transfer function. The performance of the networks that use this transfer function in the first hidden layer was compared to the network that uses the logarithmic sigmoid transfer function instead.

For the even input distribution, the results are displayed in Table 12.

Table 12: Comparison of network performance - activation function - even distribution

Activation function	tansig	logsig
Network configuration	20x23	24x21
Total MSE	0.000552	0.000526
MSE Reflux ratio	0.000038	0.000009
R^2 Reflux ratio	1	1
MSE Reboiler duty	0.001066	0.001043
R^2 Reboiler duty	1	1
Test MSE	0.000597	0.000551

The network that uses the logarithmic sigmoid transfer function in the first layer proved to be more accurate in its prediction. For the network with the even distribution inputs, the 'logsig' transfer function was chosen to be further optimized.

For the Gaussian distribution, the results are displayed in Table 13.

Table 13: Comparison of network performance - activation function - Gaussian distribution

Activation function	tansig	logsig
Network configuration	17x11	13x14
Total MSE	0.000450	0.000459
MSE Reflux ratio	0.000007	0.000006
R^2 Reflux ratio	1	1
MSE Reboiler duty	0.000894	0.000913
R^2 Reboiler duty	1	1
Test MSE	0.000482	0.000471

In contrast to the results of the network with the even distribution, the total network performance did not improve by altering the transfer function for the Gaussian distribution. The network with the default transfer function 'tansig' was chosen to be further optimized.

6.3. Training algorithm

As a final step, the model was trained using the default Levenberg-Marquardt training algorithm as well as the Bayesian regularization algorithm.

For the even input distribution, the results are displayed in Table 14.

Table 14: Comparison of network performance - training algorithm - even distribution

Training algorithm	trainlm	trainbr
Network configuration	24x21	24x12
Total MSE	0.000526	0.000559
MSE Reflux ratio	0.000009	0.000053
R^2 Reflux ratio	1	1
MSE Reboiler duty	0.001043	0.000889
R^2 Reboiler duty	1	1
Test MSE	0.000551	0.000584

The total network performance and performance of the test set for the even distribution did not improve by adjusting the used training algorithm. The error on the reboiler duty decreased, but the error on the reflux ratio increased significantly.

For the Gaussian distribution, the results are displayed in Table 15.

Table 15: Comparison of network performance - training algorithm - Gaussian distribution

Training algorithm	trainlm	trainbr
Network configuration	17x11	24x18
Total MSE	0.000450	0.000449
MSE Reflux ratio	0.000007	0.000010
R^2 Reflux ratio	1	1
MSE Reboiler duty	0.000894	0.000889
R^2 Reboiler duty	1	1
Test MSE	0.000482	0.000461

Unlike in the case of the even distribution, the total network performance as well as the performance of the test set of the Gaussian distribution improved in case of using the Bayesian Regularization training algorithm. However the performance did not improve by much, the 'trainbr' training algorithm was chosen for the final design.

6.4. Final model configuration

The optimization process resulted in the optimal network configurations displayed in Table 16.

Table 16: Final network settings

Distribution	even	Gaussian
Network size	24x21	24x18
Activation function	logsig	tansig
Training algorithm	trainlm	trainbr

7. PROCESSING OF NETWORK RESULTS

In this step, the constructed neural network was put into action. As the goal is to compare our neural network solution to the solution achieved by using the driving force method, the feed composition at the top needed to be set fixed at 0.99-0.01-0 for the respective composition of benzene, toluene and p-xylene in the distillate.

10,000 data entries were generated to be entered in the neural network. These parameters include the distillate flow rate, the reboiler duty and the feed stage. The following boundaries were chosen for the data to be generated within.

a. Feed stage: 3 – 19

   The boundaries for the feed stage are defined like in the first step. The first and last two stages will not be included within these boundaries.

b. Reboiler duty: 800 -1400 kW

   The lower limit of the generated reboiler duties is fixed at 800. At low duties the separation will become more difficult. The upper limit is set a little higher than the reboiler duty used in the case of the shortcut design.

c. Total distillate flow rate: 40 – 50.5051 kmol/hr

   Because a comparison is needed between the two methods, the distillate composition remained fixed at $X_{BENZENE} = 0.99$ and $X_{TOLUENE} = 0.01$. The upper limit of the total flow rate was defined as the case where 100% of the benzene is recovered in the distillate. The lower limit was set at 40 kmol/hr as anything lower might be considered too low of a recovery.

Furthermore, the data was processed by the neural network and the settings that lead to a minimal duty were selected. The results predicted by the neural network for the optimal settings are displayed in the following paragraphs.

7.1. Results even distribution

The results for the even distribution are displayed in Table 17 together with an accuracy check obtained in Aspen. This accuracy check was performed using the optimal design predicted by the neural network as an input for a simulation in Aspen.

Table 17: Result from neural network in comparison to Aspen - Even distribution

		Neural network	Aspen
No. of stages, Ns		21	21
No. of feed location, NF		18	18
Reflux ratio		0.3896	0.3005
Composition	Benzene	0.9900	0.80
at top	Toluene	0.0100	0.17
	p-xylene	0.0000	0.03
Composition	Benzene	0.02	0.20
at bottom	Toluene	0.58	0.42
	p-xylene	0.40	0.38
Energy condenser (kW)		550.4	X
Energy reboiler (kW)		800.8	779.0
Total energy (kW)		1351.21	1329.4

To check the results, the condenser duty calculated by the network was entered in Aspen together with the specified feed stage and distillate to feed ratio. As can be noticed in Table 17, a significant difference exists between the predicted results and the actual results. The reason why this might be happening will be discussed in DISCUSSION OF PRELIMINARY RESULTS.

7.2. Results Gaussian distribution

The result for the Gaussian distribution is displayed in Table 18 together with an accuracy check obtained in Aspen. This accuracy check was performed using the optimal design predicted by the neural network as an input for a simulation in Aspen.

Table 18: Result from neural network in comparison to Aspen - Gaussian distribution

		Neural network	Aspen
No. of stages, Ns		21	21
No. of feed location, NF		5	5
Reflux ratio		0.3567	0.2849
Composition	benzene	0.99	0.83
at top	toluene	0.01	0.14
	p-xylene	0.00	0.03
Composition	benzene	0.00	0.16
at bottom	toluene	0.60	0.46
	p-xylene	0.40	0.38
Energy condenser (kW)		548.8	X
Energy reboiler (kW)		800.5	781.0
Total energy (kW)		1349.3	1329.8

The results for the Gaussian distribution also display a significant error in the case of the Gaussian distribution. The reason why this might be happening will be discussed in DISCUSSION OF PRELIMINARY RESULTS.

8. DISCUSSION OF PRELIMINARY RESULTS

The obtained results show a deviation in terms of the distillate purity in comparison to the distillate composition obtained by performing an accuracy check in Aspen. The inability of the network to correctly link the flow rates to the reboiler duty and reflux ratio might be caused by the following reason.

As discussed earlier, the influence of the distribution of the Aspen input features on the neural network performance was investigated (condenser duty, feed stage and distillate to feed ratio). A possible problem that might arise is the distribution of the neural network input parameters and thus the Aspen output parameters. While the reboiler duty and the reflux ratio both show a similar distribution to the Aspen input parameters, the values for the flow rates are very unevenly distributed, which influences the network performance. The dataset is imbalanced. The histograms for the distillate flow rates are displayed in the following figures. The x-axis presents the flow rates expressed in kmol/hr.

a) Even distribution

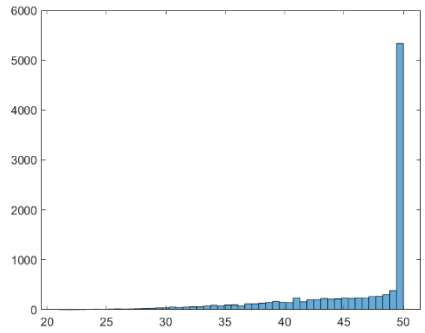
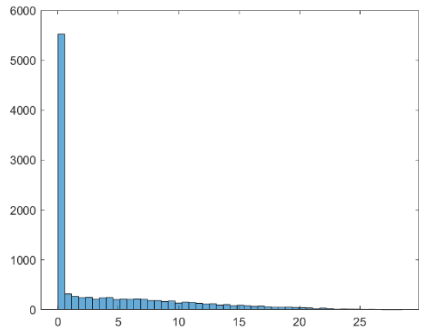
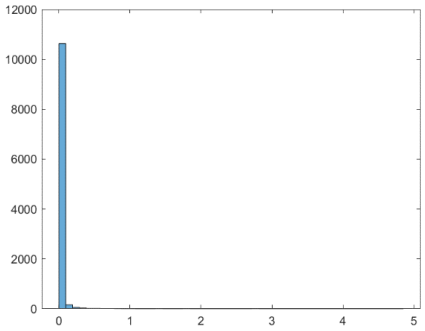
Flow rate benzene	Flow rate toluene
	
Flow rate p-xylene	
	

Figure 16: Histograms of flow rates – Even distribution

b) Gaussian distribution

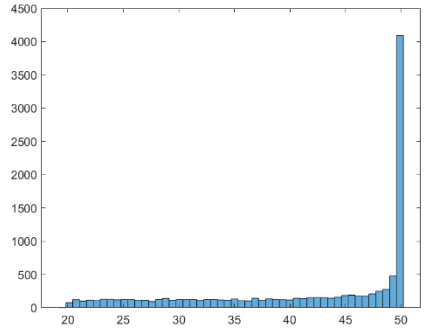
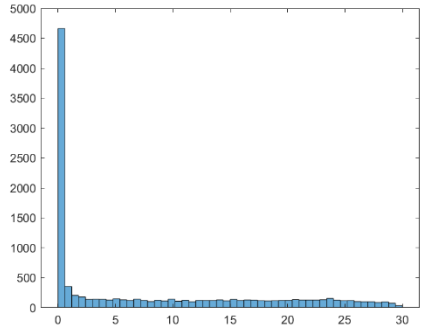
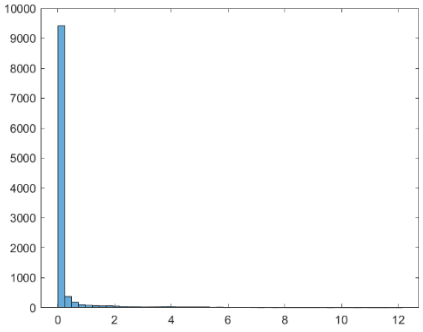
Flow rate benzene	Flow rate toluene
	
Flow rate p-xylene	
	

Figure 17: Histogram of flow rates - Gaussian distribution

There is a significant difference between the amount of data used to train the network at higher flow rates and at lower flow rates. For example, the number of data entries used to train the network between a distillate flowrate of benzene of 49.50 and 50.00 is 4,093, while the amount of data to train the network between a distillate flowrate of benzene of 49.00 and 49.50 is 477. A possible option to solve this problem is to adjust the boundaries set for the data generation for the simulations in Aspen, leading to a broader distribution for the flow rates. The broader distribution will have a positive impact on the network performance.

The largest issue is present in the case of toluene. When generating the data for the last step, the boundaries for the distillate stream of toluene are set between 0.40 and 0.5051. The network is trained with only 104 samples between 0.4 and 0.5 for the even distribution and 123 samples for the Gaussian distribution as can be seen in Figure 18 and Figure 19 . This explains why the error on the predictions is high when the overall MSE was very low.

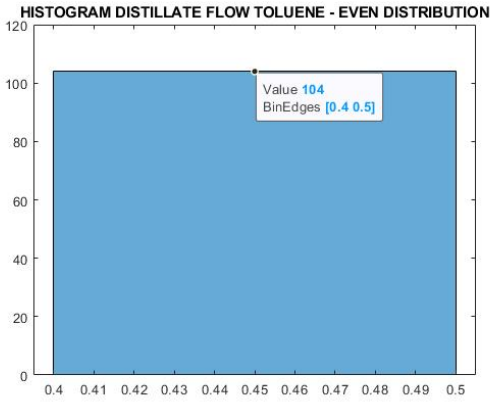


Figure 18: Histogram distillate flow toluene - Even distribution

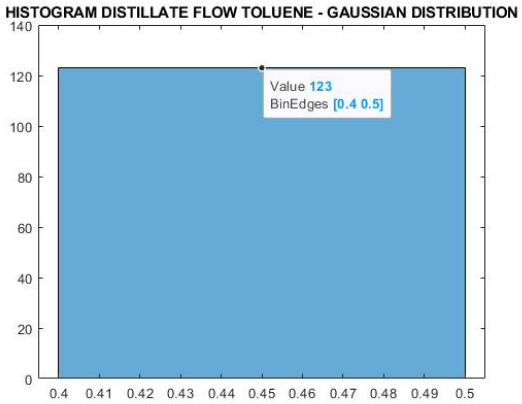


Figure 19: Histogram distillate flow toluene - Gaussian distribution

This issue will be kept in mind when performing experiments for the second case study.

9. ADJUSTMENT

Before moving on to the second case study, an attempt was made to correct the procedure used to obtain the data for the first case study. As mentioned in the discussion of the results, the uneven distribution of the simulated flow rates of the distillate might cause the insufficient performance of the neural network model in this particular range. To solve this problem, a different approach was taken when generating data for the simulations in Aspen.

9.1. Creating logical structures

Just like in the case of the normal approach of this case study, several logical structures were tested. Unlike previous results, a neural network with both the condenser duty and reboiler duty seemed to result in a good performance in case of adjusting the training data set. When using this adjusted input data set, the number of Aspen simulations that converged using the following model raised significantly. This was not the case for earlier experiments. The Aspen model that will be used for further experiments is pictured in Figure 20.

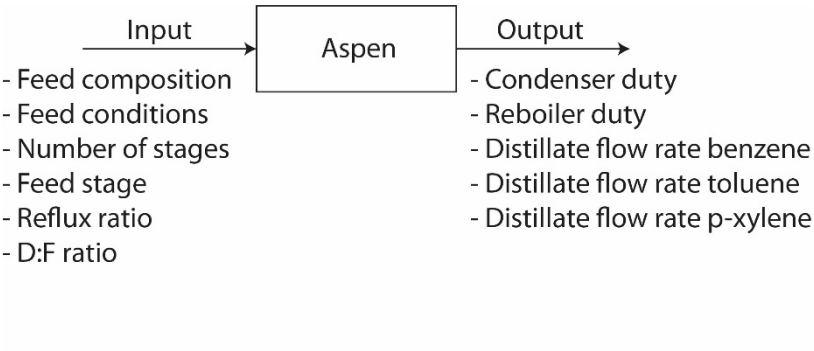


Figure 20: Aspen model for case study 1 modifications

The neural network model that will be used is pictured in Figure 21.

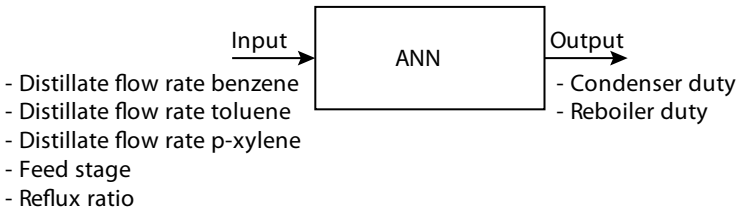


Figure 21: ANN model for case study 1 modifications

9.2. Generating input data for Aspen

As previously mentioned in the discussion of the result, the bad performance is likely caused by low number of training samples in the desired range of the distillate flow of toluene. A design specification analysis was performed to find the range for the reflux ratio needed for each feed stage to finally obtain a distillate flow rate of toluene within the desired range.

The new ranges for the reflux ratio were defined for each feed stage and performing new simulation resulted in a number of 5,483 training samples around a fractional composition of 1% for toluene, shown in Figure 22.

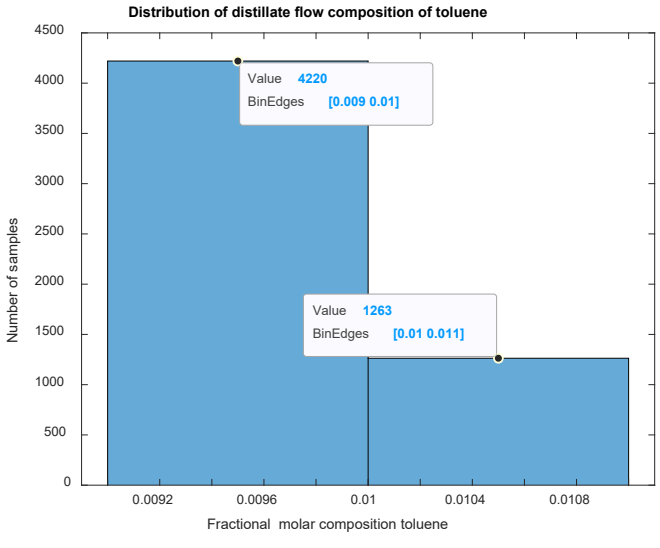


Figure 22: Histogram distillate flow toluene - adjusted ranges Aspen

9.3. Hyperparameter optimization

Like in the case of the normal approach, several network configuration parameters were varied to find the optimal network configuration. In addition to the regular iterative process to find the optimal network size, a Bayesian Optimization (BO) algorithm was used. This algorithm is used to scan larger network sizes for better performances. Because this algorithm does not take overfitting into account, the objective function – a function that is minimized to determine the optimal configuration – that will be used is the root square of the MSE of the test set, as low performances of the test set are not influenced by overfitting.

First, the performance within boundaries of 30 nodes in the first hidden layer and 30 nodes in the second hidden layer were tested. Additionally, another transfer function was applied in order to improve the performance of the network. If this improved the performance, the transfer function was kept the same and the training algorithm is applied to the new configuration. This training procedure resulted in the following optimal configurations (Table 19).

Table 19: Neural network training configurations and results

Training algorithm and activation function	trainlm - tansig	trainlm - logsig	trainbr - logsig
Network configuration	24x19	21x23	30x15
Total MSE	7.6961e-04	7.7164e-04	8.0758e-04
MSE Condenser duty	6.7012e-04	6.7411e-04	7.2146e-04
R^2 Condenser duty	1	1	1
MSE Reboiler duty	8.6910e-04	8.6917e-04	8.9371e-04
R^2 Reboiler duty	1	1	1
Test MSE	8.1801e-04	8.1739e-04	8.3055e-04
Test RMSE	2.8601e-02	2.8590e-02	2.8819e-02

Of the test configurations, the network with the 'logsig' transfer function in the first layer and trained with the 'Levenberg-Marquardt' training algorithm showed the best results for the test set.

Afterwards, the BO algorithm is performed for each configuration (training algorithm and transfer function) within a range of 50x50 (50 nodes in the first hidden layer and 50 nodes in the second hidden layer). No network configurations were found to have a better performance for the test set. As the configuration 'trainlm-logsig' resulted in the lowest root mean square error (RMSE) on the test set, this configuration was used for the next step.

9.4. Processing of network results

The final step of the process consists of using the constructed neural network to predict the optimum column settings. In order to make a comparison between the DFM and the neural network solution, the same recovery was considered for the input of the neural network. The settings leading to the lowest sum of duties in the network are reported in Table 20 together with simulation results of the same settings in Aspen performed as an accuracy check. As can be noticed, there is still a difference between the distillate flow rates used by the ANN and the distillate flow rates obtained by performing the accuracy check, but the deviation is noticeably smaller than in the case of using the old approach. The predicted duties are close to the simulated ones, with a total error percentage of 0.25% on the combined duty.

Table 20: Comparison of predicted settings for optimized energy requirements by neural network and results in Aspen Plus.

		Neural network	Aspen	
No. of stages, Ns		21	21	
No. of feed location, NF		15	15	
Reflux ratio		1.3722	1.3722	
Composition at top	Benzene	0.9900	0.9718	
	Toluene	0.0100	0.0282	
	p-xylene	0.0000	0.0000	
Composition at bottom	Benzene	0.0415	0.0585	
	Toluene	0.5714	0.5543	
	p-xylene	0.3871	0.3871	
Energy condenser (kW)		933.6	937.3	(-0.39%)
Energy reboiler (kW)		1181.3	1182.9	(-0.13%)
Total energy (kW)		2114.9	2120.2	(-0.25%)

The fact that the predicted duties are close to the real ones is confirmed by the error histograms. The following histograms (Figure 23 and Figure 24) display the number of samples of the test set with their respective percentage error.

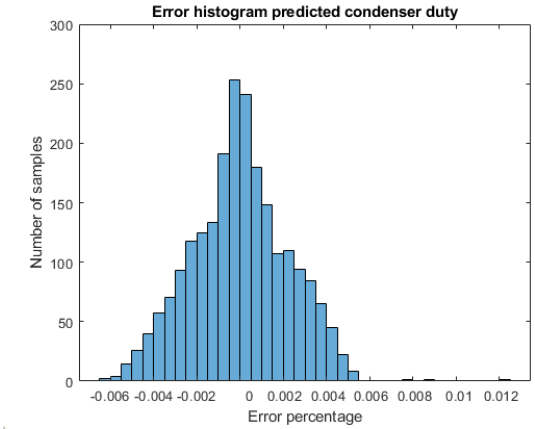


Figure 23: Error histogram condenser duty

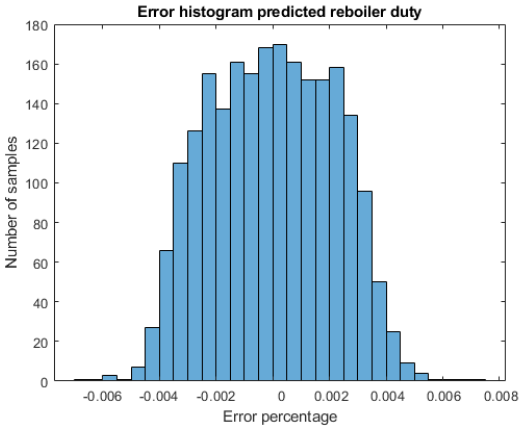


Figure 24: Error histogram reboiler duty

9.5. Comparison between shortcut, DFM and ANN

At last, a comparison is made between the results obtained by the shortcut design method, the driving force method and the neural network. The results for all three methods are included in Table 21.

Table 21: Comparison between results shortcut design, driving force method and neural network

		Shortcut	DFM	Neural network
No. of stages, Ns		21	21	21
No. of feed location, NF		11	13	15
Reflux ratio		1.60	1.41	1.37
Composition	Benzene	0.9900	0.9900	0.9718
at top	Toluene	0.0100	0.0100	0.0282
	p-xylene	0.0000	0.0000	0.0000
Composition	Benzene	0.0100	0.0415	0.0585
at bottom	Toluene	0.5900	0.5714	0.5543
	p-xylene	0.4000	0.3871	0.3871
Energy condenser (kW)		1064.4	953.9	933.6
Energy reboiler (kW)		1315.4	1201.6	1181.3
Total energy (kW)		2379.7	2155.5	2114.9
Economization			**-9.42%**	

As there is still a small difference between the distillate flow rates used by the neural network and the flow rates calculated by entering the features in Aspen as an accuracy check, it is difficult to compare them in terms of economization. The predicted settings, however, are close to the settings predicted by using the DFM method. When further optimized, the neural network solution vows to be a useful tool in predicting the optimal settings.

CHAPTER 4: CASE STUDY TWO – CREATION OF AN ADVANCED PREDICTION MODEL

This chapter contains the results obtained for the second case study. The goal of this case study is to provide an extended model in comparison to the model created in the first case study. For a specified feed composition, the model needed to provide an optimal design of the distillation column consisting of the feed temperature, the number of stages of the column, the stage at which the feed enters the column and the reflux ratio with a focus on energy minimization. This model was optimized with the intention to return a prediction with a minimal error.

1. DATA

For the purpose of this case study, some of the starting data of the first case study was reused. This data is displayed in Table 22.

Table 22: Basic data used for case study 2

Components	benzene, toluene, p-xylene
Total feed flow rate	100 kmol/hr
Feed pressure	2 atm
Column pressure	2 atm

2. CREATING LOGICAL STRUCTURES

Before moving on to the simulation of the data, a decision needed to be made about the selection of the input and output data that needed to be used in the artificial neural network as well as the input data that was used to collect data in Aspen.

As previously discussed, the goal of this case study is to predict the optimal column settings and specifications for a given feed composition. Therefore, the following parameters needed to be included in both the data collection model and the neural network model: feed composition, feed temperature, number of stages, feed stage and reflux ratio. Another parameter that was necessary to include was the distillate composition to be able to assess the model accuracy.

2.1. Logical structure for Aspen data collection

Following the parameter selection, the data needed to be collected from performing simulations in Aspen. A criterium for this stage is that simulations need to happen as efficient as possible. This is why a variety of models have been tested and compared for their efficiency. The models are based on a specific set of parameters. A limited amount of combinations are available for the specification of the column in Aspen. As determined in the analysis of the degrees of freedom, four column parameters can be specified. Of these four, the number of stages and feed stage are already chosen, two parameters remained to be specified. Concerning the specification of the feed, C+2 parameters can be specified. As the feed pressure was already determined in advance (2 atm) C+1 feed parameters remained to be defined.

Aspen allows for a range of parameters to be chosen to fill the remaining two slots when using the 'Radfrac' model. In the previous case study, the reflux ratio together with the distillate to feed ratio proved to be an efficient way to perform the simulations in Aspen. In addition to the model previously used, the composition and temperature of the feed stream were varied. This lead to the following model used for collecting data from Aspen (Figure 25).

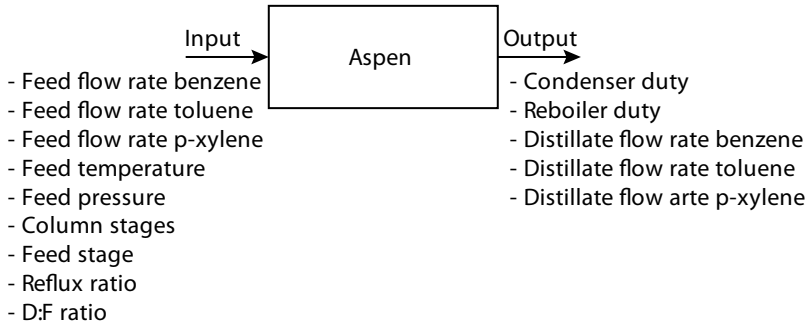
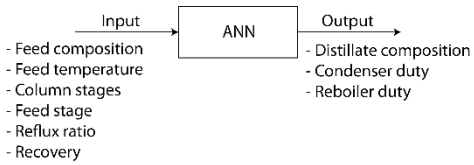
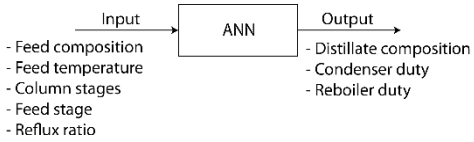
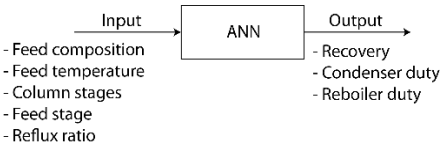
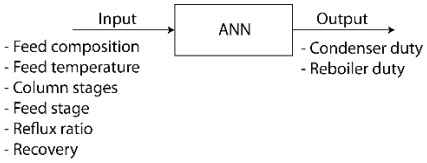
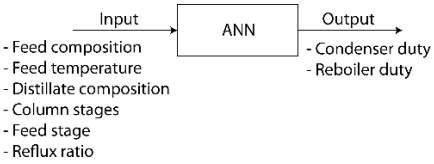


Figure 25: Selected Aspen model case study 2

2.2. Logical structure for neural network

Just like in the case of the first case study, a variety of neural network models were tested. The selection for the appropriate model was performed experimentally and was considered a process of 'trial and error'. The reasoning behind the choice of the final model is discussed in Table 23. The composition is entered in the model as the separate flow rates of each component.

Table 23: Model selection process neural network

Model	Model structure	Result
1		Model 1, 2, 3 and 4 were tested with the goal to circumvent the imbalanced data limitations. Their performance however, proved to be inadequate for application in this case study.
2		
3		
4		
5		Model 5 was the final model tested and is an extended version of the model used in the first case study. The performance of this neural network model is better than any of the other models previously tested.

3. ASPEN SIMULATIONS

To ease the simulation procedure, an Aspen 'template' file was created. The specifications that were used are described in the following paragraphs.

Properties – Specification

The components were entered in the 'specifications' tab of properties. In this case, the components that needed to be entered were benzene, toluene and p-xylene.

Properties – Methods

In the methods section, the appropriate property model was selected. The model was chosen according to the property model selection flow chart mentioned in the Aspen User Guide [31].

The feed mixture only contains three (nonpolar) components. Accordingly, the property models that could be selected were 'Redlich-Kwong-Soave' and 'Peng Robinson' (PENGROB). For the sake of this experiment, 'Peng Robinson' was selected as the property model.

Simulation – Flowsheet

Next, the block model was specified. For the sake of this experiment, the rigorous 'RadFrac' model was chosen. The RADFRAC column allows for the following parameters to be specified: feed stage, number of stages, condenser duty, distillate to feed ratio.

Simulation – Feed stream

The goal of the ANN model is to supply optimal settings for any given feed composition. Therefore the simulations needed to include very impure feed streams as well as pure feed stream.

Simulation – Column specifications

As a final step, the column settings will be specified. The goal of this experiment is to find the optimal column settings for a minimized duty. To reach this goal several parameters will be varied:

- number of column stages,
- feed stage,
- reflux ratio,
- distillate to feed ratio.

4. GENERATING INPUT DATA FOR ASPEN

The next step consists of generating the input data for the simulations in Aspen. The imbalanced data problems that occurred in the first case study proved that this is a crucial step.

4.1. Problem solving approach

The approach of solving the imbalanced data set problem for the first case study consisted of performing a design specification analysis to obtain the reflux ratio boundaries for each and every feed stage. Applying this approach here would be quite inefficient as there are too many features that vary. There is a variety of approaches that can be applied to offset this negative effect.

a) Under-sampling

Under-sampling consists of removing data that is overrepresented. Data samples that are too similar to other data samples are removed from the data set. This might cause a negative effect on the network performance as information is removed and might cause a shortage of representable data.

b) Over-sampling

This technique is often applied to classification problems with an imbalanced data issue. Over-sampling consists of duplicating data samples and uses regression to add more samples to the underrepresented range. Applying this method to regression problems is unconventional but has proven to be efficient [32], [33].

c) Combination of under- and over-sampling

Over- and under-sampling are techniques often applied to imbalanced data issues of classification problems and have been adapted to work with regression problems. Branco et al. published a paper elaborating on the use of a method called 'Synthetic Minority Oversampling Technique for Regression in combination with Gaussian Noise' (SMOGN). This method will apply a combination both under- and over-sampling to improve the performance of the data set for regression [34].

d) Gathering more qualitative data

An imbalanced data set can cause two problems. Next to scarcity in the underrepresented region, it also leads to an overrepresentation in a specific region. As models will be trained using the mean error, the overrepresented region has a larger influence on the training. Gathering more qualitative data might positively influence the imbalanced data set.

This strategy can be applied by adding the qualitative data of the first case study as this case study is just an expansion of the first one and the ranges of the first case study are still included. Because more parameters are varied, the amount of simulations is increased to 50,000 to ensure enough samples are present to provide enough relations for the network to learn from.

Both approach c and d were applied to this problem and are discussed in the following paragraph. In the first case study a Gaussian distribution as well as a uniform distribution were applied to the input parameters. As this did not have a major influence on the distribution of the output parameters, the even distribution is used for the generation of the Aspen input parameters.

4.2. Generating input parameters

50,000 data entries were prepared in an even distribution. The boundaries were chosen as following:

- number of stages: 10-42

  For a very pure feed stream, a small column might suffice while for a less pure feed stream a larger column is required to achieve a pure distillate stream. For the sake of this experiment the number of stages was varied between 10 and 42.

- feed stage: 1-41

  When generating the feed stage, the maximum feed stage needs to be taken into consideration. The feed stage cannot be larger than the number of stages specified for the data entry.

- reflux ratio: 0.4-10
- distillate to feed ratio: 0.3-0.6
- feed temperature: 70-80°C

  One of the additional parameters to be varied this time is the feed temperature. According to literature, the feed temperature also has an influence on the required energy for the separation of the feed stream [35]. The feed temperature is varied in a narrow range around the temperature used in case study 1 (75°C).

- Feed composition

  For the sake of this experiment, only feed compositions were considered with more benzene than toluene and more toluene than p-xylene. Specifying the feed composition is tricky. While two of them can be specified, the third one is always fixed as the total feed flow rate is specified at 100 kmol/hr. More important is to include an even distribution of feed compositions. This can be achieved by using an algorithm that computes random vectors with a fixed sum in MATLAB [36]. Using this code resulted in the following composition distribution displayed in Figure 26 for a number of 50,000 simulations.

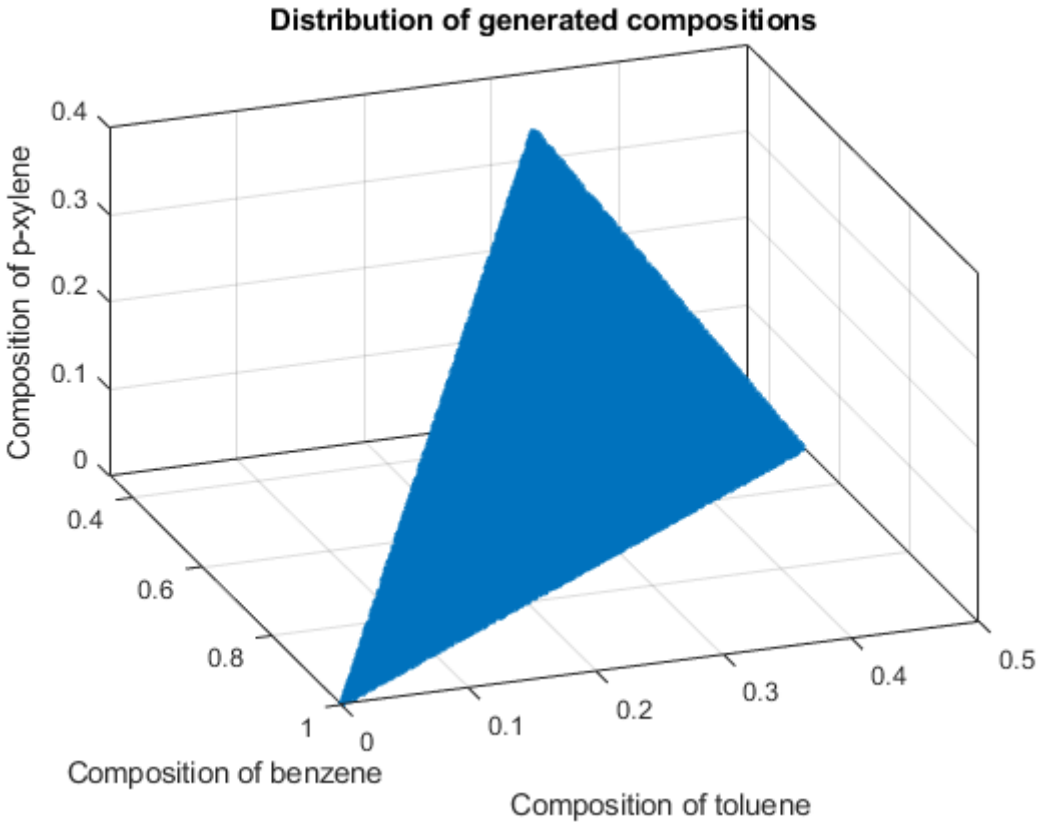


Figure 26: Distribution of generated feed compositions

5. EXTRACTING DATA AND PRE-PROCESSING

The data entries were entered as column specifications in Aspen. Simulations were run and non-converging data entries were removed. Afterwards the data was pre-processed. As discussed earlier, the data will need to be normalized in order for the network to be trained efficiently. For this purpose, mean normalization was be used.

6. HYPERPARAMETER OPTIMIZATION

A network configuration needed to be found with a minimal error on the predicted results. To find a network that is able to accurately predict the desired result, it needs to be optimized by tuning the hyperparameters. For the second case study the hyperparameter optimization was performed in four steps.

6.1. Data set

Six data sets were tested for training the model and compared based on their performance to predict the reboiler and condenser duty.

- NORMAL

  Regular data set.

- NORMAL + CS1

  Regular data with the additional qualitative data of the first case study.

- SMOGN

  Data set constructed by applying the SMOGN algorithm. The regular data set was enriched by over-sampling and overrepresented data was removed.

- SMOGN(U)

  Data set constructed by applying the SMOGN algorithm and only keeping the original, under-sampled data.

- SMOGN + CS1

  Data set constructed by applying the SMOGN algorithm. Afterwards, the qualitative data of the first case study was added.

- SMOGN(U) + CS1

  Data set constructed by applying the SMOGN algorithm and only keeping the original, under-sampled data. Afterwards, the qualitative data of the first case study was added.

The data sets were used to train neural networks within the range of 15 to 25 nodes for both hidden layers. The training was performed using the default hyperparameters mentioned in Table 11. The results of training are displayed in Table 24.

Table 24: Training results of different data types

	normal	SMOGN	SMOGN(U)
Configuration	24x20	19x23	15x19
Total MSE	0.5928	1044.2557	0.9496
Test MSE	0.5005	965.7267	1.6500

Use of the SMOGN dataset did not have a positive influence on the prediction of the reboiler and condenser duty. Accordingly, the SMOGN dataset was rejected for use in further experiments.

Afterwards, the data of the first case study was added to reduce the error caused by the imbalanced data. The results for the data set combined with the data of the first case study are displayed in Table 25.

Table 25: Training results of data types combined with results of first case study

	normal+CS1	SMOGN(U)+CS1
Configuration	21x24	19x20
Total MSE	0.0354	0.0494
Test MSE	0.0368	0.0704

Both data sets resulted in a similar error on both the test set and the total data set. To move on to the tuning of the other hyperparameters, a choice needed to be made as to which data set would be used. An important aspect of this choice is the distribution of the distillate flow rates. The training algorithms will always try to train by minimizing the MSE on the training set. If more samples are present in a specific range, these samples will have a more significant influence on the training procedure. The distributions are shown in Figure 27, Figure 28, Figure 29 and Figure 30.

NORMAL+CS1	SMOGN(U)+CS1

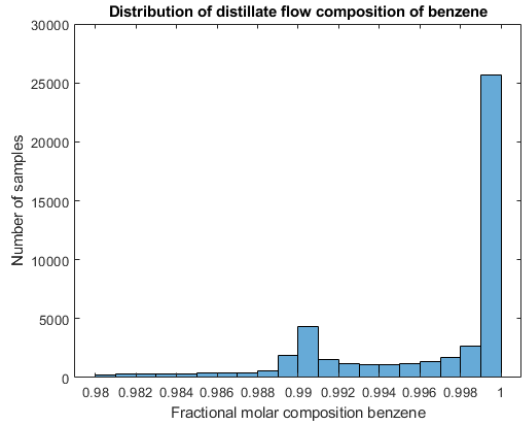


Figure 27: Distribution of distillate flow composition of benzene - NORMAL+CS1

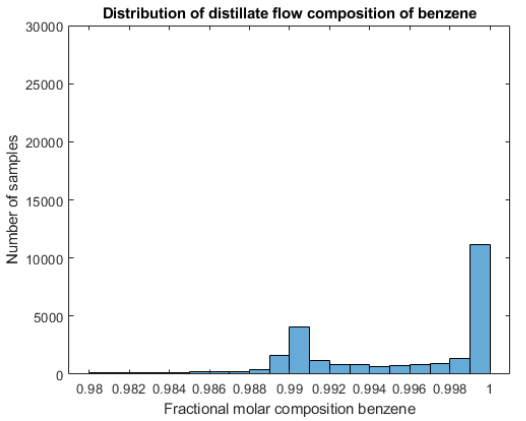


Figure 28: Distribution of distillate flow composition of benzene - SMOGN(U)+CS1

NORMAL+CS1	SMOGN(U)+CS1

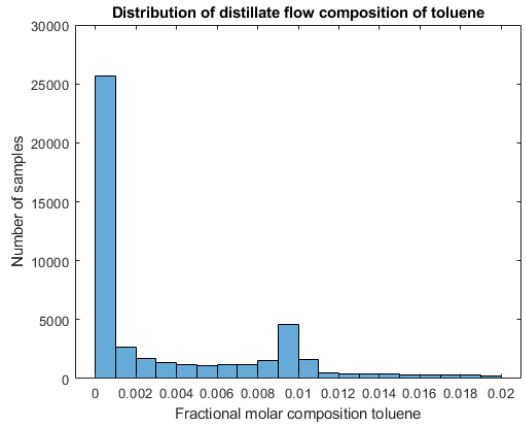


Figure 29: Distribution of distillate flow composition of toluene - NORMAL+CS1

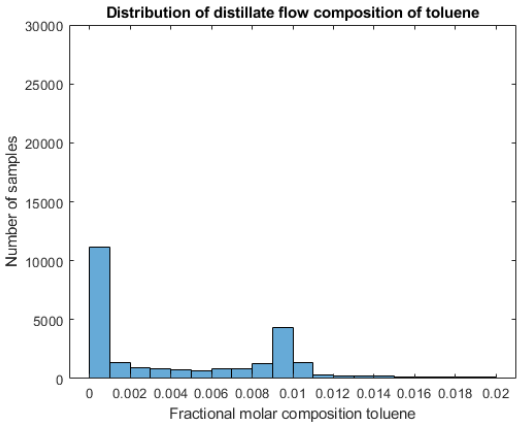


Figure 30: Distribution of distillate flow composition of toluene - SMOGN(U)+CS1

When using the random under-sampling method of the SMOGN algorithm, almost half of the samples in the undesired range are removed thus diminishing the effect of the overrepresented samples on training. The SMOGN(U) dataset combined with the samples of the first case study was used to make further improvements.

6.2. Network size, activation function and training algorithm

The similar procedure to the one of the first case study was followed for the tuning of the other hyperparameter. This resulted in the following results mentioned in Table 26.

Table 26: Hyperparameter optimization results case study 2

Training algorithm and activation function	trainlm - tansig	trainlm - logsig	trainbr - tansig
Network configuration	19x20	11x19	16x25
Total MSE	0.0494	0.2128	0.0089
MSE Condenser duty	0.0511	0.2423	0.0090
R^2 Condenser duty	1	1	1
MSE Reboiler duty	0.0477	0.1833	0.0089
R^2 Reboiler duty	1	1	1
Test MSE	0.0704	0.3085	0.0213
Test RMSE	0.2653	0.5554	0.1459

The 'logisg' activation function did not improve the performance of the neural network, while changing the training algorithm to 'trainbr' resulted in a better performance.

Expanded 'trainbr' configurations were tested using the Bayesian optimization algorithm. The results of this algorithm showed that no better configuration could be found within a range of 40x40 nodes in the hidden layers.

6.3. Final model configuration

The optimization process resulted in the optimal network configurations displayed in Table 27.

Table 27: Final network settings

Network size	16x25
Activation function	tansig
Training algorithm	trainbr

7. PROCESSING OF NETWORK RESULTS

In this step, the constructed neural network was put into action. The goal of this case study is to provide the optimal design given a specified feed composition and purity of the distillate stream. Accordingly, both the feed composition and the distillate composition were set fixed.

The optimal conditions were determined for two cases mentioned in Table 28.

Table 28: Feed compositions for second case study

	Feed flow rate benzene (kmol/hr)	Feed flow rate toluene (kmol/hr)	Feed flow rate p-xylene (kmol/hr)
Case 1 (impure feed)	40	35	25
Case 2 (pure feed)	80	15	5

10,000 data entries were generated to be entered in the neural network. These parameters include the number of column stages, the feed stage, the feed temperature, reflux ratio and the distillate flow rate. The following boundaries were chosen for the data to be generated within.

a. Number of column stages: 10-42
b. Feed stage: 1- Number of column stages

   The feed stages were generated within the range of 1 and the total number of column stages. The feed stage cannot be larger than the number of stages specified for the data entry.

c. Reflux ratio: 0.4-10
d. Distillate flow rate

   The distillate flow rate was generated to result in a recovery of at least 90%. The distillate flow rate is defined as a distillate stream with 90-100% recovery of benzene consisting of 99 mol% benzene and 1 mol% toluene.

e. Feed temperature: 70-80°C

7.1. Results feed 1

The results for the impure feed are displayed in Table 29 with an accuracy check obtained in Aspen using the predicted design.

Table 29: Results for impure feed

		Neural network	Accuracy check	
No. of stages, Ns		26	26	
No. of feed location, NF		15	15	
Reflux ratio		0.4184	0.4184	
Feed temperature		78.9628	78.9628	
Composition	benzene	0.9900	0.7964	
at top	toluene	0.0100	0.1832	
	p-xylene	0.0000	0.0204	
Composition	benzene	0.0286	0.1618	
at bottom	toluene	0.5640	0.4503	
	p-xylene	0.4074	0.3880	
Energy condenser (kW)		423.2	452.8	-6.54%
Energy reboiler (kW)		682.1	694.0	-1.72%
Total energy (kW)		1105.3	1146.8	-3.62%

While the predicted energy requirement is close to the required energy obtained in Aspen, there is still a difference between the distillate flow rate used by the neural network and the distillate flow rate acquired by performing the accuracy check.

7.2. Results feed 2

The results for the pure feed are displayed in Table 30.

Table 30: Results for clean feed

		Neural network	Accuracy check	
No. of stages, Ns		17	17	
No. of feed location, NF		9	9	
Reflux ratio		0.4007	0.4007	
Feed temperature		77.5054	77.5054	
Composition	benzene	0.9900	0.9593	
at top	toluene	0.0100	0.0400	
	p-xylene	0.0000	0.0007	
Composition	benzene	0.2540	0.3424	
at bottom	toluene	0.5523	0.4661	
	p-xylene	0.1937	0.1916	
Energy condenser (kW)		836.4	851.8	-1.81%
Energy reboiler (kW)		983.3	996.3	-1.30%
Total energy (kW)		1819.7	1848.1	-1.54%

The error on the predicted required energy is lower than in the case of the impure feed. It can be noted that just like in the case of the impure feed stream there is still a deviation present in the distillate flow rates used by the ANN compared to the ones obtained by the accuracy check. However, this deviation is noticeably smaller than in the case of the impure feed stream.

7.3. Discussion of results

There are no obvious reasons that explain the difference between the performance of the pure and impure feed stream.

In the first part of the research it was thought that the imbalanced data problem was to blame for the deviations in the distillate flow rates. Later on, however, it was established that another factor caused the deviations in the distillate flow rates. The reflux ratio is part of the input features of the neural network. An important observation to be made is that the reflux ratio is the only parameter of the input features that is directly corelated to the energy supplied to the column. As the distillate composition is also part of the input features, this causes a problem. The reflux ratio would need to accurately correspond to the reflux ratio needed to reach a distillate composition of 99 mol% benzene and 1 mol% toluene.

To solve this problem using the same model, simulations would need to be performed to calculate the reflux ratio for each sample, corresponding to the reflux ratio required to reach the proposed distillate composition. This would defeat the purpose of using the ANN solution.

Another approach that can be taken to solve to decrease the influence of this problem is deducted from the approach used to solve the supposed imbalanced data problem in the first case study. The data collection can be divided into smaller parts, with reflux ratio boundaries – to reach a distillate composition of approximately 99 mol% benzene and 1 mol% toluene – being determined by performing design specifications analyses. Attention has to be payed towards the fact that this will still not provide an accurate solution, but the predicted distillate flow rates will be a lot closer to the actual distillate flow rates.

An accurate solution can only be found if a new, efficient model is created with the reflux ratio as one of the output features, together with the reboiler and condenser duty.

8. TIME MEASUREMENT

The most significant advantage of using neural networks to perform simulations is the time it takes to perform the simulations. While performing experiments in Aspen Plus can take a few seconds for each simulation, neural networks are able to perform thousands of simulations in a fraction of a second. To be able to assess the time consumption of both methods, an experiment was performed. For each method (Aspen Plus and a neural network model), the time needed to perform simulations was recorded. The results were interpolated to one simulation. It has to be noted that these time measurements are dependent of the hardware specifications of the system on which the experiments were performed. To be able to compare the time, both experiments were run on the same system. The results are displayed in Table 31.

Table 31: Results of time measurement

Number of simulations	1,000,000	100,000	10,000	1
ANN	3.1 e-01 s	3.0 e-02 s	4.9 e-03 s	4.9 e-07 s
Aspen Plus	x	x	2.0 e+04 s	2.0 s

What can be noted from these time measurements is that in the time it takes for Aspen Plus to perform one simulation, the ANN can perform more than a million simulations indicating the importance of using ANN's for various applications.

CHAPTER 5: CONCLUSION & OUTLOOK

1. CONCLUSION

In the first part of this dissertation, a neural network model was constructed which predicted the required energy for the separation of a feed stream with a fixed composition. The neural network model was used to find an optimized design of a distillation column, which consists of the feed stage and the reflux ratio. The neural network was able to accurately predict the required energy for the separation of the feed stream (total error percentage of 0.2%). However, the usage of imbalanced data to train the neural network model supposedly caused a deviation in the distillate flow rates used in the model compared to the distillate flow rates obtained by performing an accuracy check. By performing new simulations, the number of samples within the desired range improved and the used distillate flow rates were closer to the ones obtained by the accuracy check. Because there was still a deviation present its prediction could not be compared against the optimal design of the DFM.

The second part of this dissertation consisted of the construction of an expanded version of the neural network created in the first part. Additionally, the network needed to predict the optimal number of stages and the feed temperature for different feed compositions. Although it was established that obtaining evenly distributed data for the distillate flow rates would be best practice, it was not possible to obtain an evenly distributed data set concerning the distillate flow rates. Instead, to diminish the effect of the imbalanced data set: more simulations were performed, qualitative data of the first case study was taken into consideration and the SMOGN method was applied. The over-sampling of the SMOGN method did not have a positive influence on the network performance, thereby only the under-sampling technique was used. The optimal design was determined for two different feed compositions: one impure feed and one more pure feed. The neural network models were again able to accurately predict the required energy for the separation (max total error percentage of 3.62%). But once again, the used distillate flow rates by the neural network model deviate from the ones determined by performing the accuracy check. The deviation of the distillate flow rates was found to be higher for the impure feed than for the pure feed. Later on, it was established that the reflux ratio being present on the input side of the ANN was to blame for the distillate flow rates deviation issue as the reflux ratio is strongly corelated to the required energy for the separation.

As a side experiment, the time needed for one simulation was measured for Aspen Plus and the ANN solution. While the average time it took for one simulation in Aspen Plus was 2 seconds, the ANN solution can perform one million simulations in 0.31 seconds. This time reduction is an important advantage of the use of neural networks and is one of the reasons why neural networks are being deployed in various domains of science.

2. FUTURE OUTLOOK

To confirm if neural networks can be applied to design problems, future experiments need to be performed using a balanced data set or using more data within the desired ranges. The model used in this research can be improved by dividing the data collection in multiple small sections with reflux ratio boundaries determined by multiple design specification analyses. Attention has to be payed towards the fact that using this method, an accurate prediction cannot be performed, but it can give an idea as to the ranges the optimal configuration is situated in.

For the further optimization of the networks the following options can be explored:

- acquire more qualitative data between the desired ranges;
- vary the starting weights: Up until now, every training iteration uses the same starting weights to train the network. Varying the weights might help the training algorithm to find a more suitable configuration;
- vary the division of the data set: The goal of the hyperparameter optimization in this case study was to compare different network configurations by their total MSE and test MSE. Therefore, fixed data division was used in order to compare them. A more accurate approach might be to test multiple divisions of the data set and use to average error to define the performance of the network configuration;
- consider other options for the transfer function and training algorithm.

Additionally, to be able to make an accurate prediction a new, accurate network has to be created with the reflux ratio on the output side of the network together with the reboiler and condenser duty.

LIST OF REFERENCES

[1] "KU Leuven investeert in nieuwe onderzoeksinfrastructuur op Campus Diepenbeek – Faculteit Industriële Ingenieurswetenschappen," *KU Leuven*, 2019. https://iiw.kuleuven.be/nieuws/nieuwonderzoekslabodiepenbeek (accessed Mar. 08, 2020).

[2] M. Faris, A. Noor, F. Mohd, I. Asri, I. Norazana, and K. Mohd, "Design of Energy Efficient Distillation Columns Systems," in *International symposium on green manufacturing and applications*, 2014, no. June, pp. 66–69, [Online]. Available: https://www.researchgate.net/publication/263354820.

[3] V. Goncalves, K. Maria, and A. B. F. da Silv, "Applications of Artificial Neural Networks in Chemical Problems," in *Artificial Neural Networks - Architectures and Applications*, InTech, 2013, pp. 203–223.

[4] C. Nicholson, "A Beginner's Guide to Neural Networks and Deep Learning," *Pathmind*, 2020. https://pathmind.com/wiki/neural-network (accessed Feb. 07, 2020).

[5] A. Ng, "Machine Learning | Coursera," *Coursera*, 2020. https://www.coursera.org/learn/machine-learning (accessed Feb. 07, 2020).

[6] C. W. Coley *et al.*, "A graph-convolutional neural network model for the prediction of chemical reactivity," *Chem. Sci.*, vol. 10, no. 2, pp. 370–377, 2019, doi: 10.1039/c8sc04228d.

[7] H. Seunggu, "Neurons: What are they and how do they work?," *Medical News Today*, 2017. https://www.medicalnewstoday.com/articles/320289.php#neurons_look_like (accessed Feb. 07, 2020).

[8] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox ™ 7 User ' s Guide*. 2010.

[9] C. W. Coley, W. H. Green, and K. F. Jensen, "Machine Learning in Computer-Aided Synthesis Planning," *Acc. Chem. Res.*, vol. 51, no. 5, pp. 1281–1289, 2018, doi: 10.1021/acs.accounts.8b00087.

[10] J. Dotai, "Everything You Need to Know About Artificial Neural Networks," *Medium*, 2015. https://medium.com/technology-invention-and-more/everything-you-need-to-know-about-artificial-neural-networks-57fac18245a1 (accessed Feb. 07, 2020).

[11] C. W. Coley *et al.*, "A graph-convolutional neural network model for the prediction of chemical reactivity - Electronic Supplementary Material 2," *Chem. Sci.*, vol. 10, no. 2, 2019, doi: 10.1017/CBO9781107415324.004.

[12] C. W. Coley *et al.*, "A graph-convolutional neural network model for the prediction of chemical reactivity," *Chem. Sci.*, vol. 10, no. 2, pp. 370–377, 2019, doi: 10.1039/c8sc04228d.

[13] H. Gao, T. J. Struble, C. W. Coley, Y. Wang, W. H. Green, and K. F. Jensen, "Using Machine Learning to Predict Suitable Conditions for Organic Reactions," *ACS Cent. Sci.*, vol. 4, no. 11, pp. 1465–1476, 2018, doi: 10.1021/acscentsci.8b00357.

[14] A. Moghadassi, F. Parvizian, and S. M. Hosseini, "Application of artificial neural network for prediction of liquid viscosity," *Indian Chem. Eng.*, vol. 52, no. 1, pp. 37–48, Mar. 2010, doi: 10.1080/00194501003759688.

[15] I. V. Tetko and V. Y. Tanchuk, "Application of associative neural networks for prediction of lipophilicity in ALOGPS 2.1 program," *J. Chem. Inf. Comput. Sci.*, vol. 42, no. 5, pp. 1136–1145, 2002, doi: 10.1021/ci025515j.

[16] C. C. Chen and Y. Song, "Extension of nonrandom two-liquid segment activity coefficient model for electrolytes," *Ind. Eng. Chem. Res.*, vol. 44, no. 23, pp. 8909–8921, 2005, doi: 10.1021/ie0503592.

[17] L. Ma, C. Cheng, X. Liu, Y. Zhao, A. Wang, and P. Herdewijn, "A neural network for predicting the stability of RNA/DNA hybrid duplexes," *Chemom. Intell. Lab. Syst.*, vol. 70, no. 2, pp. 123–128, Feb. 2004, doi: 10.1016/j.chemolab.2003.10.002.

[18] G. W. Dombi and J. Lawrence, "Analysis of protein transmembrane helical regions by a neural network," *Protein Sci.*, vol. 3, no. 4, pp. 557–566, Dec. 1994, doi: 10.1002/pro.5560030404.

[19] J. Meiler, "PROSHIFT: Protein chemical shift prediction using artificial neural networks," *Journal of Biomolecular NMR*, vol. 26, no. 1. pp. 25–37, May 01, 2003, doi: 10.1023/A:1023060720156.

[20] K. Petritis *et al.*, "Use of artificial neural networks for the accurate prediction of peptide liquid chromatography elution times in proteome analyses," *Anal. Chem.*, vol. 75, no. 5, pp. 1039–1048, Mar. 2003, doi: 10.1021/ac0205154.

[21] C. Cimpoiu, V. M. Cristea, A. Hosu, M. Sandru, and L. Seserman, "Antioxidant activity prediction and classification of some teas using artificial neural networks," *Food Chem.*, vol. 127, no. 3, pp. 1323–1328, Aug. 2011, doi: 10.1016/j.foodchem.2011.01.091.

[22] D. M. Himmelblau, R. W. Barker, and W. Suewatanaku, "Fault classification with the aid of artificial neural networks," *IFAC Symp. Ser.*, vol. 24, no. 6, pp. 541–545, 1992, doi: 10.1016/s1474-6670(17)51197-0.

[23] D. M. Himmelblau, "Applications of Artificial Neural Networks in Chemical Engineering," *Korean K. Chem. Eng.*, vol. 17, no. 4, pp. 373–392, 2000, doi: 10.5772/51275.

[24] P. C. Wankat, *Separation Process Engineering*. Westford: Pearson, 2011.

[25] L. Braeken, *Scheidingsprocessen*. Diepenbeek: UHasselt, 2018.

[26] B. Satola, "About ActiveX automation with Aspen Plus," *chejunkie*, 2019. https://chejunkie.com/knowledge-base/activex-automation-aspen-plus/ (accessed Jan. 12, 2020).

[27] A. Abril, "Aspen Plus - Matlab Link." MATLAB Central File Exchange, 2018, Accessed: Jan. 12, 2020. [Online]. Available: https://nl.mathworks.com/matlabcentral/fileexchange/69464-aspen-plus-matlab-link.

[28] "Random seed," 2020. https://en.wikipedia.org/wiki/Random_seed (accessed Apr. 05, 2020).

[29] G. Heath, "Difference between trainlm and trainbr," *Mathworks*, 2018. https://nl.mathworks.com/matlabcentral/answers/428741-what-is-the-difference-between-trainlm-and-trainbr (accessed Apr. 05, 2020).

[30] D. Mathis, "Bayesian optimization algorithm." MATLAB Central File Exchange, 2018.

[31] *Aspen Plus ® Aspen Plus User Guide*, 10.2. Cambridge: Aspentech, 2000.

[32] P. Branco, L. Torgo, and R. P. Ribeiro, "Pre-processing approaches for imbalanced distributions in regression," *Neurocomputing*, vol. 343, no. May, pp. 76–99, May 2019, doi: 10.1016/j.neucom.2018.11.100.

[33] E. Oliveira, J. Gama, Z. Vale, H. Lopes, C. Eds, and R. Goebel, "Progress in Artificial Intelligence," in *Progress in Artificial Intelligence*, 2013, vol. 8154, no. October 2015, pp. 378–389, doi: 10.1007/978-3-642-40669-0.

[34] P. Branco, R. P. Ribeiro, L. Torgo, B. Krawczyk, and N. Moniz, "SMOGN: a Pre-processing Approach for Imbalanced Regression," *Proc. Mach. Learn. Res.*, vol. 74, no. September, pp. 36–50, 2017.

[35] S. H. Lee and M. Binkler, "Optimize design for distillation feed," *GTC Tech*, 2011. https://www.gtctech.com/optimize-design-for-distillation-feed/ (accessed Apr. 28, 2020).

[36] R. Stafford, "Random vectors with fixed sum." MATLAB Central File Exchange, 2006.

APPENDIX

Appendix A: Training code

```
% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
% Created 10-May-2020 22:44:25
%
% This script assumes these variables are defined:
%
%   InputModel1 - input data.
%   OutputModel1 - target data.

x = InputModel1';
t = OutputModel1';

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm';  % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 1;
net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivision
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse';  % Mean Squared Error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression', 'plotfit'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
```

```
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
```