

# Data-driven Optimization: Applications to Energy Infrastructure and Process Industry

by

Mohammed Adel Abdullah Al-Katheri

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Chemical Engineering

Waterloo, Ontario, Canada, 2021

© Mohammed Adel Abdullah Al-Katheri 2021

## Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner

SIMANT UPRETI  
Professor

Supervisor(s)

ALI ELKAMEL  
Professor

PETER DOUGLAS  
Professor

Internal Member

ERIC CROISET  
Professor

Internal Member

BOXIN ZHAO  
Professor

Internal-external Member

RAMADAN EL-SHATSHAT  
Professor

## **Author's Declaration**

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Abstract

Nowadays, the existence and ease of access to massive amounts of data encourage proposing data-driven solutions. As optimization has always been based on the interchange between models and data, high-level optimization tasks such as planning and scheduling will extremely benefit from information mined from massive data sets. The development of big data tools (i.e., machine learning) has proven superiority over traditional data tools in dealing with vast amounts of data, data with undefined structure and capturing important information from data in a very efficient and computationally tractable manner. Therefore, in this work, big data tools are implemented to address the challenges associated with planning models of energy infrastructure that incorporate renewable resources and chemical engineering processes, namely, uncertainty handling, multiscale modelling, and unit process equation complexity.

A Data-driven stochastic optimization framework that leverages big data in design and operation of power generation planning is proposed. A k-means clustering algorithm is adopted to generate uncertainty scenarios for the stochastic optimization framework. These scenarios are used as inputs to the stochastic model where the proposed model is formulated as a mixed integer linear program (MILP) and solved using GAMS. The proposed approach is applied to different power planning models that include unit commitment (UC) characteristics where the size of uncertainty scenarios is reduced. Results show that the proposed approach is an effective tool to generate reduced size stochastic scenarios.

The design and operation of energy hub problem involves the integration of decision levels with different time scales that usually lead to multiscale models which are computationally expensive. The multiscale (i.e., planning and scheduling) energy hub systems that incorporate renewable energy resources become more challenging to model due to a high level of intermittency associated with renewable energy. A mathematical programming-based general clustering approach is applied to reduce the size of multiple attributes demand data and tackle the computational complexity of multiscale energy hub problems. Multiscale with multiple attributes energy hub incorporating hydrogen storage is modelled as a MILP stochastic optimization problem under wind uncertainty. Different case studies are generated under different environmental consideration to assess the efficiency of the clustering approach and stochastic formulation. Assessments conclude that the clustering approach is an effective tool to reduce the size of the original model while maintaining good results.



Recent advancements in supervised machine learning tools have demonstrated their ability to achieve accurate and efficient prediction results. Therefore, in this study, these tools are employed as alternative approaches to model a specific application in the gas industry. The chosen application is a natural gas condensate stabilization process based on operating data. Natural gas condensate treatment involves condensate stabilization process in which light end components are removed and thus condensate vapour pressure is reduced to meet storage and transportation specification. Different supervised machine learning models are developed to predict the performance of two industrial condensate stabilizer units. Large datasets of the two different industrial condensate stabilizers, including operating data of input-output variables, are utilized to develop and evaluate these models. The main purpose of developing these machine learning models is to predict the important parameters of the final stabilized liquid. Results attained from this study showcase the capability of the developed models to offer reliable and accurate predictions. A data-driven surrogate-based optimization framework is developed, where the generated machine learning models can serve as a convenient replacement for detailed first principle models, to find the optimal values of the variables corresponding to the minimal operational energy consumption. The proposed framework can benefit the gas industry to simultaneously achieve process efficiency, profitability, and safety.

## **Acknowledgements**

I would like to acknowledge and thank Prof. Ali Elkamel for his patient guidance, encouragement, and advice he has provided throughout my time as his student. I have been extremely lucky to have a supervisor who cared about me not only as a student but also as a person where he provided all kind of mental and technical supports.

I must also express my gratitude to Prof. Peter Douglas for his continuous follow up and guidance throughout my program.

Furthermore, I am thankful to the examining committee members, Prof. Simant Upreti, Prof. Eric Croiset, Prof. Boxin Zhao and Prof. Ramadan El-Shatshat for reviewing the presented work and providing insightful feedback.

Lastly, I would like to thank my parents for all their supports and my beautiful family, my brother and sisters, who always remember me in their prayers.

## **Dedication**

*To my mother Asia and my father Adel*

# Table of Contents

Examining Committee Membership .....	ii
Author’s Declaration.....	iii
Abstract .....	iv
Acknowledgements.....	vi
Dedication .....	vii
Table of Contents .....	viii
List of Figures .....	xii
List of Tables .....	xviii
List of Abbreviations .....	xx
List of Symbols .....	xxi
Chapter 1 Introduction.....	1
1.1 Project Motivations .....	1
1.2 Project Goals and Contributions .....	4
1.3 Dissertation Outline.....	5
Chapter 2 Background and Literature Review .....	7
2.1 Mathematical Optimization Methods.....	7
2.1.1 Deterministic Approach .....	8
2.1.2 Worst-Case Approach.....	10
2.1.3 Model with Chance (Probabilistic) Constraint .....	10
2.1.4 Model with Recourse .....	12
2.2 Big Data Tools .....	14
2.3 Unsupervised Machine Learning Methods .....	17
2.3.1 K-Means Clustering .....	17
2.4 Supervised Machine Learning Methods.....	18

2.4.1	Supervised Machine Learning Evaluation .....	18
2.4.2	Ordinary Least Square Linear Regression .....	21
2.4.3	Ridge Linear Regression.....	22
2.4.4	Lasso Linear Regression.....	22
2.4.5	Support Vector Regression .....	23
2.4.6	Artificial Neural Network.....	28
2.4.7	Example of Applying a Machine Learning Method .....	36
2.5	General Literature Review .....	39
Chapter 3	Machine Learning Framework for the Formulation of Efficient Scenarios for Stochastic Programming: Application to Power Generation Planning Under Demand and Wind Data Uncertainties.....	41
3.1	Introduction .....	41
3.2	Deterministic Design and Operation Formulation of Power Generation Model .....	43
3.2.1	Results and Discussions of the Deterministic Power Generation Planning Model	49
3.3	Stochastic Data-Driven Design and Operation of Power Generation Planning model..	53
3.3.1	Stochastic Model Formulation.....	53
3.3.2	Data-Driven Uncertainty Scenario Construction Using Clustering.....	55
3.3.3	Comparison Between Stochastic and Deterministic Results When There Are No Environmental Considerations.....	61
3.3.4	Comparison between Stochastic and Deterministic Results When There Are Environmental Considerations.....	63
3.4	Application of the Proposed Clustering Approach to Generate Stochastic Scenario for Day-Ahead - Real Time UC Power Generation Planning Model.....	63
3.5	Conclusion and Future Work .....	67
Chapter 4	Clustering Approach for the Efficient Solution of Multiscale Stochastic Programming Problems: Application to Energy Hub Design and Operation Under Uncertainty	69

4.1	Introduction .....	69
4.2	Methodology: Clustering Algorithm and Stochastic Scenario Generation .....	74
4.2.1	General Clustering Algorithm Formulation .....	75
4.2.2	Heuristic Approach for Multiple Attributes Data Size Reduction .....	77
4.2.3	Clustering Assessment .....	80
4.2.4	Uncertainty Modelling of Wind speed .....	85
4.3	Application of the Multiple Attribute Clustering Approach to Energy Hub .....	88
4.3.1	Energy Hub Model Formulation .....	88
4.4	Results and Discussions .....	95
4.4.1	Results Without GHG Emissions Constraint .....	96
4.4.2	Environmental Considerations (CO <sub>2</sub> emission regulation) .....	99
4.4.3	Stochastic Energy Hub Formulation Assessment .....	107
4.5	Conclusion and Future Work .....	109
Chapter 5 Machine Learning Approach for Modeling and Optimization of Complex Systems: Application to Condensate Stabilizer Plant .....		
5.1	Introduction .....	113
5.2	Condensate Stabilizer Process .....	116
5.3	Data Pre-processing .....	118
5.3.1	Outlier Removal .....	125
5.3.2	Feature Dependency and Selection .....	131
5.3.3	Data Scaling .....	140
5.4	Machine Learning Models Developments .....	142
5.4.1	Linear Regression Models .....	142
5.4.2	Development of Detailed Models .....	146
5.4.3	Effect of Some Operating Conditions on the Output Variables .....	160

5.5	Surrogate-Based Optimization Model Developments.....	164
5.5.1	Trusted-Region Algorithm.....	164
5.5.2	Optimization Problem Model Formulation and Solution .....	167
5.5.3	Integration of Machine Learning Model within Optimization Constraint (phase 1) 168	
5.5.4	Integrating of Machine Learning Model Within Optimization Constraints and Objective Function.....	174
5.6	Conclusion and Future Work .....	186
Chapter 6	Conclusions and Future Work .....	188
References	.....	191
Appendices	.....	206
Appendix A	.....	206
Appendix B	.....	211

## List of Figures

Figure 1.1. Data-driven optimization project scope.....	5
Figure 2.1. Schematic representation of GA process.....	28
Figure 2.2. Schematic representation of a single neuron .....	29
Figure 2.3. Graphical representation of example of feed-forward fully connected three-layer perceptron .....	32
Figure 2.4. Generated data example .....	37
Figure 2.5. The variation of training score and validation score as a function of $C$ .....	38
Figure 2.6. Comparison of model curves using different values of $C$ .....	38
Figure 3.1. <i>Vestas V90-1.8</i> wind turbine power output as a function of wind speed. ....	49
Figure 3.2. Average demand profile .....	50
Figure 3.3. Average wind speed profile.....	50
Figure 3.4. Power output results for units in each time .....	51
Figure 3.5. Energy scheduling results for all units in each time.....	51
Figure 3.6. The effect of CO <sub>2</sub> emissions on the objective function (blue line) and number of wind turbine needed to be installed (red line).....	52
Figure 3.7. Process of rearranging the dimension of wind speed and electric demand.....	57
Figure 3.8. Demand profile for selected days .....	58
Figure 3.9. Wind speed profile for selected days.....	58
Figure 3.10. Actual electricity demand and its computed cluster centres for 1-year time horizon .....	58
Figure 3.11. Actual wind speed and its computed cluster centres for 1-year time horizon.....	58
Figure 3.12. Scenario generation for the stochastic data-driven power generation planning model .....	59
Figure 3.13. Effect of cluster number on the average error for electricity demand.....	60
Figure 3.14. Effect of cluster number on the average standard deviation for electricity demand	60
Figure 3.15. Effect of cluster number on the average error for wind speed .....	60
Figure 3.16. Effect of cluster number on the average standard deviation for wind speed.....	60
Figure 3.17. Electricity demand clusters.....	60
Figure 3.18. Wind speed clusters .....	60
Figure 3.19. 50 scenarios per-unit wind realization factor for existing wind farm [94].....	65



Figure 3.20. 6 clusters per-unit wind realization factor for existing wind farm .....	65
Figure 3.21. Schematic representation of the four-node power system.....	66
Figure 3.22. Dispatch day-ahead decisions for full-size and reduced size UC planning model..	67
Figure 4.1. Application of the proposed clustering approach to the multiscale decision-making problem .....	74
Figure 4.2. Conceptual representation for Pareto frontier .....	75
Figure 4.3. Graphical representation of the heuristic size reduction algorithm for multiple attributes.....	79
Figure 4.4. Graphical representation of the heuristic size reduction algorithm for multiple attributes for a single weight factor .....	80
Figure 4.5. Pareto frontiers for normal and sequence clustering using different number of clusters .....	85
Figure 4.6. Actual electricity (top) and heat demand (bottom) and its computed cluster curves (4,5 and 6 clusters) using normal (left) and sequence clustering approach (right) for 1-year time horizon .....	85
Figure 4.7. Actual and best fit distribution wind speed profile .....	87
Figure 4.8. Wind speed stochastic scenarios .....	87
Figure 4.9. Understudy energy hub architecture.....	90
Figure 4.10. Comparison between original and clustered energy hub solution in terms of solution quality and time.....	97
Figure 4.11. Design results comparison between original and clustered energy hub cases .....	98
Figure 4.12. Energy hub’s utility production rates comparison between original and clustered cases .....	99
Figure 4.13. The effect of CO <sub>2</sub> emission regulation on the objective function (lines) and number of wind turbine needed to be installed (square marker).....	101
Figure 4.14. Comparison between original and clustered energy hub solution in terms of solution quality and time under CO <sub>2</sub> emissions restriction.....	101
Figure 4.15. The number of energy hub units powered by fossil fuel that are installed under CO <sub>2</sub> emissions regulation.....	103
Figure 4.16. Number of installed wind turbines suggested by original and clustered cases under CO <sub>2</sub> emissions regulation.....	104

Figure 4.17. Number of installed storing facilities suggested by original and clustered cases under CO <sub>2</sub> emissions regulation.....	105
Figure 4.18. Comparison between original and clustered cases utilities production rates of energy hub units powered by fossil fuel under CO <sub>2</sub> emissions regulations.....	106
Figure 4.19. Total utilities produced by wind turbines and storing units for clustered cases and original cases under CO <sub>2</sub> emissions regulation.....	106
Figure 4.20. Average charging power for each stochastic scenario .....	109
Figure 4.21. Average discharging power for each stochastic scenario.....	109
Figure 5.1. Schematic representation of the proposed data-drive surrogate-based optimization framework.....	116
Figure 5.2. Schematic representation of condensate stabilizer process .....	117
Figure 5.3. Scatter matrix plot of ‘plant 1’ hourly data set.....	121
Figure 5.4. Scatter matrix plot of ‘plant 1’ daily data set .....	122
Figure 5.5. Scatter matrix plot of ‘plant 2’ data set .....	124
Figure 5.6. Variation in condensate flow rate for pump A and pump B and the total flow that leaves the column.....	125
Figure 5.7. F-test values for plant 1 features vs target variables .....	133
Figure 5.8. F-test values for plant 2 features vs target variables .....	133
Figure 5.9. Effect of feature selection on plant 1 response variables prediction error using F-test value.....	135
Figure 5.10. Effect of feature selection on plant 1 response variables prediction accuracy (R <sup>2</sup> ) using F-test value .....	135
Figure 5.11. Effect of feature selection on plant 2 response variables prediction error using F-test value.....	136
Figure 5.12. Effect of feature selection on plant 2 response variables prediction accuracy (R <sup>2</sup> ) using F-test value .....	136
Figure 5.13. Mutual information score values for plant 1 features vs target variables.....	138
Figure 5.14. Mutual information score values for plant 2 features vs target variables.....	138
Figure 5.15. Effect of feature selection on plant 1 response variables prediction error using using MI score .....	139

Figure 5.16. Effect of feature selection on plant 1 response variables prediction accuracy ( $R^2$ ) using MI score .....	139
Figure 5.17. Effect of feature selection on plant 2 response variables prediction error using MI score .....	140
Figure 5.18. Effect of feature selection on plant 2 response variables prediction accuracy ( $R^2$ ) using MI score.....	140
Figure 5.19. Validation curves for linear regression models of ‘plant 1’ where (a), (b) and (c) denote Ridge models for RVP, $H_2S$ content and water content respectively, while (d), (e) and (f) represent Lasso models for RVP, $H_2S$ content and wate content respectively .....	144
Figure 5.20. Zoomed validation curve of water content for plant 1 condensate .....	145
Figure 5.21. Validation curves for linear regression models of ‘plant 2’ where (a) and (b) denote Ridge models for RVP, water content respectively, while (c), and (d) represent Lasso models for RVP and water content respectively .....	145
Figure 5.22. Validation surface (cross-validation MSE) as function of SVM regression hypermeters for plant water content prediction model .....	149
Figure 5.23. Fitness value of ‘plant 1’ water content prediction model as a function of generation number .....	150
Figure 5.24. Application of GA on SVM regression hypermeters tuning process .....	151
Figure 5.25. Variation of cross-validation MSE as a function of generation numbers for RVP of ‘plant 1’ prediction model.....	152
Figure 5.26. Variation of cross-validation MSE as a function of generation numbers for $H_2S$ content of ‘plant 1’ prediction model.....	152
Figure 5.27. GA searching process to find optimal C-value and gamma for RVP of ‘plant 1’ prediction model .....	152
Figure 5.28. GA searching process to find optimal C-value and gamma for $H_2S$ content of ‘plant 1’ prediction model .....	152
Figure 5.29. Variation of cross-validation MSE as a function of generation numbers for RVP of ‘plant 2’ prediction model.....	153
Figure 5.30. Variation of cross-validation MSE as a function of generation numbers for water content of ‘plant 2’ prediction model.....	153

Figure 5.31. GA searching process to find optimal C-value and gamma for RVP of ‘plant 2’ prediction model .....	153
Figure 5.32. GA searching process to find optimal C-value and gamma for water content of ‘plant 2’ prediction model.....	153
Figure 5.33. Cross-validation evaluation of L <sub>1</sub> -norm and L <sub>2</sub> -norm penalty parameters that implemented for ‘plant 1’ water content ANN prediction model.....	157
Figure 5.34. Learning curve for model A .....	157
Figure 5.35. Learning curve for model B .....	158
Figure 5.36. Learning curve for model C .....	158
Figure 5.37. Normalized predictions vs. actual outputs of the linear developed models for plant 1 target variables: (a) Ridge model predicting RVP; (b) Ridge model predicting H <sub>2</sub> S content; (c) Ridge model predicting H <sub>2</sub> O content (d) Lasso model predicting RVP; (e) Lasso model predicting H <sub>2</sub> S content; (f) Lasso model predicting H <sub>2</sub> O content.....	161
Figure 5.38. Normalized predictions vs. actual outputs of the detailed developed models for plant 1 target variables: (a) SVM model predicting RVP; (b) SVM model predicting H <sub>2</sub> S content; (c) SVM predicting H <sub>2</sub> O content (d) ANN model predicting RVP; (e) ANN model predicting H <sub>2</sub> S content; (f) ANN model predicting H <sub>2</sub> O content.....	162
Figure 5.39. Normalized predictions vs. actual outputs of the linear developed models for plant 2 target variables: (a) Ridge model predicting RVP; (b) Ridge predicting H <sub>2</sub> O content (c) Lasso model predicting RVP; Lasso model predicting H <sub>2</sub> O content .....	163
Figure 5.40. Normalized predictions vs. actual outputs of the detailed developed models for plant 2 target variables: (a) SVM model predicting RVP; (b) SVM model predicting H <sub>2</sub> O content; (c) ANN model predicting RVP content (d) ANN model predicting H <sub>2</sub> O content.....	163
Figure 5.41. Effect of feed temperature on condensate RVP and H <sub>2</sub> S content using ANN and SVM regression prediction models.....	164
Figure 5.42. Effect of reboiler temperature on condensate H <sub>2</sub> S content using ANN and SVM regression prediction models .....	164
Figure 5.43. Schematic representation of the proposed optimization framework where machine learning models are integrated with optimization problem constraints .....	170
Figure 5.44. Normalized actual vs ANN predictions of steam flowrate for plant1 condensate stabilizer column.....	176

Figure 5.45. Normalized actual vs ANN predictions of steam flowrate for plant 2 condensate stabilizer column.....	176
Figure 5.46. Learning curve for ANN steam flowrate of ‘plant 1’ prediction model.....	176
Figure 5.47. Learning curve for ANN steam flowrate of ‘plant 2’ prediction model.....	176
Figure 5.48. Normalized actual vs ANN predictions of RVP for plant 1 using actual steam flowrate data (I) and predicted steam data (II).....	177
Figure 5.49. Normalized actual vs ANN predictions of H <sub>2</sub> S content for plant 1 using actual steam flowrate data (I) and predicted steam data (II).....	177
Figure 5.50. Normalized actual vs ANN predictions of RVP for plant 2 using actual steam flowrate data (I) and predicted steam data (II).....	177
Figure 5.51. Normalized actual vs ANN predictions of water content for plant 2 using actual steam flowrate data (I) and predicted steam data (II) .....	177
Figure 5.52. Schematic representation of the proposed optimization framework where machine learning models are integrated with optimization problem constraints and objective function .	179
Figure 5.53. Schematic representation of the proposed optimization framework where machine learning models are integrated with optimization problem constraints and objective function under different initial guess .....	184

## List of Tables

Table 2.1. Productivities (oil, gasoline), and demand of gasoline for the motivating optimization problem .....	8
Table 2.2. Objective function and solution quality for different mathematical formulations. ....	14
Table 2.3. Selected types of commonly used neural networks activation function .....	31
Table 3.1. Data for the thermal generating unit [86] .....	45
Table 3.2. Wind turbine and thermal (conventional) generating unit capital cost and carbon emissions factors .....	49
Table 3.3. Objective function and design decision results for the deterministic formulation .....	50
Table 3.4. Comparison between deterministic, stochastic and worst-case solution of power generation model without environmental consideration.....	62
Table 3.5. Comparison between stochastic solution with external electricity supply and worst-case scenario objective function of power generation model without environmental consideration .....	62
Table 3.6. Comparison between deterministic and stochastic solution of power generation model with environmental consideration.....	65
Table 3.7. Comparison between full-size and reduced size solution of stochastic power generation planning model in day-ahead and real time stages .....	66
Table 4.1 Literature review summary on energy hubs optimization problems .....	73
Table 4.2. Attributes weight factors for the multi-objective function (overall clustering similarity measure) .....	81
Table 4.3 Computational performance of heuristic and general formulation clustering approaches .....	82
Table 4.4. Solution time of heuristic clustering approach under different runs .....	83
Table 4.5. Summary of relative error statistics for normal and sequence clustering using weight factor 0.5 (365 days-4,5 and 6 clusters) .....	84
Table 4.6. Technical and economic information of energy conversion and storing technologies	90
Table 4.7. Values of objective function for the RP, EV and EEV problems.....	108
Table 5.1. List of input and output variables of plant 1 .....	119
Table 5.2. Summary of ‘plant 1’ hourly raw data set .....	119
Table 5.3. Summary of ‘plant 1’ daily raw data set.....	120

Table 5.4. List of input and output variables of plant 2.....	122
Table 5.5. Summary of ‘plant 2’ raw data set.....	123
Table 5.6. Outlier methods performance comparison for ‘plant 1’ hourly data set.....	128
Table 5.7. Outlier methods performance comparison for ‘plant 1’ daily data set .....	128
Table 5.8. Outlier methods performance comparison for ‘plant 2’ data set .....	128
Table 5.9. Summary of ‘plant 1’ cleaned hourly data set .....	129
Table 5.10. Summary of ‘plant 1’ cleaned daily data set.....	130
Table 5.11. Summary of ‘plant 2’ cleaned data set.....	131
Table 5.12. P-values of plant 1 features against response variables .....	134
Table 5.13. P-values of plant 2 features against response variables .....	134
Table 5.14. Linear regression models cross-validation evaluation.....	143
Table 5.15. Comparison between cross-validation strategy and GA in finding best fit SVM hyperparameters .....	149
Table 5.16 SVM regression models cross-validation evaluation.....	149
Table 5.17. Comparison between ANN and SVM regression validation prediction performance .....	159
Table 5.18. Trust region constrained method parameters setting .....	171
Table 5.19. Plant 1 optimization result with machine learning prediction models present in constraints only .....	173
Table 5.20. Plant 2 optimization result with machine learning prediction models present in constraints only .....	174
Table 5.21. Plant 1 optimization results with ANN models used to predict condensate specification (constraints) and steam flowrate (objective function) .....	182
Table 5.22. Plant 2 optimization results with ANN models used to predict condensate specification (constraints) and steam flowrate (objective function) .....	183
Table 5.23. The optimal operating conditions of plant 1 over all initial guess.....	185
Table 5.24. The optimal operating conditions of plant 2 over all initial guess.....	185

## List of Abbreviations

ANN	Artificial Neural Network
CHP	Combined Heat and Power unit
DER	Distributed Energy Resource
DRER	Distributed Renewable Energy Resource
EEV	Expected result of using the EV solution
Elyzr	Electrolyzer
ESS	Energy Storage System
EV	Expected Value
IF	Isolation Forest
IQR	Inter Quartile Range
LOF	Local Outlier Factor
MI	Mutual Information
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Non-Linear Programming
MSE	Mean Square Error
NG	Natural Gas
NLP	Nonlinear programming
OCSVM	One Class Support Vector Machine
ReLU	Rectified Linear Unit activation function
RP	Recourse Problem
SDG	Stochastic Gradient Descent
SQP	Sequential Quadratic Programming
SVM	Support Vector Machine
TRCM	Trust-region Constrained Method
TRM	Trust-region method
UC	Unit Commitment
VSS	Value of Stochastic Solution
WT	Wind Turbine



## List of Symbols

List of symbols for the power generation planning model.

### Indices

$t$	time period
$i$	power generating units
$s$	stochastic scenarios

### Discrete variables

$u_{i,t}$	binary operational/scheduling decision variable representing the on/off status of unit $i$ at period $t$
$x_i$	binary design decision variable representing whether unit $i$ should be installed or not
$y_{wind}$	integer design decision variable representing the number of wind turbine needed to install

### Continuous variables

$p_{i,t}$	power output variable of unit $i$ at period $t$ ;
$p_{wind,t}$	power output variable of all wind turbines at period $t$ ;
$c_{i,t}^{su}$	start-up cost variable of unit $i$ at period $t$ ;
$c_{i,t}^{sd}$	shut-down cost variable of unit $i$ at period $t$

### Parameters

$A_i, B_i$	coefficients of the fuel cost function of unit $i$ , their values are listed in Table 3.1
$C_i^g$	the capital cost of $i$ generating unit (Table 3.2)
$C_{wind}$	the capital cost of one wind turbine (Table 3.2)
$N_d$	number of operating days per year
$N_{life}$	lifetime of power generation plant (years)
$Prob_s$	probability of each stochastic realization scenario $s$
$P_i^U$	upper power generating limit of unit $i$ (MW)
$P_i^L$	lower power generating limit of unit $i$ (MW)
$\beta_i$	the emission factor associated with thermal power unit $i$ , (kg-CO <sub>2</sub> eq./MWh)
$T^{cold}$	the cold start hour of unit $i$ , (h)
$TU_i$	minimum up time for unit $i$ (h)
$TD_i$	minimum down time for unit $i$ (h)
$T_i^{ini}$	denotes the number of periods that unit $i$ has been initially offline ( $T_i^{ini} < 0$ ) or online ( $T_i^{ini} > 0$ ) (h)
$RU_i$	ramp-up rate of unit $i$ (MW/h)
$RD_i$	ramp-down rate of unit $i$ (MW/h)
$Hsc_i$	hot start up cost for unit $i$ (\$)
$Csc_i$	cold start up cost for unit $i$ (\$)

## List of symbols for the energy hub planning model

Indices	
$a$	attributes (in this study either heat or electricity (elec))
$c$	clusters
$d$	days, and $D$ is the total number of days
$h$	hours
$i$	energy carrier
$n$	index of the number of initial guess cluster scenarios and the total is $N$
$s$	stochastic scenarios
$st$	index for storing units storing units set $\{Elyzr, Fuelcell, Tank\}$
$u$	index for conventional energy production units set $\{CHP1, CHP2, CHP3, boiler1, boiler2, boiler3\}$
$wt$	index for wind turbine set $\{1,2\}$ where 1 is Vergent (20 kW) and 2 is Fuhrlander (30 kW)
Discrete variables	
$x_{d,c}$	binary variable allocating loads for day $d$ joining cluster $c$
$y$	integer design variable that represents the number of each unit needed to be installed
$ch_{d,h,s}$	binary variables that represent the on and off states of electrolyzer units at each $s$ scenario and $h$ hour of the $d$ day
$dis_{d,h,s}$	binary variables that represent the on and off states of fuel cell units at each $s$ scenario and $h$ hour of the $d$ day
Continuous variables	
$AD_{a,d,h}$	absolute difference between load curve $l$ and clustered curve $c$ for hour $h$ in day $d$ for attribute $a$
$IAE_a$	the integral absolute error ( $L_1$ -norm) used as similarity measure for the ( $a$ ) attribute
$H_{d,h,s}^{Elyzr}$	mass flow rate of hydrogen gas produced by electrolyzer at $s$ scenario and $h$ hour of the $d$ day
$H_{d,h,s}^{Tank}$	mass flow rate of hydrogen gas leaving the hydrogen tank at $s$ scenario and $h$ hour of the $d$ day
$HL_{h,d}$	amount of hydrogen stored in the hydrogen tank at $h$ hour of the $d$ day
$D_{a,c,h}$	the representative demand of attribute $a$ for hour $h$ hour in cluster $c$ .
$P_{i,d,h,s}$	the operational decision variable that represents the amount of energy flow ( $i$ denote the type of energy heat or electricity) consumed or produced by each energy hub unit at $s$ scenario and $h$ hour of the $d$ day.
$P_s^{wt}$	the operational decision variable that represents the amount of power produced by all wind turbines under each scenario $s$ flow
$NG_{d,h,s}^u$	amount of natural gas consumed by conventional units $u$ at $s$ scenario and $h$ hour of the $d$ day

## Parameters

$k$	shape parameter of Weibull distribution
$c$	scale parameter of Weibull distribution
$CAP$	capital cost (\$)
$OM$	operational and maintenance cost parameters
$\eta_i$	thermodynamic efficiency of converted utilities $i$ produced by energy hub units
$\beta_s$	probability of each stochastic realization scenario $s$
$z_{rated}$	rating capacity of each energy hub unit (Table 4.6).
$\gamma_d$	number of repetitions (frequency) for corresponding $d$ day
$L_{elec,d,h}$	hourly electricity demand
$L_{heat,d,h}$	hourly heat demand
$W_a$	attribute $a$ 's weighting factor ( $W_a \geq 0, \sum_a W_a = 1$ ).

# Chapter 1 Introduction

## 1.1 Project Motivations

Optimization or mathematical programming is extensively used in many strategic decision-making problems[1]. It is a core concept within process systems engineering and operations research. It can help guide the decision maker over which strategy and operational conditions to apply in order to minimize the overall cost or maximize the profit while satisfying problem constraints. Its application has proven its superior performance by increasing the profits while maintaining customer or/and decision maker satisfaction [2]. Typical applications of optimization can be found in engineering, transportation, production, operational research, supply chain management and many other fields.

Process optimization under deterministic conditions can lead to solutions for only certain process parameters (e.g., fixed fuel price, fixed power demand profile and fixed feedstock price). However, most real-life problems include some sort of uncertainty in which deterministic models are incapable of solving them or give unpractical solutions that are optimal only under certain conditions. Many model parameters are uncertain and challenging to predict in real life, such as the availability of renewable energy. Perfect information that includes assigning probability distributions to the random variables (uncertain parameters), is one typical way to tackle the decision-making problem. Another traditional way to tackle this problem, is to feed the optimization model under uncertainty with large number of possible uncertain scenarios. However, prior knowledge on uncertain parameter distribution is usually unknown and requires extensive effort to employ, as well as, using a very large number of scenarios is computationally impractical [3]. The availability of massive amounts of data and the recent advances of data analytics tools such as machine learning, encourages the implementation of these methods in optimization problems under uncertainty. Therefore, one goal of this project is to investigate how important information from real-life available data can be captured through the advances of data analytics tools and applied to optimization problems under uncertainty. More specifically, the focus will be on proposing a simple data-driven approach for power generation planning models that incorporate intermittent renewable energy sources.

Conventionally, modelling approaches focus on a mono-scale perspective. When macroscale behaviour of a system is the focal point, the microscale is modelled using constitutive relations. On the other hand, if the subject matter is the microscale, it's assumed nothing compelling occurs at a macroscale level, and larger scales have a homogenous process. However, it is quite challenging to extend such simple empirical methods to more complex systems. The need to tackle the restrictions of both aforementioned approaches (macro- and micro-scale) is the reason for implementing multiscale modelling approach. Therefore, multiscale approach targets simultaneously the efficiency of macroscale models while preserving the microscale's models precision. A more comprehensive modelling approach can be achieved when the problem is evaluated from different scales and levels perspective at the same time [4]. The integration of planning (e.g., design) and scheduling (e.g., operation) is an example of a multiscale model. The integration of different decision level improves decision level management which results in lower net cost. Yet, large-scale problems are formed as a result of different time scales integration that are typically computationally intractable. The design and operation of energy hubs faces similar challenges. The multiscale (i.e., planning and scheduling) energy hub systems that incorporate renewable energy resources become more challenging to be modelled due to a high level of intermittency associated with renewable energy. To tackle this problem different modelling and solution approaches have been proposed. Clustering has shown potential as an appropriate data-driven solution approach to deal with such problems. Similar input parameters (e.g., demand or price) that exhibit similar trends are aggregated using clustering. Accordingly, clustering can serve as an effective tool to reduce model size and enhance computational tractability while maintaining acceptable solution accuracy. Therefore, in this work, the application of clustering approach to multiscale energy hub planning and operation model under intermittent wind energy is investigated.

Modelling of processing facilities tailored for the production of specific chemical products from a specific set of raw materials is considered to be one of the fundamental problems in chemical and process systems engineering. These processes aim to perform specific physicochemical transformations, in an economically profitable way while satisfying production requirements and several other constraints including raw material availability, operational safety, environmental regulations etc. However, modelling these processes involve many complex unit equation blocks which can be solved using conversion laws or physicochemical engineering fundamentals and

available simulation software [5]. Despite the significant developments in realistic unit operation models (i.e., the kind of models featured in commercial process simulators considering non-ideal thermodynamics, kinetics, and transport properties calculations) and the availability of commercial process simulators [6] (e.g., ASPEN [7], HYSYS, ProMax [8], gProms [9]); modelling them based on detailed realistic unit operation equations require significant effort and are computationally expensive to solve. It would be even more complicated to solve these detailed models when they are combined with optimization routine [10], [11]. On the other hand, commercial simulation software can obtain accurate results, nonetheless, these commercial software are not open-source, plus combining them with optimization models are challenging [12]. At the same time, recent advancement in technology have allowed the industry to collect and store massive amounts of data from their processes [13]. Data-driven surrogate modelling can be defined as a black-box modelling approach that can utilize available data. It can relate relevant inputs to relevant outputs to describe process operations. Such models have been used in industry and literature to describe processes by replacing existing expensive models (serve as surrogates to reduce model complexity) and correlations which have not yet been theoretically explained [14]. Given the existence of vast amounts of data and the recent developments of data analytics tools, such as machine learning methods, and the need for reduced order models which can relate relevant inputs to relevant model outputs to represent process operations. Therefore, the role of data-driven surrogate modelling can be extremely valuable. Another reason that motivates this research to use the big data analytics tools (i.e., machine learning), is that these tools have proven their ability to generate accurate and computationally efficient surrogate or reduced order models [13]. Thus, applying the data-driven surrogate modelling approach in an optimization framework will reduce the mathematical complexity of the entire optimization framework model and impose a suitable mathematical representation, which can be solved numerically by current state-of-art numerical solvers [5]. Thus, in this project, data-driven surrogate-based optimization framework is proposed. In this approach, we will leverage so called machine learning tools into process optimization by replacing the unit operation's detailed model with a surrogate reduced order data-driven model. In future, with promising research, the proposed methods and frameworks can be applied to different energy infrastructure and chemical process systems where these systems can gain the full benefits from existing information.

## 1.2 Project Goals and Contributions

Considering the motivations mentioned above, the main goal of this research is to develop data-driven solutions that can benefit energy infrastructure planning and industrial process operation optimization models, by improving their solutions reliability and computational tractability. In line with this research work, the following are the goals of this study:

- Develop a data-driven stochastic optimization framework that integrates machine learning tools into power generation planning model. As renewable energy availability suffers from intermittency and uncertainty, it is very important to model their uncertainty and determine their behaviour. Unsupervised machine learning algorithm (k-means clustering) is employed to generate uncertainty scenarios from the historical weather and demand data. Accordingly, reduced size uncertainty scenarios, that feature underlying patterns from uncertain parameters, are generated. These scenarios are used as inputs to the stochastic model where the proposed model is formulated as a mixed integer linear programming (MILP).
- Develop multiscale approach to model stochastic energy hub systems under the uncertainty of renewable energy resources. A mathematical programming-based general clustering approach is applied to reduce the size of multiple attributes energy hub demand data. Evaluation of heuristic approach derived from the mathematical programming-based clustering approach to reduce clustering computational time, is carried out. A data-driven statistical method is employed to model the intermittent behaviour of uncertain renewable energy. Following the aforementioned methods, the design and operation (multiscale) of an energy hub with hydrogen storage is reformulated as a two stage stochastic model.
- Develop data-driven surrogate-based optimization framework for chemical process (condensate stabilizer process) based on real plant data. Collected data are undergone cleaning process in which outliers are detected and removed. Using cleaned data, different supervised machine learning models, that describe process operations, are developed to relate inputs to outputs. The predictions from the developed models are validated against actual plant operating data. An optimization framework based on trust-region constraint algorithm is proposed, where the machine learning model with highest prediction accuracy, is integrated as a surrogate model that describe the process.

The main outcome of this study will be different general frameworks that can connect between data-driven approaches and optimal planning and operation of energy infrastructures (e.g., power

generation, energy hub systems) and industrial processes (e.g., condensate stabilization process). Through the implementation of these approaches, different types of intermittent renewable energy resources can be integrated to power generation (or power generation capacity expansion) planning model that involve CO<sub>2</sub> emissions regulations, different energy hub topology with multiple energy carrier demands under different energy resources intermittency at reasonably low computational expenses can be investigated, and the surrogate-based optimization approach can serve as computer-aided software where it can be applied to different industrial process using either actual plant data or simulated data from commercial software's.

High-level optimization tasks such as planning and scheduling can highly benefit from information mined from data, through the proposed data-driven approaches, since optimization has always been based on the interchange between models and data [14]. Graphical representation of the scope of this project is depicted in the following Figure 1.1.

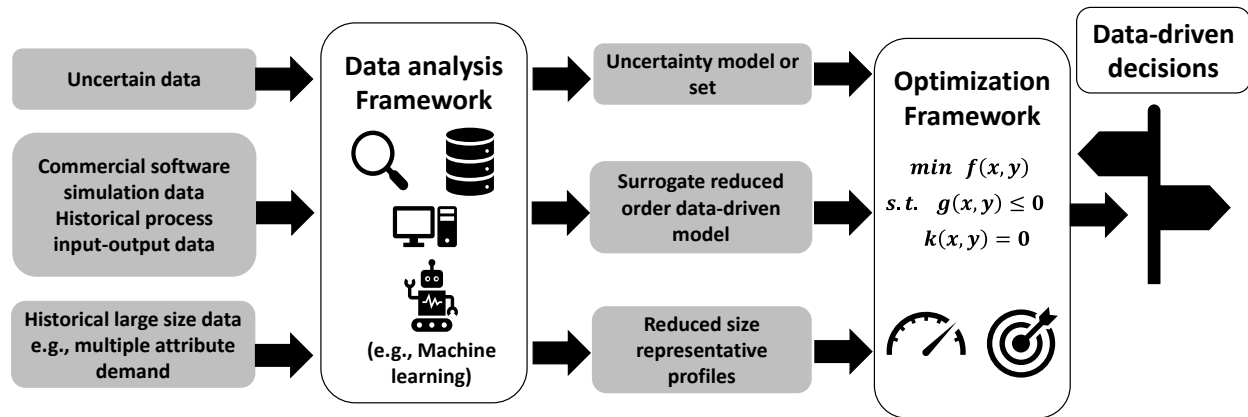


Figure 1.1. Data-driven optimization project scope

### 1.3 Dissertation Outline

Chapter 2 presents the background of mathematical programming techniques and big data tools that are relevant to this project. The background information is also linked to previous research projects that have studied data-driven optimization.

Chapter 3 illustrates a case study application on data-driven optimization. In this chapter, a data-driven optimization approach is used to optimally design and operate a power generation plant under demand and wind energy uncertainty.



Chapter 4 presents a case study where a multiscale clustering approach is applied to a stochastic energy hub model.

Chapter 5 shows a case study that natural gas processing (condensate stabilizer) is modelled, and process optimization is performed based on machine learning approach. In this chapter, surrogate-based optimization framework that leverage data-driven machine learning models for condensate stabilizer is generated.

Chapter 6 includes the dissertation's concluding remarks and potential future works.

## Chapter 2 Background and Literature Review

This chapter is divided into three main sections namely: 1) background on mathematical programming methods, 2) data analytics tools that will be used in the current project and 3) a literature review on data-driven optimization.

The objective of this chapter is to give the reader an overview on what has been done so far in this field. It will demonstrate different mathematical programming formulations used for process optimization. It will also give an overview of the recent advances of data analytics tools and it will describe some of the main machine learning algorithms that will be used in this project. Finally, the various contributions related to the topic of the thesis will be presented.

### 2.1 Mathematical Optimization Methods

Mixed integer programming problem (MIP) is broadly used in chemical design and process engineering. Typical applications include superstructure modelling, allocation problems, scheduling problems, and so on. A mixed integer programming problem (MIP) is an optimization model that has both integer and continuous variables. Conventionally, the integer variables in typical process system engineering applications refer to the binary variable (0-1 variables/ e.g., develop – not to develop).

$$\begin{aligned} \min C &= f(x, y) \\ \text{s. t. } g(x, y) &\leq 0 \\ k(x, y) &= 0 \\ y &\in \{0,1\}^m \end{aligned} \tag{2.1}$$

Where the objective function  $C = f(x, y)$  in general, represents a desired economic or environmental measure, while the equality and inequality constraints  $k(x, y)$  and  $g(x, y)$  are imposed to satisfy unit equations (e.g., thermodynamics, mass and energy balances) design equations, physical constraints, design specifications or logical conditions. Continuous variables, (x) can be attributed to power, flowrates, equipment sizes, pressures and temperatures; whereas 0-1 binary variables (y) can be attributed to the existence of units (to be developed or not, design decisions), scheduling (assignment to units to tasks and time periods), whether units are operating or not (online or offline, operational decision) at each time period.

For this thesis, the General Algebraic Modelling System (GAMS) [15] is used to formulate and solve the mathematical programming models. GAMS is one of the leading commercial modelling systems for mathematical programming and optimization framework. It enjoys simple programming syntax and a wide range of integrated solvers that can be called based on the mathematical programming type. GAMS has the ability to deal with complex and large scale modelling applications[15]. The existing literature has a large number of studies on the optimal design and operation of energy and process systems models that have been formulated and solved in GAMS [16].

Typically, mathematical programming models can be validated using two approaches namely validation by construction and validation by results. Validation by construction relies on procedure believed to appropriate by the model builder. This approach involves the modelling the problem based on experience and theory and the specification of the problem data are either based on scientific reasonable estimation or based on real world observation. On other hand validation by result involves consists of comparing the model results with corresponding real-world outcomes[17]. The validation by construction methodology was followed to validate and construct the mathematical programming models in this study.

### 2.1.1 Deterministic Approach

In this section, we will demonstrate different mathematical programming approaches for handling uncertainty through a simple example optimization problem [18]. The example was adopted from [18] and modified for the sake of this study. Assume that we have two types of oil namely national (oil<sub>1</sub>) and imported oil (oil<sub>2</sub>), two types of gasoline will be produced (standard and premium) The output of gasoline per unit of the raw oil (productivity) and the demands for each type of gasoline (d<sub>gas1</sub>, d<sub>gas2</sub>) are also shown in Table 2.1. The unit costs of the raw oil materials are (2 unit of currency per unit mass for oil<sub>1</sub> and 3 for oil<sub>2</sub>). The maximum total amount of oil that can be processed in the plant is 100. According to the given information, the problem can be formulated as a linear programming model.

Table 2.1. Productivities (oil, gasoline), and demand of gasoline for the motivating optimization problem

	Standard gasoline	Premium gasoline
Oil <sub>1</sub>	2	3

Oil <sub>2</sub>	6	3
Demand (d)	180	162

The deterministic formulation of this problem is presented in the following equations (equations (2.1) to (2.6))

*Objective function:*

$$\min Cost = 2x_{oil1} + 3x_{oil2} \quad (2.2)$$

*Plant capacity constraint*

$$s. t. \quad x_{oil1} + x_{oil2} \leq 100 \quad (2.3)$$

*Demand constraints*

$$2x_{oil1} + 6x_{oil2} \geq 180 \quad (2.4)$$

$$3x_{oil1} + 3x_{oil2} \geq 162 \quad (2.5)$$

*Positive variable constraints*

$$x_{oil1} \text{ and } x_{oil2} \geq 0 \quad (2.6)$$

The optimal solution for the above problem is given as follows:

$$x_{oil1} = 36, x_{oil2} = 18, cost = 126$$

In the prior case the productivities and demands are assumed to be fixed and known to the decision maker before deciding on the production plan. The solution of this preliminary optimization problem is called the *Deterministic solution* (it is also called optimal on average). However, this is obviously not always the case. Most of the time, given data are not certain (i.e., demand and productivity), they vary within certain limits, and that the decision should be made on the production plan before knowing the exact values of the data. For the sake of this demonstration let's be more specific. Assume that the demand of standard and premium gasoline ( $d_{standard}$   $d_{premium}$ ) are varying randomly. The two demands are represented by the two following random variables, respectively,  $\xi_{standard}$  and  $\xi_{premium}$  with normal distributions, i.e.,

$$dist \xi_{standard} \sim \mathcal{N}(180,12)$$

$$dist \xi_{premium} \sim \mathcal{N}(162,9)$$

We assume that these two random variables are independent. The two demands' random variables are denoted by  $\xi_{stand}$ ,  $\xi_{premium}$ . Also, it is assumed that they are restricted by 99% confidence

intervals, respectively. The 99% confidence intervals bound for the two random variables are shown as follows:

$$\xi_{standard} \in [149.09, 210.91]$$

$$\xi_{premium} \in [138.82, 185.18]$$

### 2.1.2 Worst-Case Approach

One possible solution against uncertainty issues in the demand is to look for a solution that is “safe” and can satisfy all possible realizations of the demand. This solution should be feasible for all possible realizations of the demands. This approach is called the Worst-Case Scenario. This approach is conservative and does not take any risk (very safe) [19]. Let’s assume that the demand random variables are approximated by K scenario (i.e., each scenario derived from the normal distribution within 99% of its confidence interval). The demand constraint can be re-written as follows:

$$2x_{oil1} + 6x_{oil2} \geq \xi_{standard}^k, \quad k \in K \tag{2.7}$$

$$3x_{oil1} + 3x_{oil2} \geq \xi_{premium}^k, \quad k \in K \tag{2.8}$$

We need to solve for all possible k scenarios, however it’s clear that the solution of the extreme point will be feasible for all other realizations. Therefore, the solution of the Worst-Case Scenario can be obtained by solving the refinery problem when the demand is maximum. The demand constraints (2.7) and (2.8) can be written as follows

$$2x_{oil1} + 6x_{oil2} \geq 210.91$$

$$3x_{oil1} + 3x_{oil2} \geq 185.18$$

The solution for the above worst-case scenario approach problem is shown below:

$$x_{oil1} = 40, \quad x_{oil2} = 21, \quad cost = 145$$

However, enquiring feasibility for any future realization of uncertainty can be too restrictive. Extreme rare events may exist depending on the data and they can make the almost feasible set empty (leads to an infeasible [19], [20] programming is the alternative method that can be used to overcome these restrictions [18].

### 2.1.3 Model with Chance (Probabilistic) Constraint

Instead of solving for the Worst-Case, we will solve for the points that are with only some probability [18], [19] approach also called the Chance Constraint or Probabilistic Constraint. In

this approach, the problem is solved with some sort of risk introduced by the decision maker. The illustration of this method is applied to the refinery example. Therefore, we will solve the probability of the demand constraint to be within a certain level of acceptability  $(1-\varepsilon)$  where  $\varepsilon$  is the risk probability. Therefore, we will find a solution for the refinery problem that can satisfy the following probabilities of the demand constraints:

$$Pr(2x_{oil1} + 6x_{oil2} \geq \xi_{standard}) \geq 1 - \varepsilon$$

$$pr(3x_{oil1} + 3x_{oil2} \geq \xi_{premium}) \geq 1 - \varepsilon$$

Where  $Pr$  denotes the probability of that constraint. These probabilities are called *Individual/ Separate Chance Constraints*. Since  $\xi_{stand}$  and  $\xi_{premium}$  are random variables that follow the normal distribution, the above probabilities which use the inverse cumulative distribution (CDF) can be calculated as follows [19], [20]:

$$2x_{oil1} + 6x_{oil2} \geq F_{\xi_{standard}}^{-1}(1 - \varepsilon) \quad (2.9)$$

$$3x_{oil1} + 3x_{oil2} \geq F_{\xi_{premium}}^{-1}(1 - \varepsilon) \quad (2.10)$$

Where  $F_{\xi}^{-1}$  is the inverse of the CDF (cumulative distribution function) of the random variable  $\xi$ . Since  $\xi_{stand}$  and  $\xi_{premium}$  are scalar, we can have an expression for the generalized CDF inverse as follows [19]:

$$F_{\xi_i}^{-1}(1 - \varepsilon) = \tilde{\xi}_i + \phi^{-1}(1 - \varepsilon)\sigma_i \quad (2.11)$$

Where  $\xi_i$  denotes the random variable mean,  $\phi^{-1}$  is the standard inverse cumulative distribution function ( $N(0,1)$ ) and  $\sigma_i$  is the standard deviation. Hence, using this formula (2.11) the demand chance constraints (equations (2.9) & (2.10)) can be written as follows:

$$\begin{aligned} 2x_{oil1} + 6x_{oil2} &\geq \widetilde{\xi_{standard}} + \phi^{-1}(1 - \varepsilon)\sigma_{standard} \\ 3x_{oil1} + 3x_{oil2} &\geq \widetilde{\xi_{premium}} + \phi^{-1}(1 - \varepsilon)\sigma_{premium} \end{aligned}$$

The right-hand side of the above equation can be calculated, and the demand constraints will be as follows:

$$\begin{aligned} 2x_{oil1} + 6x_{oil2} &\geq 203.52 \\ 3x_{oil1} + 3x_{oil2} &\geq 179.64 \end{aligned}$$

The solution of the refinery problem with *Chance Constraint* for the demand is as follows:

$$x_{oil1} = 38.4007, \quad x_{oil2} = 20.940, \quad cost = 140.7$$

The chance constraint is concerned with not violating the feasibility [14].

#### 2.1.4 Model with Recourse

In this case, the model is defined for the extreme events that do not constrain the “almost sure” feasible points significantly. In this approach, the model is formulated with a recourse variable. Recourse variables represent the amount of penalty (correction) after observing the realization of uncertainty. These variables are called wait-and-see variables. On the other hand, the variables that are decided before the uncertainties are realized are called here-and-now variables [18]–[20]. A typical model of stochastic optimization with recourse is the Two Stage Stochastic Programming With Recourse [20]. At the first stage, certain decisions (i.e., here and now) are made before the realization of uncertainty, whilst at the second stage corrective actions (wait and see) are taken after uncertainties are revealed. The second stage variables work as corrective action to avoid infeasibility when random events have presented themselves. For instance, in an energy system process and design, the first stage variables can be represented by design decisions and the second stage variables can be represented by operational level decisions.

In the refinery example, the problem is modelled as a two stage stochastic program. The refinery company will pay a penalty if they do not satisfy the market demand. They will have to buy the amount of gasoline that is in shortage and supply the market demand. The amount of gasoline that will be imported by the company when there is a shortage in their production (i.e., can't meet the market demand) is denoted by  $(y_{\text{standard}} \text{ and } y_{\text{premium}})$ . These variables (i.e.,  $y$ ) are the recourse variables and they are functions of the realization of our random variable (i.e., uncertain demand). It is assumed that the cost of the imported gasolines per unit are (7 for standard and 12 for premium). In this type of problem, it is a common practice to assume that the random variables have a finite discrete distribution. The reason being that when a continuous distribution is assumed, there will be an evaluation of the expected value which appears in the objective. The evaluation of the expected value of continuous distribution requires multivariate numerical integration; and an implicit definition of the recourse variable/function (i.e., as they are a function of the random variable having the continuous distribution). Therefore, the problem will be highly nonlinear and intractable. More details on this can be found in [18]. The normal distribution of the random variable is approximated by a discrete distribution. A naïve sampling method was used to generate scenarios for the refinery problem [21]. The two stage stochastic programs can be naturally reformulated into an equivalent single-level optimization problem [20]. Accordingly, the former deterministic refinery problem can be formulated as follows (equations (2.12) - (2.14)):

$$\min \left\{ \underbrace{2x_{oil1} + 3x_{oil2}}_{\text{First stage obj}} + \underbrace{\sum_{i=1}^K p_i [7y_{standrad,i} + 12y_{premium,i}]}_{\text{Second stage obj}} \right\} \quad (2.12)$$

$$\begin{aligned} s.t. \quad & x_{oil1} + x_{oil2} \leq 100 \\ & 2x_{oil1} + 6x_{oil2} + y_{standrad,i} \geq \xi_{standrad,i}, i = 1, \dots, K \end{aligned} \quad (2.13)$$

$$3x_{oil1} + 3x_{oil2} + y_{premium,i} \geq \xi_{premium,i}, i = 1, \dots, K \quad (2.14)$$

$$x_{oil1} \text{ and } x_{oil2} \geq 0 \quad (2.15)$$

Where the subscript  $i$  represents the index for the total number of scenarios  $K$ . In order to solve this problem, 20 realizations were drawn for each demand ( $\xi_{stand}$  and  $\xi_{premium}$ ). A total of 400 scenarios were considered ( $K = 20^2$ ). All the case study problems are solved on GAMS 24.5 and CPLEX [15] was selected as a solver. There were 803 equations (400 for each demand constraint and one objective and one capacity constraint) and 802 variables (2 first stage variables and 800 second stage recourse/second stage variables). The optimal solution of this problem is reported as follows:

$$\bar{x}_{oil1} = 37.352, \quad \bar{x}_{oil2} = 20.897, \quad cost = 139.85$$

The first stage cost is 137.33 unit cost.

The quality of each solution can be assessed by defining reliability. Reliability is the probability of constraints that are subjected to uncertain random variables to be feasible [18] (in the case of the refinery example it's the probability of the solution substituted in the demand constrains), see following equation (2.16):

$$pr \left( \begin{array}{l} 2\bar{x}_{oil1} + 6\bar{x}_{oil2} \geq \xi_{standrad} \\ 3\bar{x}_{oil1} + 3\bar{x}_{oil2} \geq \xi_{premium} \end{array} \right) \quad (2.16)$$

As it was assumed that the demand is normally distributed and the answer can be calculated by employing Python function (multivariate normal from SciPy package [22]). Table 2.2 shows the assessment of the solution quality for different mathematical formulation.



Table 2.2. Objective function and solution quality for different mathematical formulations.

Model	$\bar{x}_{oil1}$	$\bar{x}_{oil2}$	First stage cost	Feasibility/ Reliability
Deterministic	36.0	18.0	126.0	0.25
Worst-Case	40.0	21.0	145.0	0.99
Stochastic with recourse	37.6	20.8	137.3	0.88

As it can be noticed from this Table 2.2 the deterministic solution gives the best cost but a less reliable decision. The worst-case is expensive, however, it gives the best reliability. On the other hand, the solution of the two stage stochastic approach with recourse is not too expensive with a reasonable probability of being feasible. From this discussion and example, we can see the advantages of modelling using two stage stochastic programming especially for process and design of energy systems. As it is required to obtain less expensive decision with some sort of satisfactory reliability, for instance, design and operation of a power plant with uncertain demand and wind data, multiscale energy hub modelling under uncertain wind energy, capacity expansion with uncertain fuel price and demand and retrofitting current power system by adding renewable energy generation units (renewable energy availability is one the most uncertain parameters that is needed to be modelled). In this research the stochastic programming with recourse is one of the most essential tools that is used in this project as can be seen in Chapter 3 and Chapter 4.

As we have seen for the stochastic solution, it was assumed that the uncertain parameter follows some known distribution. Nevertheless, in real life the distribution and the bounds of uncertain data are unknown, and here the role of data analytics tools becomes more apparent. The recent advances in data analytics provide very powerful and efficient tools in determining patterns and discovering interesting structure from real historical data (e.g., demand, solar intensity, natural gas supply). Therefore, one objective of this research is to use these tools to recognize/learn from historical uncertain data (with unknown distribution) and draw conclusions that can be used as an input for the optimization framework, that is needed to make high level planning and strategic decisions.

## 2.2 Big Data Tools

Data Science is the art of mining knowledge and driving conclusions from data. It is an interdisciplinary field that uses different scientific methods, algorithms and processes to extract

insights from diverse data sources [23]. Statistics, data analysis, machine learning and their related methods are integrated to form the general concept of data science[24]. It employs techniques and theories drawn from many fields within the context of mathematics, statistics, information science, and computer science. Data analysis is the process of applying said data science concepts [21], [22]. The purpose of data analysis is to discover useful information, derive conclusions, and support decision-making processes from data [22], [23].

The data models that are typical of traditional data analytics are often static and of limited use in addressing fast-changing and unstructured data. The advances in machine learning drive the data analytics tools to evolve tremendously. Machine learning, which has become a major branch in computer science and artificial intelligence [28], [29], is a method of data analysis used to design a model to learn the trends, findings and dependence between attributes and target variables without programming explicitly [30]–[32]. With the recent progress of the internet, smart and wireless sensors, wireless communications, mobile devices, smart devices, e-commerce, and smart manufacturing; the amount of data gathered and stored has grown exponentially. The need for automated methods for data analysis has emerged as a key driver for industry [30], [32]. The goal of machine learning is to develop methods that can automatically (i.e., without programming explicitly) detect patterns in data, and then to use the uncovered patterns to predict future data or other conclusions of interest [30]. Machine learning has recently become one of the most popular technology, motivated by well-publicized advancements like deep learning and the extensive commercial interest in big data analytics [33]. In the last two decades, our lifestyle, the way we live and do business, has been transformed by generating many petabytes of data, because of the Internet. Currently, machine learning, and big data analytics are playing a significant role in revolutionizing our society again by translating that data into useful predictions and decisions [33] Recommendation engines, speech and handwriting recognition systems, content identification, image classification/retrieval, automatic captioning, spam filters, and demand forecasting are examples of commercial applications that are based on machine learning and big data analytics tools [34]. Recent statistical machine-learning development enjoys the following attractive features [33]:

- (1) The ability to extract knowledge from data, whereas traditional methods focus on making the machine learn.

(2) Applying traditional data analysis becomes impossible when data sets are large and heterogeneous as it is always characterized by trial and error. However, machine learning is proposing clever alternatives to analyse huge volumes of data through fast, efficient algorithms and data-driven models also establish data analysis as a theoretical basis in statistics as a discipline to control errors in inference.

(3) The traditional data analysis tools emphasize the cleanliness of the data to prevent potential misleading conclusions, while big data analytics can deal with data errors or messiness and use the massive amount of data to develop models and extract features that are robust to the imperfections.

(4) It is data-driven and target-driven and enjoy new contributions from information industry sectors.

Generally, machine learning can be categorized as unsupervised learning and supervised learning and reinforcement learning [34].

The supervised or the predictive learning uses given labelled sets ( $X$  and  $y$ ) of input-output pairs to learn by mapping from inputs  $X$  to outputs  $y$  (i.e., predict their relationship  $P(y|X)$ ), which used in the classification and [30], [33]. In other words, supervised learning algorithm uses training data to learn a function (model) that generate the desired output. The training set contains inputs and correct outputs, which will allow the model to measure error between predicted output and actual correct output. The function that is used to measure the error is called loss function/ cost function. The task of any supervised algorithm is to minimize the cost/ loss function by adjusting the model. Typically input observations can be referred as features, predictors or independent variables while the output observation (instance) can be called as response, performance, target variables or labels. Examples of regressions methods include, Simple Linear, Generalized Linear, Multi-Linear Regression, Non-Linear Regression, Gaussian Processes Regression (e.g., kriging) and Support Vector machines (SVM) regression. Examples of classification includes Naïve Bayes, Decision Tree, Logistic Regression, SVM and k-Nearest Neighbour, Bayesian Network and Artificial Neural Network (ANN) [30],[31].

Unsupervised learning, in which the training data consists of a set of input vectors  $X$  without any corresponding target values, involves the analysis of unlabelled data under assumptions about structural properties of the data (e.g., algebraic, combinatorial, or probabilistic) [35]. The goal of unsupervised learning is to discover patterns (interesting structure) and identify commonalities. Additionally, unsupervised methods can be used to automatically detect outliers and anomalies in

data sets; therefore, it can be applied as a pre-processing step for supervised machine learning model development as we will see in Chapter 5. Unsupervised methods involve, clustering (i.e., discover groups of similar patterns), density estimation (i.e., determine the distribution of data within the input/feature space, such as kernel density estimation) and dimensionality space reduction of data (e.g., Principal Component Analysis (PCA)) [34].

Reinforcement learning is the process of learning how to act or behave when given occasional rewards or punishment signals [30]. It can combine the learning and acting phases at the same time to online learning and provides a self-optimizing feature [33].

The following are the descriptions of unsupervised learning algorithm that is related to the case study presented in Chapter 3. After that, a background on supervised machine learning methods that are used in Chapter 5 is presented.

## 2.3 Unsupervised Machine Learning Methods

### 2.3.1 K-Means Clustering

The algorithm starts by considering the problem of identifying clusters of data points in a multidimensional space. Suppose there is a data set  $X = \{x_1, \dots, x_N\}$  consisting of  $N$  observations of a random  $d$ -dimensional Euclidean variable,  $x$ , (i.e., the distance between pairs of points in Euclidean spaces). The objective of this algorithm is to separate the data set into some number  $K$  of clusters, for a given value of  $K$ . Intuitively, a cluster can be defined as a group of data points whose inter-point distances are small compared with the distances to points outside of the cluster [30], [31]. The K-means algorithm can be summarized as follows [31], [36].

1. Randomly choose an initial  $k$  centres  $\{\mu_1, \mu_2, \mu_3, \dots, \mu_K\}$ .
2. For each  $k \in \{1, 2, \dots, K\}$ , set the cluster  $C_k$  to be the subset of points in  $X$  (where  $X$  is set of  $N$  data set  $X = \{x_1, \dots, x_N\}$ ) that are closer to  $C_k$  than they are to  $C_j$  for all  $k \neq j$ .
3. For each  $k \in \{1, 2, \dots, K\}$ , set  $\mu_k$  to be the center of mass of all points in  $C_k$ :  $\mu_k = \frac{1}{|C_k|} \sum_{x \in C_k} x$
4. Repeat Steps 2 and 3 until  $\mu_k$  no longer changes.

The K-means algorithm aims to choose  $k$  centers (centroids) that minimize the inertia (see equation (2.17)) given an integer  $K$  and a set of  $N$  data:

$$\sum_{i=0}^N \min_{\mu_k \in C_K} (\|x_i - \mu_k\|^2)$$

(2.17)

It is standard practice to choose the initial centres uniformly at random from  $X$ . However, David Arthur and Sergei Vassilvitskii in 2007 [36] enhanced the k-means algorithm with a simple randomized seeding technique for initial centres selection that improves both the speed and the accuracy of k-means. More details on the enhanced k-means (k-means++) algorithm can be found in [30]. This k-mean++ is supported by the Scikit-learn library (free software machine learning library for the Python programming language [37], [38]). However, the nature of k-means clustering suffers from various drawbacks [36], [37].

## 2.4 Supervised Machine Learning Methods

### 2.4.1 Supervised Machine Learning Evaluation

As it was mentioned before that supervised learning generally tends to find the relationship between set of features and response or target variables. Before presenting how different supervised machine learning algorithm are working, I will discuss how supervised machine learning models can be evaluated. The purpose of model evaluation is to find the best model that represents the seen (current or training) data and future (unseen) input data well. The model evaluation helps comparing different models, and guides the selected model to carry out parameter tuning, that will result in accurate future predictions. The first concept to understand evaluation is evaluation metrics. The idea behind a metric is a measure to determine how good the model predictions actually match the observed data [39]. The focus will be on regression metrics since only regression was used as supervised learning method in Chapter 5. In the regression setting, one of the most used measure (metrics) is the mean squared error (MSE), given by the following equation[31], [39], [40]:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 \tag{2.18}$$

Where  $f(x_i)$  is the prediction from the machine learning model  $f$  for the  $i$ -th observation of given input  $x_i$ , and  $y_i$  is the actual  $i$ -th observation of the output variable. The smaller the MSE value, the better the model performance and the closer the prediction values to the real observed values.

Another commonly used regression metrics is the coefficient of determination ( $R^2$ ). The coefficient of determination summarizes the explanatory power of the regression model and is computed from the sums-of-squares terms as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - f(x_i))^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (2.19)$$

Where  $\bar{y}_i = \frac{1}{n} \sum_{i=1}^n y_i$

Nevertheless, evaluating the model using the training data (i.e., data used to construct/ build the model) is not practical. In fact, it is a methodological mistake to use the same data set to learn the parameters of a prediction function (model) and test/or evaluate the model performance [38]. Instead, the emphasis is to examine the prediction model when the model is applied to previously unseen test data. A model is called good when it can be generalized beyond the given data and have the ability to generate a useful predicting/classifying for future/unseen data. Therefore, it is common practice when performing a (supervised) machine learning experiment, to divide the labelled data (i.e., features that has a corresponding outputs/targets) data set into two subsets: a training data set and a testing data set. The training data set will be used to construct the model. Whereas the testing data set will be used to test (i.e., evaluate, validate) the model.

Overfitting occurs when the prediction model is not able to be generalized. It happens when the prediction model performs well (high prediction accuracy/ low error) on the training set, while its performance on the unseen test data set is poor. On the other hand, underfitting occurs when the model is not able to capture the pattern of the training data set. Specifically, when, the model is not able to achieve a sufficiently low error or high prediction accuracy value on the training set.

Different methods that are well established, can be applied to validate a machine learning model, overcome the overfitting issue and achieve good model generalization ability as follows [30], [31], [39], [40]:

#### **2.4.1.1 The Validation Set Approach**

In this method the available data set is randomly divided into, a training set and testing set. The testing set is also called the validation set and the hold-out set. The model is constructed based on the training set, and the constructed (fitted) model is utilized to estimate the response for the observations (output of the observation or target) in the validation set. The resulting validation metrics (e.g., prediction accuracy or MSE in case of regression) provides an estimate of the test

error/accuracy rate. However, it is not clear how to divide the labelled dataset, which may lead to statistical uncertainty associated with the estimated average test error of the validation set approach. As the validation estimate of the test error rate can be highly variable, depending on how the data are divided into training and testing. [39], [40].

#### **2.4.1.2 K-fold Cross-Validation**

A K-fold cross-validation provide a solution to the above dilemma. It randomly divides the labelled dataset into  $k$  subsets, called folds that have the same size [39]. Then, it works as follows:

for  $i = 1$  to  $K$ :

1. Build a model using all data subset but the  $k$ -th fold
2. Test the model on the  $k$ -th fold and record the error rate or prediction accuracy
3. If  $i \neq k$ , repeat with next  $k$  (go to step1), else continue to 4.
4. End for and return the average of the error rates or accuracy of for all  $k$ -folds obtained in line 2.

Leave-one-out-cross-validation is a special case of cross-validation, when the number of folds is equal to the data set size. At every iteration of the cross-validation only one data point is left out the model construction (training) for testing for testing. As we will see in the following sections that cross-validation is very useful to find optimal model tuning parameters (hyperparameters) that result in low bias and variance error.

The main objective of any supervised machine learning algorithm is to best estimate the mapping function ( $f$ ) for the output variable ( $y$ ) given the input data ( $x$ ). Supervised machine learning algorithm prediction error can be classified into three types, namely: bias error, variance error, irreducible error, The irreducible error cannot be reduced regardless how well the model ( $f$ ) is estimated. It is the error that may be associated with selecting the framing of the problem, or not including variables that influence the mapping function ( $f$ ) [40]. Variance can be described as the amount by which mapping function will change if different training data set is used to estimate “construct” it. It is necessary for a model not to vary too much between training sets (low variance). However, if a method has high variance, then, small changes in the training data can result in large changes in the mapping function (model). On the other hand, bias can be defined as the error that is raised when oversimplifying the model problem. For instance, approximating an extremely complex phenomena using much simpler model [39]. A mapping function that exhibits low variance but high bias will underfit the target (response variable), while a model with high variance

and low bias will overfit the target. Therefore, a model can be called good when it has the right good balance of low bias and variance (without overfitting or underfitting). Therefore, it is important to trade-off between bias and variance (trade-off in model complexity). It is easy to construct very complex model with extremely low bias but high variance or a very simple model with very low variance but high bias. However, the challenge is to generate a model which is low in both variance and bias.

The following are the description of supervised machine learning methods used in this study.

## 2.4.2 Ordinary Least Square Linear Regression

Linear regression is one of the simplest forms of supervised machine learning methods. It named linear because the target (output) value is expected to be a linear combination of the features (input variables). Mathematically the predicted value of an output can be written as follows:

$$\tilde{y}_i = \omega_1 x_{1,i} + \omega_2 x_{2,i} + \dots \dots \dots \omega_m x_{m,i} + \omega_0 = \sum_{j=1}^m \omega_j x_{j,i} + \omega_0 \quad (2.20)$$

Where  $\omega$  is the coefficient of each feature,  $\omega_0$  is the intercept term and  $m$  is the total number of features,  $x_{j,i}$  is the  $i$ -th observation of  $j$ -th feature and  $\tilde{y}_i$  is the predicted output (response). The goal of the algorithm is to find a best fit of the coefficients ( $\omega$ ) by minimizing the residual sum of squares between the observed targets (actual output) in the dataset, and the targets predicted by the linear approximation  $\tilde{y}_i$ . The following mathematical problem should be solved to find the vector  $\omega$  (coefficients)

$$\min_{\omega} \sum_{i=1}^n (\tilde{y}_i - y_i)^2$$

$$\min_{\omega} \sum_{i=1}^n \left( \sum_{j=1}^m \omega_j x_{j,i} + \omega_0 - y_i \right)^2 \quad (2.21)$$

Where  $\omega_j$  is the coefficient of feature  $j$ ,  $\omega_0$  is the intercept,  $n$  is total number of training data points. and  $y_i$  is the actual output variable of  $i$ -th observation. This cost function called the residual sum squares (RSS). There are usually two ways to solve this mathematical problem and finds  $\omega$ . One is to use the singular value decomposition of  $X$  [41]. The singular value decomposition works very well for of several number of features ( $m$ ). Whereas for large number of features ( $m$ ) the a gradient



descent approach scales very well [39]. However, this simple form of linear regression algorithm is missing a regularization concept that can prevent the training from overfitting. Following are linear regression methods that include two types of regularization penalties.

### 2.4.3 Ridge Linear Regression

Ridge regression is very similar to ordinary least squares linear regression, except that the coefficients are estimated by minimizing a slightly different cost function. In particular, the ridge regression cost function can be written as follows [42]:

$$\min_{\omega} \sum_{i=1}^n \left( \sum_{j=1}^m \omega_j x_{j,i} + \omega_0 - y_i \right)^2 + \lambda \sum_{j=1}^m \omega_j^2 \quad (2.22)$$

where  $\lambda \geq 0$  is a tuning parameter and it is called the complexity parameter where it controls the amount of shrinkage. As it can be seen in the equation (2.22), Ridge trades off two criteria. The first term of the equation is the RSS, where, ridge regression search for coefficient estimates that fit the data well, by making the RSS small. While the second term  $\lambda \sum_{j=1}^m \omega_j^2$ , called a shrinkage penalty, is small when coefficients ( $\omega_j$ ) are close to zero. The tuning parameter  $\lambda$  controls the relative impact of these two terms on the regression coefficient estimates. On other words, the tuning parameter (hyperparameter)  $\lambda$  determines how severe the penalty is imposed. For example, when  $\lambda = 0$  ridge regression will solve an ordinary least square linear regression since the penalty term has no effect. However, when the value of  $\lambda$  increases, the impact of the shrinkage penalty become greater, and the regression coefficient estimates will be close to zero. Ridge regression will generate a different set of coefficient estimates for each value of  $\lambda$ , unlike least square where one set of coefficient estimates is produced. Therefore, selecting a good value for  $\lambda$  is critical. As it is worth mentioning that ridge regression uses the  $L_2$ -norm penalty. This penalty form makes ridge suffer from one obvious disadvantage. The ridge penalty term will shrink all the coefficients towards zero, but it will not set any of them exactly to zero (unless  $\lambda = \infty$ ).

### 2.4.4 Lasso Linear Regression

Lasso is very similar to the ridge regression, however, instead of using the  $L_2$ -norm penalty it uses the  $L_1$ -norm penalty in the cost function as follows:

$$\min_{\omega} \sum_{i=1}^n \left( \sum_{j=1}^m \omega_j x_{j,i} + \omega_0 - y_i \right)^2 + \lambda \sum_{j=1}^m |\omega_j| \quad (2.23)$$

Comparing equation (2.22) to equation (2.23), we see that the lasso and ridge regression are similar, where the only difference is that  $\omega_j^2$  in ridge penalty is replaced by  $|\omega_j|$  in lasso penalty.  $L_1$ -norm and  $L_2$ - norm of vector  $\omega$  are given by the following formula:

$$\|\omega\|_1 = \sum_{j=1}^m |\omega_j| \quad (2.24)$$

$$\|\omega\|_2 = \sum_{j=1}^m \omega_j^2 \quad (2.25)$$

Similar to ridge regression, the lasso shrinks the coefficient estimates towards zero. However, in the case of the lasso, the  $L_1$ - penalty has the ability to force some of the coefficient estimates to be exactly equal to zero when the hyperparameter  $\lambda$  is sufficiently large. Therefore, Lasso regression can also perform as variable selection (feature selection), as it eliminates the irrelevant variables by setting their coefficients to be zero ( $\omega_j = 0$ ). It can be said that the lasso produces sparse models that involve only a subset of the variables.

As we have seen in both ridge lasso, finding the tuning parameter  $\lambda$  (in both case it is the penalty parameter) is crucial. Choosing an optimum tuning parameter  $\lambda$  can be done using cross-validation. Cross-validation provides a simple framework to deal with this problem. A grid of  $\lambda$  values are generated, and the cross-validation error is computed for each value of  $\lambda$ . After that, the tuning parameter value that gives the smallest cross-validation error is selected. Finally, the model is re-fit using all the available observations and the selected value of the tuning parameter. This is called the cross-validation grid search. If more than one hyperparameters are needed to be tuned, cross-validation can still works, however the process might be a time consuming. Therefore, in this study for tuning Support vector machine (SVM) regression parameters, a Genetic Algorithm was used to find the model tuning parameters (see the following section).

#### 2.4.5 Support Vector Regression

Support vector machine (SVM) is a type of machine learning technique that is used in classification, regression and probability density function estimation [43]. In this study, the focus

is on the use of SVM in regression, since the objective involves the prediction of numerical values. SVM is based on the structural risk minimization principle from computational learning theory [44]. The core of an SVM is a quadratic programming problem, separating support vectors from the rest of the training data. The support vector regression structure can be illustrated as a series of given training data  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  where  $x \in \mathbb{R}_d$  represent the  $d$ -dimensional input samples and  $y \in \mathbb{R}$  denote output observations, the linear case regression problem can be written as follows:

$$f(x) = \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n + b = \langle w, x \rangle + b \quad (2.26)$$

Where  $\{\omega_1, \omega_2 \dots \omega_n\}^T$  denote the regression coefficients and  $b$  is the bias term. The regression goal is to find these unknown through the support vector regression optimization as follows [45]:

$$\min_{\omega, b, \xi^+, \xi^-} \left[ \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n \xi_i^+ + \xi_i^- \right] \quad (2.27)$$

Subjected to the constraints:

$$y_i - \langle \omega, x_i \rangle - b \leq \epsilon + \xi_i^- \quad , \quad \forall i \quad (2.28)$$

$$-y_i + \langle \omega, x_i \rangle + b \leq \epsilon + \xi_i^+ \quad , \quad \forall i \quad (2.29)$$

where  $\epsilon$  denotes the precision threshold,  $C$  denotes the regularization parameter (hyperparameter) and  $\xi_i^+$  and  $\xi_i^-$  denote the slack variables with nonnegative values to ensure feasible constraints. The first term in equation (2.27) represents model complexity while the second term represents the model accuracy or error tolerance. The linear  $\epsilon$ -insensitive loss function ignores errors that are within  $\epsilon$  distance of the observed value by treating them as equal to zero. The loss is measured based on the distance between observed value  $y$  and the  $\epsilon$  boundary. In other words, samples whose predictions is at least  $\epsilon$  from the true target are penalized.  $\xi_i^+$  and  $\xi_i^-$  represent the value of penalization that is added to objective depending on whether sample predictions lie above or below the  $\epsilon$  tube. The penalties imposed on observations that lies outside  $\epsilon$  tube are controlled by the positive constant  $C$ . The  $C$  value determine the trade-off between model smoothness (flatness) and minimization of prediction error, therefore helps avoiding overfitting. The constant  $C$  has an opposite effect to the regularization penalty  $\lambda$  introduced Ridge and Lasso regression as it controls the strength of error.

The former optimization problem can be solved easier in its Lagrangian dual formulation. Solving the dual problem provides a lower bound to the solution of the primal (minimization) problem. The difference between the solution of primal problem and dual problem is called the duality gap. However, when the problem is convex, and a constraint qualification condition is satisfied, the optimal solution of the primal and dual problem is the same. In order obtain the dual formulation of primal function, the nonnegative Lagrange multipliers  $\alpha_n$  and  $\alpha_n^*$  for each observation  $x_i$  are introduced. Accordingly, the dual optimization problem can be written as follows:

$$\max_{\alpha, \alpha^*} \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n [(\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle] + \sum_{i=1}^n [\alpha_i (y_i - \epsilon) - \alpha_i^* (y_i + \epsilon)] \quad (2.30)$$

subjected to the constraints

$$\sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \quad (2.31)$$

$$0 \leq \alpha_i, \alpha_i^* \leq C, \quad \forall i \quad (2.32)$$

In nonlinear case, the above objective function (equation (2.30)) can be modified by substituting the dot product  $\langle x_i, x_j \rangle$  with a kernel function  $K(x_i, x_j)$  :

$$\max_{\alpha, \alpha^*} \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n [(\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(x_i, x_j)] + \sum_{i=1}^n [\alpha_i (y_i - \epsilon) - \alpha_i^* (y_i + \epsilon)] \quad (2.33)$$

The theory of Kernel function was developed [46] which is basically used to transform the input feature vector to a higher dimensional space and can be expressed as follows:

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j) \quad (2.34)$$

Where  $\varphi$  is the feature mapping. Kernel is a function that takes input vectors in original space and returns the dot product of the vectors in the enlarged feature space. A kernel is a function that quantifies the similarity of two observations. The advantages of kernel instead of explicitly applying the transformations on the enlarged feature space, the dot product calculation take place at the original input space (i.e., one need only compute  $K(x_i, x_j)$  for all  $\binom{n}{2}$  distinct pairs  $(i, j)$  [39].

The regression function that used to perform new predictions can be written as follows:

$$f(x) = \sum_{i=1}^{sv} (\alpha_i - \alpha_i^*)K(x, x_i) + b \quad (2.35)$$

This function does not depend on the whole training data set  $n$ , it depends only on the number of support vectors  $sv$ . Then, the regression coefficient  $\omega$  parameter can be completely described as a linear combination of the training observations using the following equation:

$$\langle \omega, x \rangle = \sum_{i=1}^n (\alpha_i - \alpha_i^*)K(x, x_i) \quad (2.36)$$

The bias can be computed as follows:

$$b = f(x) = \begin{cases} y_i - \sum_{i=1}^n (\alpha_i - \alpha_i^*)K(x, x_i) + \epsilon, & \text{if } 0 \leq \alpha_i \leq C \\ y_i - \sum_{i=1}^n (\alpha_i - \alpha_i^*)K(x, x_i) - \epsilon, & \text{if } 0 \leq \alpha_i^* \leq C \end{cases} \quad (2.37)$$

Generally, there are several kernels that are used in SVM such as linear, polynomial, Radial Basis Function (RBF). RBF is one of the most popular Kernel that has been applied extensively [47]. In this study, RBF is used as the kernel for SVM because it is practical and relatively easy to tune. The RBF kernel function for two points  $x_i$  and  $x_j$  measures the similarity of these points to each other, RBF kernel can be mathematically represented as follows:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \quad (2.38)$$

Where  $\gamma$  is a kernel hyperparameter and it is the inverse of the standard deviation of the RBF kernel (Gaussian function). From the former discussion, the understudy SVM regression model has two hyperparameters needed to be tuned  $C$  and  $\gamma$  (gamma).

#### 2.4.5.1 Genetic Algorithm for Hyperparameters Tuning

The optimum numerical values of these two parameters are calculated using Genetics Algorithm (GA) optimization. GA from its metaheuristic optimization approach inspired by genetics and the process of natural selection introduced by J. Holland in the 1960s and 1970s [48]. It is widely used to find high quality solutions for optimization problem.

The general idea is to search optimal solution over a population through transforming population (set) of individual objects, each with an associated fitness value, into a new generation of the population using the Darwinian principle of reproduction and survival of the fittest. During the process of GA several similar operations to the natural operations such as crossover (sexual recombination) and mutation is occurring [49].

The algorithm starts by generating an initial population, where population consists of *individuals (chromosomes)* in the population. Each individual represents a candidate solution to a given problem with a unique set of genes. These individuals in the case of machine learning can be set as the tuning hypermeters (potential optimal solution), and a single gene can be represented by single hyperparameter. Then, these individuals are evaluated using a fitness function. In case of finding the optimal hyperparameter of machine learning algorithm using GA, the fitness can be any performance metric, such as MSE or coefficient of determination. In this case study the cross-validation MSE score over 5-folds is used as the fitness. After that, based on the fitness value, the top performing individuals of the population are selected (“survival of the fittest”), as the survived population (this process called selection). In this study 50% of the top performing individual are selected. These survived individuals of the population are called parents. Next, mating between parents in the survived population will take a place to produce offspring through undergoing crossover/recombination and mutation operation. In crossover, the genes (parameters) from the mating parents are randomly recombined, to produce offspring. Crossover produces new individuals (offspring) that inherited some genes of both parents’. While, in the mutation operation, some genes of the offspring are randomly altered. Mutation is necessary to maintain a genetic diversity (solution diversity) which will help protect the loss of some of the good solution and avoid the sub-optimal solutions. By this a new generation of population is formed, which contains both survived parents as well as offspring. Keeping survived parents in the new generation will help retain best fitness individuals (parameters) in the case of children’s fitness value turns out to be worse than the parents [50].

This process of keeping survived parents in the new generation is called partial replacement. Finally, the new generation population will replace the old one. This process is repeated until the stopping criterion is met. In this study the stopping criterion is the predetermined number of generations. Figure 2.1 shows a schematic representation of GA steps.

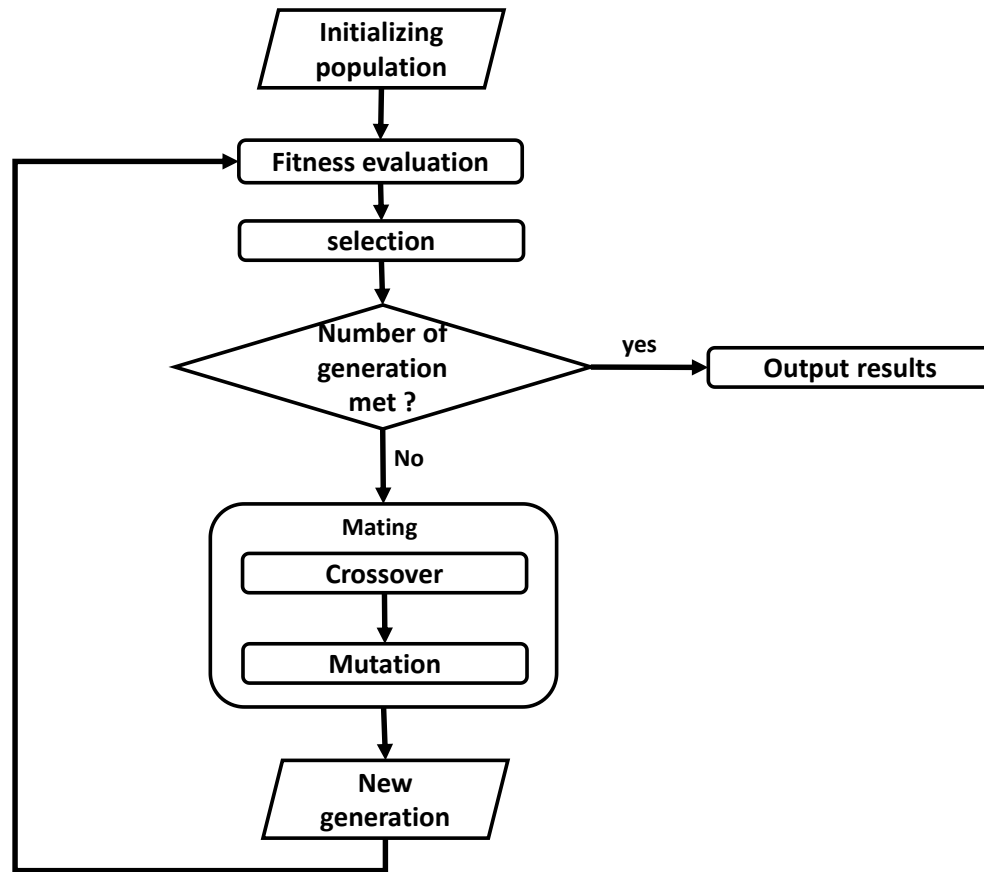


Figure 2.1. Schematic representation of GA process

#### 2.4.6 Artificial Neural Network

Artificial Neural Networks (ANN) were developed as an information processing system that is inspired by the biological nervous system, where they are capable of learning and processing complex information [51]. ANN is used as non-linear data modelling that can learn patterns in the data, and estimate functions that define the relationship between inputs and outputs [11]. There appears to be little agreement on an all-inclusive definition of an ANN [52]. The ANN is a system that consist of many simple processing elements operating in parallel. The network structure, connection strengths, and the processing performed at computing elements define the network function. These interconnected processing elements called nodes or neuron. A node in ANN computer model resembles the main functional parts of a single neuron in the nervous system. A node has multiple inputs and one output. This output may be sent to other multiple nodes or could be considered as network output (if the neuron is located at the output layer of the network), but the same signal is passed to each. Several processing steps are performed within a node. The node

has a number of  $n$  inputs  $x_1, x_2, \dots, x_n$  represented as vector  $\mathbf{x} \in \mathbb{R}^n$  and each input have an assigned weight  $w_1, w_2, \dots, w_n$ . Next, all inputs are summed with respect to the assigned weights  $\sum x_i w_i$  and the bias term ( $b$ ) is added. After that, the summed input is passed through an activation function  $f$  (transfer function) associated with the node which generates the value of an output signal ( $y$ ). So the output of a neuron can be defined as follows:

$$y = f\left(\sum x_i w_i + b\right) \quad (2.39)$$

Variety of activation functions are available and have been widely used in numerous ANN applications in different field (see Table 2.3). The output of the activation function may pass to one or more neurons. Figure 2.2 shows a schematic diagram of a single node.

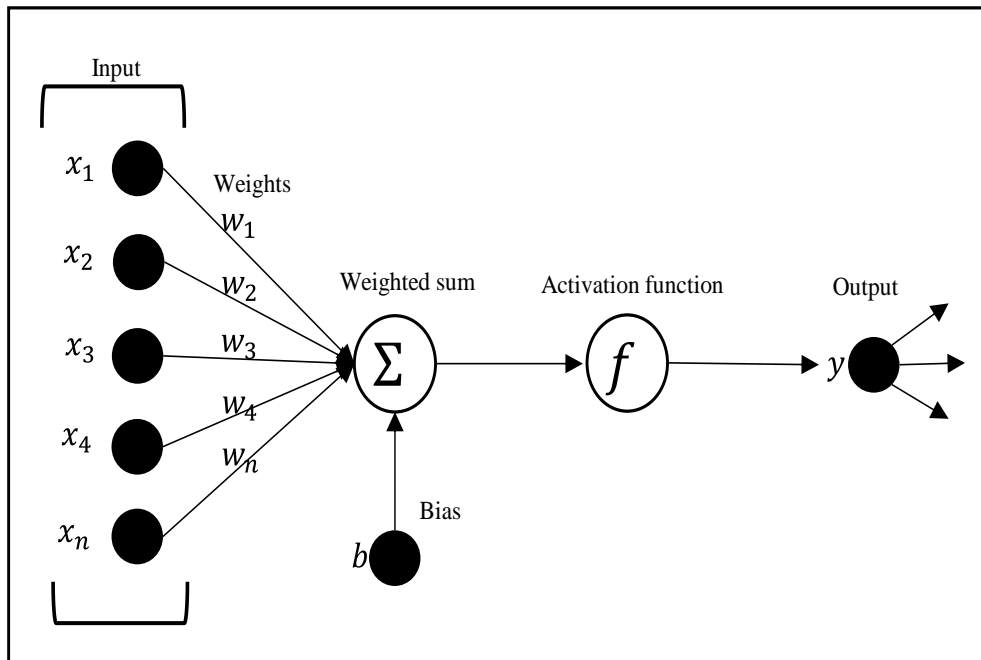


Figure 2.2. Schematic representation of a single neuron

#### 2.4.6.1 The Feed-Forward Neural Networks

A single neuron also called a Perceptron, which is basic unit of the neural neuron. Single perceptron is limited only to model simple task (linear models) whereas more complex phenomena such as nonlinear systems are difficult to explain using single Perceptron. Therefore, several neurons and layers are interconnected to form a network to solve more complex tasks, motivated

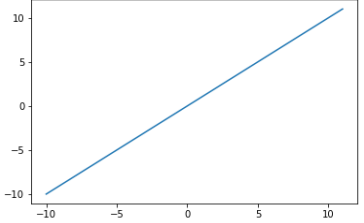
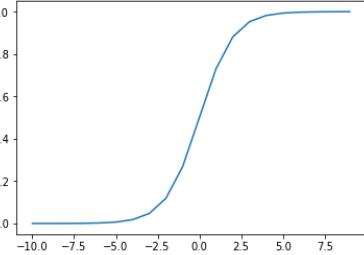
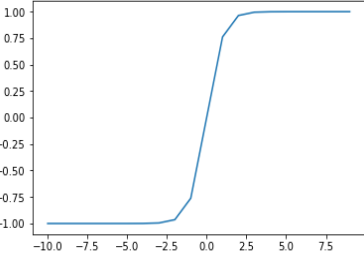
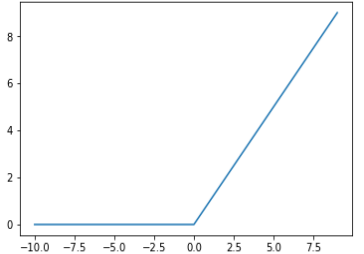


by the nervous system architecture models. The interconnected Perceptron results in fully connected feed-forward Networks, also called Multi-Layer Perceptron (MLP's), MLP's considered to be the prototypical models for ANN and deep learnings. and consists of networks of multiple interconnected layers of neurons [40]. To illustrate a fully connected ANN structure, an example of Three-Layer Perceptron is shown in Figure 2.3. Feed forward fully connected networks are a common structure where each layer is stacked one after another and each neuron is connected to every neuron in their previous layer [53]. The network involves three groups of layers: input layer, one or more intermediate layers (hidden layers) and an output layer. The input layer of the network is fed with input data presented in the problem (features), denoted here as  $\mathbf{x}_m = \{x_1, x_2, \dots, x_m\}$ . After that, this data is passed to the neurons in the following layer. These intermediate layers, which are known beforehand, are often referred as hidden layers, because neither the inputs nor the outputs of these layers are known. The function of these intermediate layer is to learn the beneficial features contained in the data needed to address a particular problem. Each neuron then performs a transformation on its inputs using the node assigned activation function, and the output of this neuron can be computed as follows:

$$x_k^l = f \left( \sum_{i=0}^{K^{l-1}} x_i^{l-1} w_{i,k}^{l-1} + b_k^l \right) \quad (2.40)$$

Where  $f$  is the activation function,  $x_k^l$  is the state of the  $k$ -th neuron in layer  $l$ ,  $w_{i,k}^{l-1}$  is the weight vector linking the neuron  $i$  of previous layer with the states of the previous layer  $x^{l-1}$ ,  $b_i^k$ , is called the bias of the neuron  $k$  at layer  $l$  and  $K^{l-1}$  is the total number of neurons in the previous layer. This type of neural network is called feedforward since computing the states of each layer requires knowledge of the states of the previous one. In the last layer (i.e., output layer), the final answer to the network problem (in which the network was designed for), is provided which may be a class label in classification problems or continuous values in general prediction (regression problem). It is worth mentioning that there are no limits on the number of layers nor number of neurons that a network can be built of. However, in feed-forward networks, there are no interconnections between neurons at the same layer and thus the data is transmitted only in one forward direction.

Table 2.3. Selected types of commonly used neural networks activation function

Activation function name	Mathematical formula	Plot
Linear	$f(x) = mx + c$	
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$	
Tanh	$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	
Rectified Linear Unit (ReLU)	$f(x) = \max(0, x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	

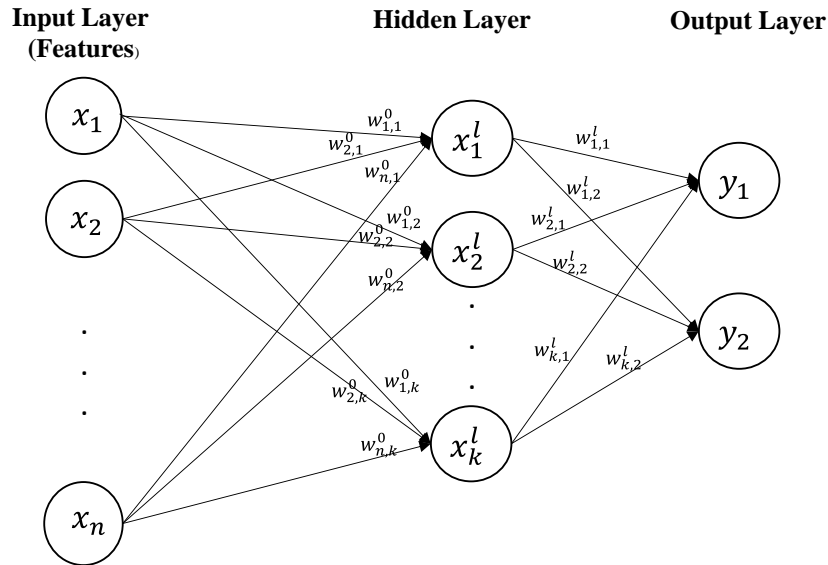


Figure 2.3. Graphical representation of example of feed-forward fully connected three-layer perceptron

The goal of training an artificial neural network is to find a combination of weights matrices and bias vectors for the whole network that minimizes the error between the states of its output layer  $y_n = \{y_1, y_2, \dots, y_n\}$  (predicted output values) and their targets (actual values). This error is referred as the loss function or the cost function. Therefore, training data must be arranged in such a way that every combination of inputs  $\mathbf{x}$  has one or several outputs (labels)  $\mathbf{y}$  that correspond to them.

ANNs have the ability to simulate complex problems (e.g., nonlinear problems) by employing a different number of nonlinear activation functions. However, introducing these nonlinear activation functions often leads to a non-convex optimization problem for the training process. Therefore, the solution of the network optimization problem cannot be solved explicitly. Hence, numerical optimization approaches (i.e., commonly gradient-based) are used to train neural network and the solutions of those approaches (values of weights and biases that best fit a given task) are not guaranteed to be the global optimal [40].

### 2.4.6.2 Activation Functions

Traditionally, the most commonly used activation functions for neurons are linear, standard logistic sigmoid and hyperbolic tangent. Table 2.3 shows the corresponding equation and graphical representation for the most popular activation functions. The first one is the linear activation

functions are easy to compute and their training process is simple, however, they are unable to learn complex nonlinear patterns. The remaining functions are the most popular nonlinear activation functions. logistic curve or sigmoid and hyperbolic tangent function are commonly used and they are similar. The sigmoidal function generates an S-shaped output between 0 and 1 while the tanh output also forms an S-shaped ranges from -1 to 1 The tanh activation function is more desirable than the sigmoidal due to the fact that it is zero-centered (near zeros it resembles the identity function ( $f(x) = x$ ), which can lead to improvements during training). However, both functions suffer from the problem of vanishing gradient. The problem occurs when the value of  $x$  (input to the activation function) falls away from the zero point, where the functions curve becomes increasingly flat (reaching saturation and the slope is almost zero) Therefore, the gradients will become vanishingly small resulting in inhibiting the weight from updating its value

Lately, recent deep learning works have adopted the rectifier linear activation  $relu(x) = \max(0, x)$  Inspired by biological studies [54]. Recently ReLU are considered to be the most used activation function in current neural network applications. ReLUs has several advantages such as allowing a network to have sparse connections (transferring only important information through the network) and avoiding the vanishing gradient problem because of its linear nature for positive  $x$  values (gradient can equal to 1 for all positive values of  $x$  and 0 otherwise)[55]. Although the ReLU function is not strictly differentiable at  $x = 0$  due to the discontinuity, this can be overcome by arbitrarily setting the gradient at this point as either 1, 0, or 0.5.

It is important to mention that any of the activation can be used in any of the layers of the network. Different activation functions at the output layer implies different task. linear activation function at the output layer should be used.

Many other activation functions have been suggested in literature and successfully and carried out in different application, Basically, many of these functions are derived from sigmoid, hyperbolic tangent, and ReLU functions.

### **2.4.6.3 ANN Training Process**

The process of selecting the values (e.g., weights and bias) of a network is referred to as training process. The training process is based on an iterative adjustment of the network parameters values such as to minimize a cost/loss function. ANN was resurrected in 1986 by the invention of backpropagation [56], which is a computational algorithm that can help solve the ANN training

problem. In feedforward neural network, the input  $x$  is passed through the network and undergo different manipulation to produce the output  $\tilde{y}$ . This process called forward propagation. On the other hand, Back-propagation [56] is an algorithm that allows the information from the cost function to flow backwards from the network's output layer towards its input, in order to compute gradients. In detail, the gradient of a cost function (loss function)  $E$  with respect to the network's internal parameters (the weight matrices  $w$  and bias vectors  $b$ ) using back-propagation. The goal of back-propagation is to compute the gradients and express the gradient in terms of network parameter as follows:

$$\nabla_{w,b}E(w, b) = \left( \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_n}, \frac{\partial E}{\partial b_1}, \frac{\partial E}{\partial b_2}, \dots, \frac{\partial E}{\partial b_m} \right)$$

Where  $E$  is the loss/cost function,  $n$  and  $m$  are total numbers of weights and bias respectively. Once these gradients are determined, a numerical optimization algorithm is used to update the network parameters, where the objective function (cost function) can be defined as the deviation between predictions  $\tilde{y}$  and desired outputs (actual output or labels  $y$ ). The selection of these cost function is largely relying on the particular task been performed by the neural network. An example of widely used cost function in regression tasks is the MSE (see equation (2.18)). In general, the gradient of the cost function with respect to such a weight for any type of activation function and at any node can be expressed using the chain rule of calculus as follows:

$$\frac{\partial E}{\partial w_{i,k}} = \frac{\partial x_k}{\partial w_{i,k}} \frac{\partial f}{\partial x_k} \frac{\partial E}{\partial f} \quad (2.41)$$

$E$  denote the cost function (e.g., mean square error), and  $w_{i,k}$  denote the weight connecting the  $i$ -th neuron in layer  $l - 1$  and the  $k$ -th neuron in layer  $l$ .

Similarly, the gradient with respect to a bias at a certain neuron  $k$  can be computed as follows:

$$\frac{\partial E}{\partial b_k} = \frac{\partial x_k}{\partial b_k} \frac{\partial f}{\partial x_k} \frac{\partial E}{\partial f} \quad (2.42)$$

After that, once every gradient is known for a given training iteration, an optimization algorithm must be applied to update the parameters of the neural network (weight matrices and bias vectors). One of the most popular optimization algorithms to train ANN is the steepest descent algorithm. In simple steepest descent algorithm, the parameter update step would be written as follows [40]:

$$\theta_{i+1} = \theta_i - \eta \nabla_i E(\theta) \quad (2.43)$$

Where  $\eta$  is the learning rate,  $\theta_i$  is the internal network parameters which includes the weight matrices and bias vectors at the  $i$ -th iteration,  $\nabla_i E(\theta)$  is the gradient of the cost function (loss function)  $E$  with respect the network internal parameter  $\theta$  at  $i$ -th iteration. In every step of iteration, not all the available training data will be used to update the network internal parameters, to avoid the optimization process from stopping at globally high local minima. Therefore, it is preferable to use smaller batches of data to prevent the training from reaching local minima [57]. The training sample used in each batch should be selected randomly, to keep the gradient estimation unbiased. In this context the optimization process become stochastic. Therefore, steepest descent becomes mini-batch stochastic gradient descent, or Stochastic Gradient Descent (SGD) for short [58].

Generally, the larger the batch size the faster the training process, whereas the smaller the batch size the slower the training process. Nevertheless, better cost function minima can be achieved by using smaller batch sizes as it introduces more stochasticity to the training process.

In ANN training, the stopping criteria are usually determined in terms of the number of iterations (epochs to be performed). The number of epochs can be defined as number times (iterations) that the learning algorithm will work through the entire available training dataset. While batch size is number of training samples used to train the network before updating the internal model parameters.

One of the main challenges in using stochastic gradient descent is tuning the learning rate parameter  $\eta$ . Choosing a too large learning factor, results in larger step parameter corrections along the error space function that may lead to skip over the optimum (reach sub-optimal set of weights and the training process is too fast or unstable), while, selecting a small learning factor will result in very slow convergence speed. SDG has a constant learning rate by default, which might be too simplistic for many situations, since the cost function is often highly sensitive to changes in some directions of the parameter space, and highly insensitive to others.

Recently, a wide range of adaptive learning rates algorithm have been developed and implemented to adjust the learning rate during training process. Some of the most popular optimization methods that have been developed to effectively adjust the learning rate are SGD with and without momentum[59] AdaGrad [60], RMSProp [61] with and without momentum, and Adam[62].

Adam named after the term “Adaptive moments”, which is one of the fastest ANN optimization algorithm that can achieve good performance results, while does not require much hyperparameter tuning [62]. Adam is one of the best and most robust optimization algorithms for deep learning and

its popularity is growing very fast. Therefore, this study adopted Adam method to train the developed ANN in Chapter 5. Adam employs exponentially moving averages, calculated on the gradient estimated on a current mini-batch.

The mathematical representation of the first moment at any  $i$ -th mini-batch iteration can be shown as follows:

$$s_i = \rho_1 s_{i-1} + (1 - \rho_1) \frac{\partial E}{\partial \theta_i} \quad (2.44)$$

while the second moment can be computed using the following equation:

$$r_i = \rho_2 r_{i-1} + (1 - \rho_2) \frac{\partial E}{\partial \theta_i} \frac{\partial E}{\partial \theta_i} \quad (2.45)$$

It is suggested that the default parameters for  $\rho_1$  and  $\rho_2$  are 0.9 and 0.999 respectively, and the default value for the learning rate ( $\eta$ ) is 0.001. Then, to account for both moments initialization at zeros, a correction- bias step is made as follows:

$$\tilde{s}_i = \frac{s_i}{1 - (\rho_1)^i} \quad (2.46)$$

$$\tilde{r}_i = \frac{r_i}{1 - (\rho_2)^i} \quad (2.47)$$

After computing the moment corrections, the parameter (weights and bias) update is calculated as follows:

$$\theta_{i+1} = \theta_i - \eta \frac{\tilde{s}_i}{\delta + \sqrt{\tilde{r}_i}} \quad (2.48)$$

Where  $\delta$  is small constant that is added to prevent division by zero (stability constant usually it is  $10^{-7}$ ). it has been shown that the Adam method performs better than Adagrad and RMSProp because of its bias-correction step [63].

#### 2.4.7 Example of Applying a Machine Learning Method

This example demonstrates the application of supervised machine learning in regression task. In order to do that, firstly a data set was generated from a polynomial function (see equation (2.49)) After that SVM regression was used to approximate the polynomial function. The following third order function is used to generate the data.

$$y(x) = x^3 - 9x^2 + 15x + 4$$

(2.49)

Where  $x$  is the feature variable (input data/variable), and  $y$  is the response. The goal is to construct a  $f$  mapping function that can approximate this polynomial function. In order to simulate a real case scenario, a random noise is added to the data. Figure 2.4 shows the  $y(x)$  function curve and the generated data with noise.

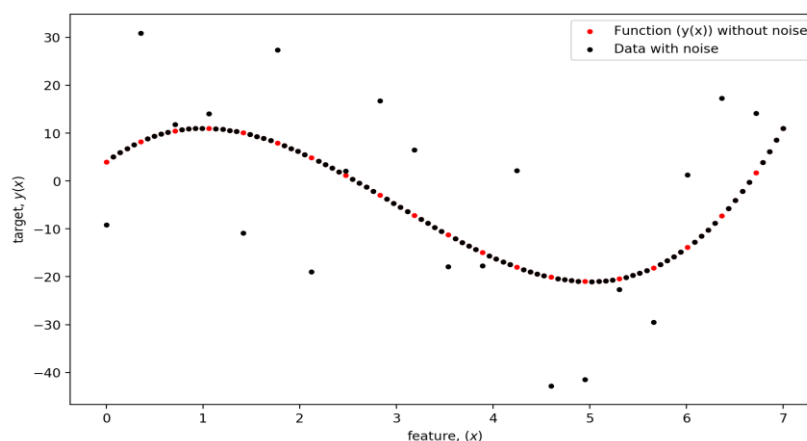


Figure 2.4. Generated data example

Then the cross-validation grid search is used to find the optimal hyperparameters of a model which results in the most accurate prediction. As it was mentioned that SVM regression with radial based function kernel has two parameters to tune. However, in this example, to keep the illustration simple, the gamma value was set to its default value and only  $C$  value is adjusted.  $C$  is a SVM regularization parameter that controls the tradeoff between the achieving a low bias and variance error. A set of values on  $C$  at  $\{0.001, 0.1, 1, 10, 100, 1000, 10000, 100000\}$  was generated and evaluated through cross-validation strategy with 5 folds. Figure 2.5 below shows the variation of cross-validation score (mean of  $R^2$  value for all folds), and cross training score as function of  $C$  value. As it can be depicted in this figure when the value of  $C$  is small both cross-validation score and training scores are low (high bias), which means that the model is so simple that cannot explain the data trend (underfitting). On the other hand, at very high value of  $C$ , the training score is high while the cross-validation score is low, that implies that model is overfitting the target. Moreover, at this high  $C$  value we can say that the bias is low while the variance is high. At midrange of  $C$  value, it is clear that both training and cross-validation scores are high (i.e., low bias and low variance), where a trade-off between bias and variance is achieved (model is not overfitting neither underfitting the predictions).



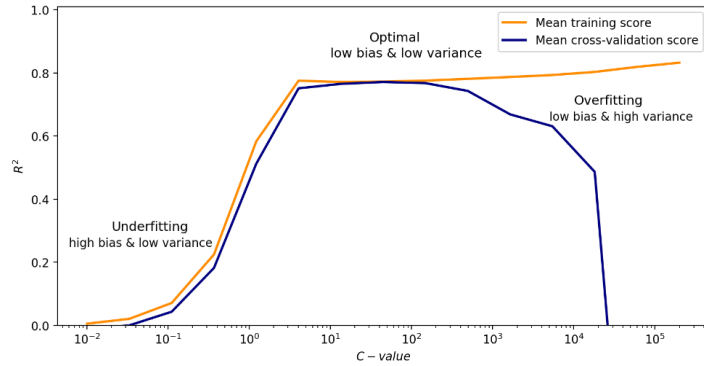


Figure 2.5. The variation of training score and validation score as a function of  $C$

After that, three values of  $C$  are used to train the SVM model using the same set of data to see effect of  $C$  value on the estimated model curve. The three  $C$  values used are 0.01, 44.7 and  $5 \times 10^4$  and the model curves are showing in Figure 2.6. As it can be shown from the figure that when the value of  $C$  is low, the model fails to capture the data trend. On the other hand, when large value of  $C$  was used, the model tries to follow the exhibited noises of the data and therefore, it overfits the target variable. We can see that at optimal value of  $C$ , the model follows the actual trend of the data (the model captures the actual data response variable without the data noises).

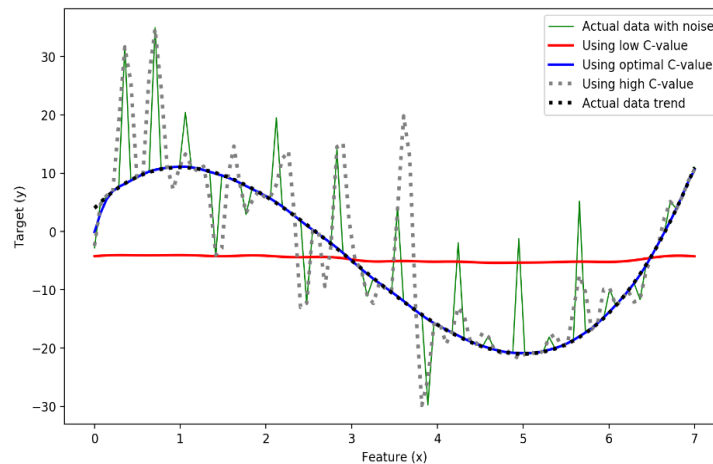


Figure 2.6. Comparison of model curves using different values of  $C$

## 2.5 General Literature Review

This section presents a general review of recent literature on the topic of data-driven optimization in recent years. More specific literature reviews for each chapter will be presented subsequently in this dissertation.

Regarding dealing with uncertainties in optimization problems using the recent advances of big data tools (e.g., machine learning), several studies have been reported in the literature. For example robust optimization using machine learning for uncertainty sets was developed by Tulabandhula *et. al.* [64]. Several studies were conducted by Chao Ning and Fengqi You [65]–[69], to address this issue. However, most of these studies were based on robust optimization approaches and were not applicable for power generation planning under intermittent renewable energy resources. A very recent review paper stated that the literature lacks studies on implementing machine learning algorithms into the planning models [70].

Many research studies have been dedicated to study the optimal planning and scheduling of energy hub systems from different perspectives such as [71]–[76], however these studies did not consider employing effective size reduction tool to reduce the size of the stochastic multiscale energy hub system that satisfies multiple energy carrier demands (e.g., heat and power).

The first reported work using surrogate models in chemical flowsheets in the literature is by Palmer and Realf [77], where polynomial and kriging models were applied as approximation in optimization of flow sheets [77]. Jin *et. al.*, [78] investigated the advantages and disadvantages of Polynomial, Kriging, Multivariate adaptive Regression Splines (MARS) and Radial Basis Functions. In 2002, Hetzel upgraded an established refinery-wide optimization that had been developed by Li [38] by replacing some the detailed unit models equality constraints with surrogate models [79]. In 2011, a superstructure-based strategy framework was proposed, where complex unit models are replaced with surrogate models built from data generated via commercial process simulators, by Henao and Maravelias [5]. Anna *et. al.*, [80] studied the pressure swing adsorption for the separation of  $N_2/CH_4$  in a bed packed with silicalite. using an ANN as a surrogate.

Qin [13], in his review paper revealed the importance of recent developments in machine learning and he highlighted that there is a gap in process system engineering and room for more use of machine learning algorithms. For example, at the higher level of production planning and scheduling, industry had accounted for the typical uncertainty (e.g., product demand and price) by

using simple probability distributions and time-series analysis. However, the application and practice of data analytics in process system engineering can be dramatically improved by relating them to the recent developments in data mining (i.e., machine learning and big data analytics) [13] Accordingly, in this work, we will take advantage of the recent advancements in big data tools by strategically utilizing them to be integrated with process industry and energy infrastructure decision making frameworks. This approach (data-driven optimization) will be useful in reducing computational time, increasing flexibility, and maximizing the utilization of available real data. From the literature review of previous data-driven optimization studies, the following research gaps can be recognized and will be addressed in this project.

- Although some studies have been conducted on data-driven optimization under uncertainty, this problem is still insufficiently explored. There is room to apply machine learning methods to further incorporate uncertain data that will emerge from the development of new energy systems which integrate renewable energy (e.g., wind and solar). These renewable energies are more challenging to be modelled within the power planning optimization framework, due to their high level of intermittency and uncertainty. The current advances of big data tools represent a promising solution to extract useful information from these renewable energies and integrate that into the power planning models.
- A knowledge gap exists in developing a stochastic optimization framework for energy hub systems which comprehensively takes into account the design and operation (multiscale) decisions, energy storage system, uncertainties of distributed energy resources and reduction of large size multiple attributes demand data. Therefore, research presented in Chapter 4 is conducted to address these gaps.
- As it can be concluded from the previous literature on data-driven surrogate models review, most of these studies used commercial software simulation data to train their model or small size of plant data. Additionally, many of these studies did not perform process optimization using the developed surrogate models. Therefore, this study showcases a comprehensive data-driven surrogate-based optimization framework that is based on actual plant data. In this study, surrogate model developments involve data cleaning, machine learning model construction, optimization and validation. Also, it proposed a unique surrogate-based optimization framework that integrate data-driven model (i.e., surrogate) within optimization models at two elements, objective function and constraints.

# **Chapter 3      Machine Learning Framework for the Formulation of Efficient Scenarios for Stochastic Programming: Application to Power Generation Planning Under Demand and Wind Data Uncertainties**

## **3.1 Introduction**

Deterministic process design and operation models can help ensure an optimal solution for certain process parameters (e.g., demand, fuel price) where it satisfies the constraints associated with that parameter (i.e., product should be greater than or equal to demand). As most real-life problems involve some sort of uncertainty which are hard to predict [81], deterministic models are incapable of resolving them. There exist a considerable number of studies from industry and academia on optimization under uncertainty[82]–[84]. However, these approaches do not take advantage of the recent advances in machine learning and big data analytics to leverage uncertainty data for optimization under uncertainty. Traditional models of decision-making under uncertainty assume perfect information, which means either accurate values for the system parameters or specific probability distributions for the random variables. Nevertheless, such exact knowledge is rarely available, prior knowledge on uncertain parameter distribution is unknown and fitting random variables (uncertain parameter) into a popular distribution is complicated and impractical [3]. Furthermore, it is mathematically intractable to deal with erroneous inputs (all sets of uncertain data) and this could lead to infeasible solutions or exhibit poor performance when implemented [3]. Therefore, in this chapter a data-driven stochastic approach for power generation planning and scheduling that can efficiently utilize the available historical demand data and wind speed data through advances of data analytics tools such as k-means clustering (a machine learning tool) is proposed. In other words, goal is to generate reduced size scenarios that are efficient in representing the wide spectrum of most probable uncertain scenarios and lead to an inexpensive computational problem. By doing this, data-driven power planning decisions are made against uncertainty realization. It is worth mentioning that the proposed clustering approach to generate stochastic scenarios is general and can be applicable to different planning models where uncertainties may emerge.

In this study, power planning model is formulated in such a way that its design and operational decisions can be determined. The main objective of this chapter is to formulate and solve the mathematical problem of the design and operation for a power generation plant integrated with wind energy. The model is expected to incorporate design and operational decision based on uncertain electricity load and varying wind speeds. Unsupervised machine learning algorithm (k-means clustering) is employed to generate uncertainty scenarios from the historical weather and demand data. Accordingly, reduced size uncertainty scenarios, that feature underlying patterns from uncertain parameters, are generated. These scenarios are used as inputs to the stochastic power planning model where the proposed model is formulated as a mixed integer linear programming (MILP). The UC problem can be defined as finding the optimal scheduling of electric power generating units over a short-term period, i.e., typically from 24 hours to one week, in order to minimize the operations costs. The unit commitment optimal solution must obey the technical constraints and must satisfy the demand. The design and operation of the power planning model integrated with the wind energy model can be divided into two phases, namely deterministic and stochastic with recourse formulation. In the deterministic model the hourly expected values (i.e., means) over one year for one day of the uncertain parameters (electricity demand and wind speed) are used as inputs. The deterministic model is solved for a one-day time horizon, as wind speed and electricity demand are assumed to be certain. We have only one day profiles for demand and wind speed that represent the entire previous year/ years. On the other hand, in the stochastic approach, the problem is formulated as two stage stochastic programming. The first stage variables are associated with power generation design decision, whereas the second stage variables are associated with unit commitment operation (i.e., scheduling). The uncertain parameters (i.e., electricity demand and wind speed) are processed and recognized using unsupervised machine learning. Different scenarios are generated for this uncertain parameter. Clustering algorithm is used to produce uncertain parameters profiles (i.e., each scenario corresponds to the profile of wind speed or electricity demand for 1 day, in other words each scenario is a vector with 24 dimensions and each cluster is associated with a certain occurrence/probability). Different scenarios are used as inputs to the stochastic model. The time horizon for the stochastic model is also one day, however, there are many scenarios for this day at each hour (e.g., the wind power at hour 2 of the day is different depending on the scenario/profile). Clustering strategy have proven their ability to aggregate cyclic data based on the concept of cyclic scheduling (e.g., electricity

demand follows daily cycle). It is used extensively in process systems [85], [86]. Cyclic scheduling requires certain demands to be processed over certain time periods repeatedly within the time horizon. However, these aggregated cyclic results, to our knowledge have only been incorporated in deterministic modelling which works only under certain information. Clustering technique was not widely used to reduce the size of the uncertain data. In other words, these previous studies tackle another aspect of size reduction problem. They dealt with the problem associated with the integration of multiscale modelling as will be presented in the next chapter. Therefore, applying clustering algorithms to extract patterns from uncertain data and use its output to be fed into a stochastic optimization formulation is an interesting research area and more exploration on this approach can be performed.

This stochastic model has two level decisions (i.e., operational and design) whose objective is to minimize the capital and operating cost. The capital costs correspond to the number of generator units needed to be installed and the number of wind turbines, while the operating costs are associated with the amount of power generated by these units while meeting electricity demands. There are several mathematical models for the unit commitment problem available in the literature [87]. In this study, we adopted [88] the UC model formulation as the basis for our formulations of power planning model. The model is formulated as a mixed integer linear programming (MILP). The following sections present the model formulations and related consideration.

The rest of this chapter is organized as follows: Section 3.2 describes the deterministic formulation of the power planning model. Section 3.3 presents the stochastic data-driven power generation planning model formulation. It also includes the generation of uncertain scenarios from historical data. Results of both formulations are discussed. Section 3.4 shows the application of the proposed approach applied to another type of stochastic power generation planning model. Section 3.5 presents concluding remarks.

## **3.2 Deterministic Design and Operation Formulation of Power Generation Model**

Consider a set of  $i$  thermal and wind turbine units to be scheduled over a time horizon  $T$ . The goal is to minimize the overall cost (i.e., capital and operating). This goal can be achieved by optimally determining the number of power generation units (both thermal and wind) and scheduling the thermal (conventional) generating units. The mathematical programming framework ensures that

these optimal solutions are meeting the electricity demands and operating within the units' capacities (i.e., technical constrains). The problem is solved for 10 thermal units and 24 hours.

The objective function in equation (3.1), represents the net present cost, including capital cost of the power units and their operating cost. In this study, the operating cost covers fuel consumption calculated by a linear function with fixed charges, and fixed start-up and shut-down costs. Net present cost is the sum of the discounted values of all the cost cash flow at the present. Assume the annual discount rate for the calculation is  $r$  and the system life span is  $L$  years. As fuel consumption considered to be the costliest component of operating expenditure in power generation, then the expected net cost value of the project over the system life-span can be minimized as:

$$\min \left[ \sum_i (x_i C_i^g) + y_{wind} C_{wind} + \sum_{L=1}^{N_{life}} \frac{N_d}{(1+r)^L} \sum_{i,t} (A_i u_{i,t} + B_i p_{i,t} + c_{i,t}^{su} + c_{i,t}^{sd}) \right] \quad (3.1)$$

$y_{wind}$  the integer design decision variable for number of wind turbine;

$x_i$  the binary decision variable for installing or not installing the conventional power unit  $I$ ;

$u_{i,t}$  the binary operational/scheduling decision variable representing the on/off status of unit  $i$  at period  $t$ ;

$c_{i,t}^{su}$  start-up cost variable of unit  $i$  in period  $t$ ;

$c_{i,t}^{sd}$  shut-down cost variable of unit  $i$  in period  $t$ ;

$p_{i,t}$  power output variable of unit  $i$  in period  $t$ ;

$A_i, B_i$  coefficients of the fuel cost function of unit  $i$ , their values are listed in Table 3.1

$C_i^g$  the capital cost of  $i$  generating unit (see Table 3.1)

$C_{wind}$  the capital cost of one wind turbine (see Table 3.1)

$\sum_{y=1}^{N_{life}} \frac{N_d}{(1+r)^y}$  coefficient to convert the daily operating cost into net present value, where  $N_d$ ,

denotes number of days per year (365 days/year),  $N_{life}$  represents the life time (i.e., system life span and it was assumed to be 25 years) of the generating units and  $r$  denotes the discount rate (12%) In this study, we assumed that all generating unit are powered by coal and the operating cost for wind turbines are negligible compared to power generating unit operating cost.

Table 3.1. Data for the thermal generating unit [88]

Unit	$P^L$	$P^U$	$A$	$B$	$TU$	$TD$	$Hsc$	$Csc$	$T^{cold}$	$T^{ini}$	$RD$	$RU$
	MW	MW	\$/h	\$/MWh	h	h	\$/h	\$/h	h	h	MW/h	MW/h
1	150	455	960.61	16.479	8	8	4500	9000	5	8	91	91
2	150	455	944.56	17.447	8	8	5000	10000	5	8	91	91
3	20	130	691.13	16.9	5	5	550	1100	4	-5	26	26
4	20	130	670.65	16.817	5	5	560	1120	4	-5	26	26
5	25	162	423.06	20.447	6	6	900	1800	4	-6	32.4	32.4
6	20	80	355.05	22.972	3	3	170	340	2	-3	16	16
7	25	85	477.93	27.827	3	3	260	520	2	-3	17	17
8	10	55	656.49	26.188	1	1	30	60	0	-1	11	11
9	10	55	663.11	27.414	1	1	30	60	0	-1	11	11
10	10	55	668.53	27.902	1	1	30	60	0	-1	11	11

*Minimum and maximum power generation*

To ensure that the power produced by unit  $i$  at time  $t$  is within the power generation limits of that unit. (i.e., upper limit  $P_i^U$  and lower limit  $P_i^L$ ). The values of the upper and lower power generating limits are shown in Table 3.1. These constraints fix units availability at zero when units are ‘off’ ( $u_{i,t} = 0$ ) and specify the lower and upper bounds of units capacity when units are active ( $u_{i,t} = 1$ ).

$$u_{i,t}P_i^L \leq p_{i,t} \leq u_{i,t}P_i^U \quad \forall t = 1, \dots, T; , i = 1, \dots, I \quad (3.2)$$

*Electricity demand and reserve*

Electricity demand should be satisfied at any  $t$  time by Equation 3.4. In this deterministic case, the average profile of Ontario demand for 2018 was used. It was assumed that we want to satisfy a portion of Ontario’s demand (i.e.,~ it was assume that 7% of total Ontario demand in 2018 [89] will be satisfied).

$$\sum_i p_{i,t} + p_{wind,t} \geq D_t \quad t = 1, \dots, T$$



(3.3)

Equation (3.4) guarantees spinning reserve by the available capacity of the active units, where,  $R_t$  represents the reserve requirements. Spinning reserve (i.e., spinning means active units that already connected to the grid) means that from the pool of available capacity, a portion is selected for a back-up role. It is assumed that the spinning reserve requirement to be met is set at 10% of the load demand for each time period.

$$\sum_i P_i^U u_{i,t} \geq D_t + R_t \quad t = 1, \dots, T \quad (3.4)$$

#### *Minimum up and down time of thermal generating units*

Once a decision has been made to turn a conventional power generating unit on or off, it must remain in that state for a minimum amount of time. Equations (3.5)-(3.6) determine the online/offline status of unit  $i$  in its earliest periods of operation which are specified by its initial status ( $T_i^{ini}$ ) and its minimum up ( $TU_i$ ) and down ( $TD_i$ ) times..  $T_i^{ini}$  denotes the number of periods that unit  $i$  has been initially offline ( $T_i^{ini} < 0$ ) or online ( $T_i^{ini} > 0$ ). The following constraints ensure that when the simulation is started if unit  $i$  is offline for  $T_i^{ini}$ , it will continue to be offline until it satisfies its minimum down requirement ( $TD_i$ ) and vice versa for the online unit.

$$u_{i,t} = 1 \quad \forall i : T_i^{ini} > 0; \quad t = 1, \dots, (TU_i - T_i^{ini}) \quad (3.5)$$

$$u_{i,t} = 0 \quad \forall i : T_i^{ini} < 0; \quad t = 1, \dots, (TD_i + T_i^{ini}) \quad (3.6)$$

Equations (3.7) and (3.8) are expressing the constraints on minimum uptime and downtime unit as follows:

$$u_{i,t} - u_{i,t-1} \leq u_{i,t+j} \quad i = 1, \dots, I; \quad t = 2, \dots, T; \quad j = 1, \dots, (TU_i - 1) \quad (3.7)$$

$$u_{i,t+j} \leq u_{i,t} - u_{i,t-1} \quad i = 1, \dots, I; \quad t = 2, \dots, T; \quad j = 1, \dots, (TD_i - 1) \quad (3.8)$$

In the first time period equations (3.7) and (3.8) are reduced to respectively.

$$u_{i,1} \leq u_{i,1+j} \quad T_i^{ini} < 0; \quad i = 1, \dots, I; \quad j = 1, \dots, (TU_i - 1) \quad (3.9)$$

$$u_{i,1+j} \leq u_{i,1} \quad T_i^{ini} > 0; \quad i = 1, \dots, I; \quad j = 1, \dots, (TD_i - 1) \quad (3.10)$$

#### *Unit ramp rates*

Thermal generating units are limited with respect to how quickly they can change their power output and also this limit is known as a unit's ramp rate ( $RU_i$  ramp-up rate,  $RD_i$  ramp-down rate,  $SD_i$  shutdown ramp rate and  $SU_i$  is start-up ramp rate per unit of time period). The ramp-up and ramp-down rates of each unit are set to be at 20% of the unit maximum power output per time period. Whereas the start-up and shutdown ramp rates of each unit are chosen to be at its maximum generation output [90], [91]. The ramp rate limits of unit  $i$  at time period  $t$  are modelled by equations (3.11) and (3.12):

$$p_{i,t} - p_{i,t-1} \leq RU_i u_{i,t-1} + SU_i (1 - u_{i,t-1}) \quad i = 1, \dots, I; t = 2, \dots, T \quad (3.11)$$

$$p_{i,t-1} - p_{i,t} \leq RD_i u_{i,t} + DU_i (1 - u_{i,t}) \quad i = 1, \dots, I; t = 2, \dots, T \quad (3.12)$$

The costs involved in turning on and off generating units are essential and considered to be an important element of the operation cost of power thermal unit. In this study, it is assumed that there are two fixed start-up costs per unit (hot start and cold start), depend on the time periods that the unit was off. The start-up cost function is defined as a hot start cost ( $c_{i,t}^{su} = Hsc_i$ ) if downtime  $\leq (TD_i + T_i^{COLD})$  and a cold start cost ( $c_{i,t}^{su} = Csc_i$ ) otherwise. Where  $Hsc_i$ ,  $Csc_i$  and  $T_i^{COLD}$  are parameters that represent the hot start cost of unit  $i$ , the cold start cost of unit  $i$ , and the cold start hour of unit  $i$ , respectively. It was assumed that there was no cost associated with shutting down the units ( $c_{i,t}^{sd}=0$ ). The values of these parameters are reported in Table 3.1. This start-up cost function can be modelled by equations (3.13) - (3.17):

$$(u_{i,t} - u_{i,t-1})Hsc_i \leq c_{i,t}^{su} \quad i = 1, \dots, I; t = 2, \dots, T \quad (3.13)$$

$$u_{i,1}Hsc_i \leq c_{i,t}^{su} \quad i = 1, \dots, I; T_i^{ini} < 0 \quad (3.14)$$

$$\left( u_{i,t} - \sum_{j < TD_i + T_i^{COLD}} u_{i,t-j} \right) Csc_i \leq c_{i,t}^{su} \quad i = 1, \dots, I; t > TD_i + T_i^{COLD} \quad (3.15)$$

$$\left( u_{i,t} - \sum_{j < t} u_{i,t-j} \right) Csc_i \leq c_{i,t}^{su} \quad i = 1, \dots, I; T_i^{ini} < 0; TD_i + T_i^{COLD} \leq t < TD_i + T_i^{COLD} + 1 \quad (3.16)$$

$$c_{i,t}^{su} \geq 0 \quad i = 1, \dots, I; t = 1, \dots, T$$

(3.17)

*Design constraints*

The following constraints equations (3.18) and (3.19) ensure to install the required thermal unit.  $x_i$  denotes a binary decision variable to determine whether unit  $i$  should be installed or not.

$$u_{i,t} \leq x_i \quad i = 1, \dots, I; t = 2, \dots, T \quad (3.18)$$

$$x_i \leq \sum_t u_{i,t} \quad i = 1, \dots, I \quad (3.19)$$

Equation (3.20) relates the total power produced from all wind turbines ( $P_{wind,t}$ ) with number of wind turbine needed ( $y_{wind}$ ) and the power produced per single wind turbine ( $P_{wind,t}^*$ ) for each time period.

$$P_{wind,t} = y_{wind} P_{wind,t}^* \quad t = 1, \dots, T \quad (3.20)$$

The power delivered by wind single turbine to the electricity grid can be calculated using the following (3.21) [92]:

$$P_{wind}^* = \begin{cases} 0 & , v < v_0 \\ C_p \frac{1}{2} \rho_{air} v^3 A \eta & , v_{max} > v \geq v_0 \\ P_{wind}^{max} & , v > v_{rated} \end{cases} \quad (3.21)$$

Where  $P_{wind}^*$  denotes the electrical power generated by one wind turbine in watt.  $v$  is the actual wind speed in (m/s),  $v_0$  represents the cut-in-speed, the minimum wind speed at which the turbine blades overcome friction and begin to rotate (typically it is 3.5 m/s). Cut-out-speed: it is a wind speed where braking system is employed to bring the rotor to a standstill to prevent the wind turbine from damage. Rated output wind speed ( $v_{rated}$ ), for this speed and above, the wind generator is limited to its maximum design output power  $\eta$  is the wind generator efficiency. The rotor swept area and the air density are represented by  $A$  and  $\rho_{air}$  respectively.  $C_p$  describes the fraction of the power in the wind that may be converted by the turbine into mechanical work. The maximum achievable value of  $C_p$  is 16/27. An industrial wind turbine, *Vestas V90-1.8*, was selected in this study for power production from wind. The specification of this wind turbine can be found in [93]. The power output of this wind turbine as a function of wind speed is illustrated in Figure 3.1.

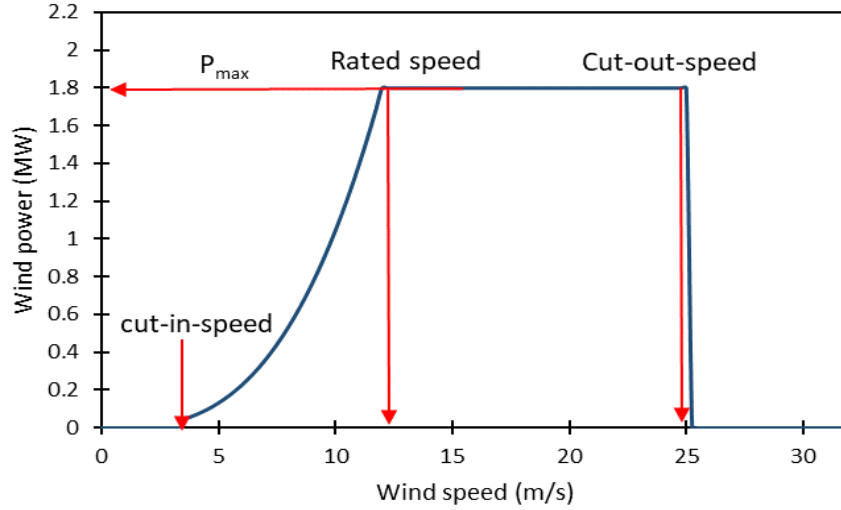


Figure 3.1. Vestas V90-1.8 wind turbine power output as a function of wind speed.

### Variable specification

Finally, the specification on the variables is as follows

$$u_{i,t} \in [0,1] \quad i = 1, \dots, I; t = 1, \dots, T \quad (3.22)$$

$$x_i \in [0,1] \quad i = 1, \dots, I \quad (3.23)$$

$$y_{wind} \geq 0 \text{ and integer} \quad (3.24)$$

$$p_{wind,t} \geq 0 \quad t = 1, \dots, T \quad (3.25)$$

After that, these equations are solved, and results obtained are shown in the next section.

### 3.2.1 Results and Discussions of the Deterministic Power Generation Planning Model

Average wind speed and electrical demand profiles that are used to solve this problem are shown in Figure 3.2 and Figure 3.3. The data for the 10-unit system and wind turbine needed to solve this model are provided in Tables (Table 3.1 and Table 3.2)

Table 3.2. Wind turbine and thermal (conventional) generating unit capital cost and carbon emissions factors

	Thermal Unit	Single wind turbine
Carbon emission factor	820 (gCO <sub>2</sub> eq/kWh) [94]	15(gCO <sub>2</sub> eq/kWh) [94]
Capital cost	3,246 \$/kW [95]	1.75 million \$ [61]

This deterministic model equations (3.1)-(3.25) was implemented in GAMS [15]. The model is solved using the MILP (Mixed Integer-Linear Programming) solver CPLEX which is based on the branch and cut algorithm [96]. The MILP problem contains 251 discrete variables (250 are binary and 1 is integer) and 758 continuous variables. The number of constraints was 2947. The GAMS program executes successfully in 0.2 seconds on an Intel Core i7 commodity personal computer. Design decision results and the value of the objective function are provided in Table 3.3. It shows that the number of generating unit that are needed to be installed along with their capacity. The optimal production (i.e., generation) schedule for the units that have been selected are shown in Figure 3.4. Moreover, the total power produced by all generating unit and the average demand, are plotted as a function of time in Figure 3.5. As it can be noticed from this figure that the simulation the demand is exactly matched with the power supplied from generating units.

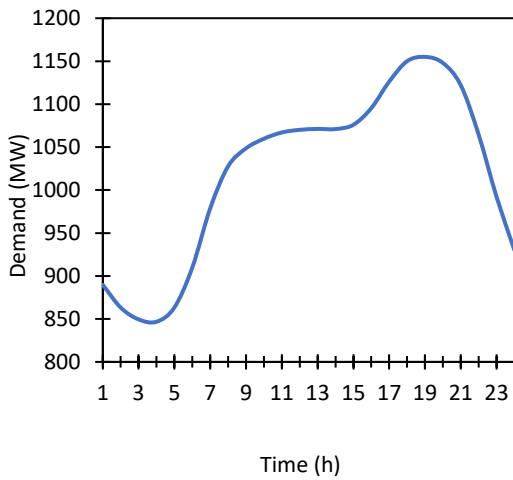


Figure 3.2. Average demand profile

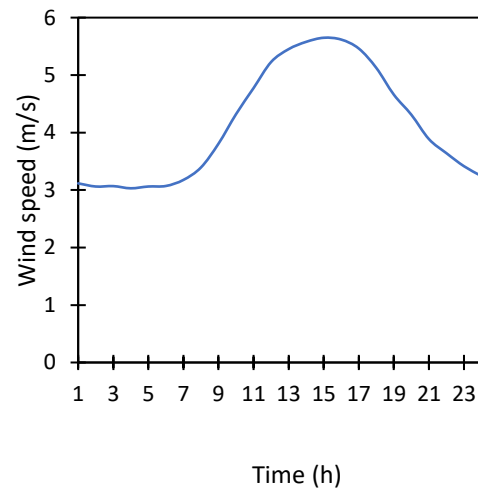


Figure 3.3. Average wind speed profile

Table 3.3. Objective function and design decision results for the deterministic formulation

Number of generating units	Capacity (MW)
2	455
1	130
1	162
1	80

Total thermal generating units	5 with total capacity of 827 (MW)
Number wind turbine	0
Objective function (net present cost)	Total cost: 5.563 billion \$ Capital: 4.161 billion\$ Operating: 1.402billion\$

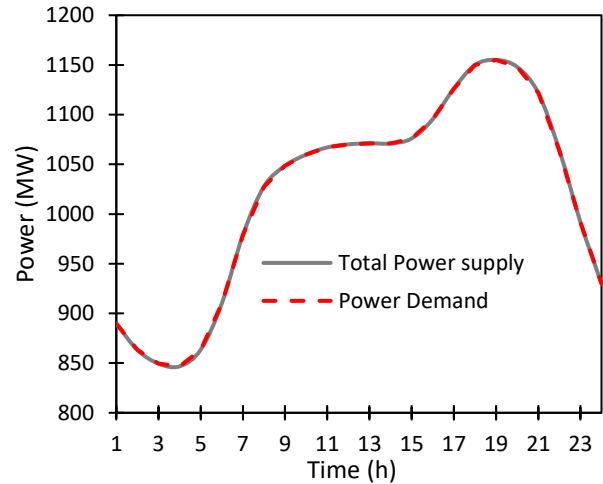
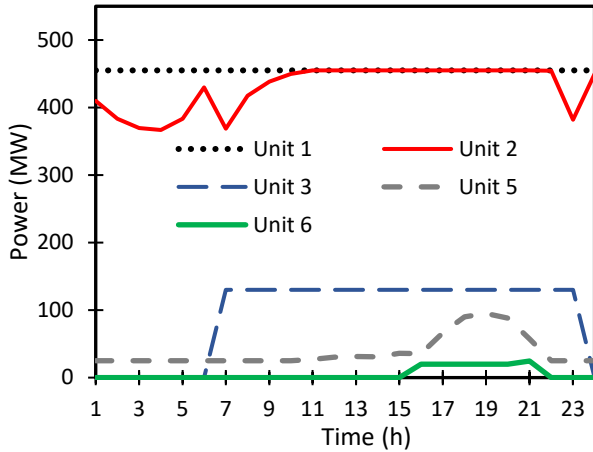


Figure 3.4. Power output results for units in each time

Figure 3.5. Energy scheduling results for all units in each time

### Environmental Considerations

As it can be seen from that according to the current parameter, the optimization program decided not to have wind turbines installed. This is because renewable energy is usually more expensive than the traditional fossil fuel. However, renewable energy resources are clean alternatives to fossil fuel, and it has been widely integrated with current power systems to mitigate Green-House-Gas emissions (i.e., CO<sub>2</sub>, CH<sub>4</sub>, and NO<sub>x</sub>). It is also clear that the essence of this study is to design a system that can integrate renewable energy represented by wind as into conventional power generation plants. Therefore, in order to force the optimization program to have some wind turbine, CO<sub>2</sub> emission constraints is introduced and imposed to the mathematical model as shown in equation (3.26).

$$Em = N_{life} N_d \sum_{i,t} (\beta_i p_{i,t} + \gamma_i p_{wind,t}) \Delta t \leq \alpha \quad (3.26)$$

Where  $Em$  denote the total mass of CO<sub>2</sub> emissions from the power system  $\beta_i$  is the emission factor of thermal unit  $i$ , (kg-CO<sub>2</sub> eq./MWh),  $\gamma_i$  the emission factor associated with wind turbine, (kg-CO<sub>2</sub> eq./MWh) (see Table 3.2)  $\alpha$  is the limits that enforce on the CO<sub>2</sub> emissions. A sensitivity analysis on ( $\alpha$ ), to check validity of our mathematical problem and see if the optimization will force to install some wind turbines, is conducted. Figure 3.6 shows the change of the present cost and the number of wind turbine needed to be installed as a function of CO<sub>2</sub> emission ( $\alpha$ ). As it can be noticed that there are upper and lower limits for ( $\alpha$ ). The upper limits occur at the lowest net present cost. This happens when the emission constraint is not active (same solution as Table 3.3) and the number of wind turbines needed are zero. After that, when the value of  $\alpha$  decreases the objective function (net present cost) increases and at the same time, the optimization program forces the installation of wind turbines. The greater the reduction in the amount of CO<sub>2</sub> emissions, the higher the number of wind turbines that are decided to be installed and at the more expensive objective function. It is worth noticing that there is a lower limit on the value of ( $\alpha$ ) where if we reduce it, there is no feasible solution. This is the minimum value of emission value that can be obtained for current problem conditions. In this study, it was proposed to reduce the CO<sub>2</sub> emission ( $\alpha$ ) by 20% from its upper limit (the CO<sub>2</sub> at the lowest cost when no environmental regulation is considered).

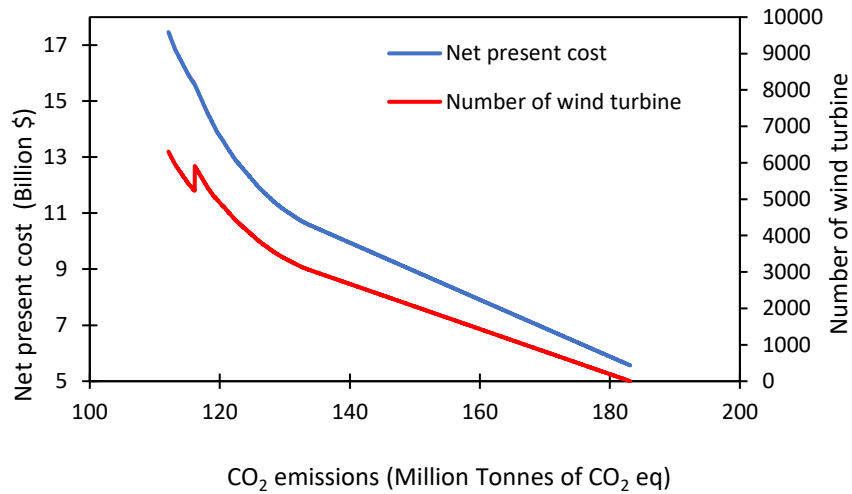


Figure 3.6. The effect of CO<sub>2</sub> emissions on the objective function (blue line) and number of wind turbine needed to be installed (red line)

### 3.3 Stochastic Data-Driven Design and Operation of Power Generation Planning model

#### 3.3.1 Stochastic Model Formulation

This section discusses the model for the stochastic problem for the design and operation of a power generation plant under uncertain demand and wind speeds. The mathematical model is formulated as a two stage stochastic with recourse, where the first-stage decisions decide the existence of the thermal generating unit and wind turbine while the second-stage decisions plan the operation of the system (i.e., power scheduling). One main difference of stochastic power planning model compared to the deterministic one, is that the optimal power scheduling can be different for each different realization of the uncertainty (in each different clusters of demand or wind speed) in the system. The two stage stochastic recourse formulation is basically bi-level optimization formulation whose inner optimization problems mimic the second stage planning process. As mentioned before (Model with Recourse section 2.1.4), due to special structure, two stage stochastic programs can be naturally reformulated into an equivalent single-level optimization problem. Therefore, the single-level optimization formulation of two stage recourse of power generation model can be directly written as follows:

##### *Objective Function*

The objective function (equation(3.27)), represents the net present cost of the stochastic power generation planning model. The second part of the equation denotes the annual net cost from operating the generation planning model (i.e., basically fuel consumption because of power generation), which depends on the scenario of uncertainty realization  $s$  with probability  $Prob_s$ :

$$\min \sum_i x_i C_i^g + y_{wind} C_{wind} + \sum_{L=1}^{N_{life}} \frac{N_d}{(1+r)^L} \sum_{i,t,s} Prob_s (A_i u_{i,t,s} + B_i P_{i,t,s} + c_{i,t,s}^{su} + c_{i,t,s}^{sd} + G^{wind} p_{wind,t,s}) \quad (3.27)$$

The remaining equations are almost the same as the deterministic one, except the new subscript  $s$ . where the new subscript  $s$  [  $\{1, \dots S\}$  ] is used in the stochastic model for all the variables and parameters whose values may be different in each stochastic scenario  $s$  (the uppercase  $s$  denotes the total number of scenarios).



Minimum and maximum Power generation

$$u_{i,t,s}P_i^L \leq p_{i,t} \leq u_{i,t,s}P_i^U \quad \forall t = 1, \dots, T; , i = 1, \dots, I; s = 1, \dots, S \quad (3.28)$$

Electricity demand and reserve

$$\sum_i p_{i,t,s} + p_{wind,t,s} \geq D_{t,s} \quad t = 1, \dots, T; s = 1, \dots, S \quad (3.29)$$

$$\sum_i P_i^U u_{i,t} \geq D_{t,s} + R_{t,s} \quad t = 1, \dots, T; s = 1, \dots, S \quad (3.30)$$

Minimum up and down time of thermal generating units

$$u_{i,t,s} - u_{i,t-1,s} \leq u_{i,t+j,s} \quad i = 1, \dots, I; t = 2, \dots, T; j = 1, \dots, (TU_i - 1); s = 1, \dots, S \quad (3.31)$$

$$u_{i,t+j,s} \leq u_{i,t,s} - u_{i,t-1,s} \quad i = 1, \dots, I; t = 2, \dots, T; j = 1, \dots, (TD_i - 1); s = 1, \dots, S \quad (3.32)$$

Unit ramp rates

$$p_{i,t,s} - p_{i,t-1,s} \leq RU_i u_{i,t-1,s} + SU_i(1 - u_{i,t-1,s}) \quad i = 1, \dots, I; t = 2, \dots, T; s = 1, \dots, S \quad (3.33)$$

$$p_{i,t-1,s} - p_{i,t,s} \leq RD_i u_{i,t-1,s} - DU_i(1 - u_{i,t,s}) \quad i = 1, \dots, I; t = 2, \dots, T; s = 1, \dots, S \quad (3.34)$$

Start-up and shut-down unit costs

$$(u_{i,t,s} - u_{i,t-1,s})Hsc_i \leq c_{i,t,s}^{su} \quad i = 1, \dots, I; t = 2, \dots, T; s = 1, \dots, S \quad (3.35)$$

$$u_{i,1,s}Hsc_i \leq c_{i,t,s}^{su} \quad i = 1, \dots, I; T_i^{ini} < 0; s = 1, \dots, S \quad (3.36)$$

$$\left( u_{i,t,s} - \sum_{j < TD_i + T_i^{COLD}} u_{i,t-j,s} \right) Csc_i \leq c_{i,t,s}^{su} \quad i = 1, \dots, I; t > TD_i + T_i^{COLD}; s = 1, \dots, S \quad (3.37)$$

$$\left( u_{i,t,s} - \sum_{j < t} u_{i,t-j,s} \right) Csc_i \leq c_{i,t,s}^{su} \quad i = 1, \dots, I; T_i^{ini} < 0; TD_i + T_i^{COLD} \leq t < TD_i + T_i^{COLD} + 1; s = 1, \dots, S \quad (3.38)$$

$$0 \leq c_{i,t,s}^{su} \quad i = 1, \dots, I; t = 1, \dots, T; s = 1, \dots, S \quad (3.39)$$

Design Constraints

$$u_{i,t,s} \leq x_i \quad i = 1, \dots, I; t = 2, \dots, T; s = 1, \dots, S \quad (3.40)$$

$$x_i \leq \sum_t u_{i,t,s} \quad i = 1, \dots, I ; s = 1, \dots, S \quad (3.41)$$

$$p_{wind,t,s} = y_{wind} P_{wind,t,s}^* \quad t = 1, \dots, T ; s = 1, \dots, S \quad (3.42)$$

### 3.3.2 Data-Driven Uncertainty Scenario Construction Using Clustering

In this section, a method on how to generate scenarios is presented from the given historical data (uncertain parameter) using clustering algorithm. These scenarios with its corresponding occurrence can be used as inputs parameter for the stochastic model. This method begins by collecting historical data of the attribute or uncertain parameter under study (in this study it was wind speed and electrical demand). Following which, the raw input data must be pre-processed into the right format. The raw data time-series (i.e., electricity demand and wind speed) are first processed (arranged) into the candidate periods (considered to be 365-days for 1-year, with each day consisting of 24 hours). This reordering process is shown in the matrix presented in Figure 3.7, in which the number of columns is defined by the multiple of the number of time steps (i.e., 24 hour), and number of rows corresponding to the number of periods (i.e., 365-days). A single row represents a candidate period (i.e., one day). Raw data of both electricity demand and wind speed are first restructured into new matrix where the number of rows represent the number of days in one year (i.e., 365 days) and the number of columns represent the number of hours in one day (i.e., 24 hours).

As it can be noticed in the demand data (Figure 3.8) there are some sort of repeating pattern in daily basis while for wind data a greater degree of unpredictability can be seen in Figure 3.9. However, the purpose of this case study is to design and have some insight on the uncertain parameters to optimally plan for future operation which will be based on the most probable behaviour. Therefore, clustering would be good also for data that experience some sort of randomness because it can present dense data centroids (representative points / trend) that most likely can happen as it can be seen in Figures (Figure 3.10 and Figure 3.11).

The benefit of using this method is that, instead of using the whole set of data only the centroids (i.e., centres of each cluster) of each uncertain parameter will be used. These centroids can represent the whole set of data. Based on the matrix introduced in Figure 3.7, k-mean [37] clustering algorithm was applied to group the independent candidate periods (i.e., each row/day of the processed data matrix) into clusters. Accordingly, representative periods are derived. Each

cluster/group of each uncertain parameter is represented by a representative profile (i.e., curve). These representative profiles are the centres of each cluster (see K-Means Clustering (2.3.1)) Each uncertain parameter (e.g., demand, wind speed, solar intensity, fuel supply) will be represented by several clusters, and each cluster corresponding to one scenario with a certain probability of occurrence. The probability of occurrence corresponds to the weight of each cluster. Figure 3.12 shows the process of scenario construction for stochastic optimization using clustering machine learning algorithm. In order to determine the most applicable cluster number needed to divide each set of data (i.e., attribute), the k-mean clustering algorithm was applied to the two set of processed data (i.e., electricity demand and wind speed) using different number of clusters.

The clustering algorithm was applied to each attribute (wind speed and electricity demand) using a different number of clusters. Figures (Figure 3.13-Figure 3.16) show the error average and standard deviation as function of the cluster number for both electricity demand and wind speed. The following relative error function (equation(3.43)) was employed as a validation metric between the representative cluster profile (centre) and the actual processed data (i.e., candidate period or row/day) which correspond to that cluster.

$$error_{c,d,h}(\%) = \frac{Cl_{c,h} - paramater_{d,h}}{paramater_{d,h}} * 100 \quad (3.43)$$

Where  $Cl_{c,h}$  is the cluster curve  $c$  at hour  $h$ . An ideal cluster would have a minimum error average with a minimum standard deviation. The error and standard deviation average in clustering appears to drop as the number of clusters increases until they reach a certain value and after that it starts to oscillate as it can be seen in these figures. This reveals that there is some sort of optimal number of representative curves (i.e., cluster) for each data set (i.e., wind speed or electricity demand).

Therefore, the number of clusters (i.e., cluster centre and its corresponding weight/ probability) with the minimum error average were selected as scenarios that represent the uncertain parameter of the stochastic power generation model. Figure 3.10 and Figure 3.11 show actual data and cluster centres used as representatives of these data for both electricity demand and wind speed respectively. As it can be noticed in these figures, clustered results are in good agreement with demand data, however, for wind speed data, we can say it mostly follows the trend and the centres are located around the most probable occurring wind speed data. It is worth noting that the error associated with wind speed is higher as they are more randomized and there is no clear pattern for their distribution.

As it can be seen from figures (Figure 3.13 - Figure 3.16) that the minimum error happens when the number of clusters is 9 for electricity and 5 for wind speed. Figure 3.17 and Figure 3.18 show the clustering results that will be used for the stochastic power planning model. By comparing the demand clustering profiles with its actual daily profile, we can say that the clustering results are following the actual demand profile. On the other hand, we can see that for wind there are five possible trends based on the most probable occurrence of wind data. The traditional approach used by power system modeler to represent a 1-year power demand data by 8 curves which comprise of weekday and weekend demand for the four seasons throughout the year. In this study it was found out that 9 clusters can sufficiently represent the electrical demand which is in good agreement with common practice.

In this study, each scenario is represented by a combination of one cluster that represents wind speed and another that represents electricity demand. All scenarios are formed by all possible combinations of both uncertain parameters. Therefore, if we found that 5 clusters can represent wind data and 9 clusters are enough to represent electricity demand, the total number of scenarios will be 45. Similarly, the probability of each scenario is calculated by multiplying the weight of the cluster (see K-mean clustering, section 2.3.1)) of each uncertain parameter with each other (i.e., electricity demand and wind speed) that form this scenario. By this way, we are imposing that both uncertainties are independent of each other. Previous studies [97] used clustering representative curves and combine more than one attribute in one cluster, which make attributes dependent on each other. By this method the occurrence of any attribute is independent of others. These previous studies tackle another aspect of size reduction. They focused on the problems associated with the integration of multiscale modelling as will be discussed in the next chapter. However, this chapter concentrated on dealing with data from the perspective of uncertainty.

$$\boxed{
 \begin{array}{l}
 \text{parameter}_{8764} = \begin{pmatrix} \text{parameter}_1 \\ \text{parameter}_2 \\ \vdots \\ \text{parameter}_{8764} \end{pmatrix} \xrightarrow{\text{rearrange}} \begin{pmatrix} \text{parameter}_{1,1} & \cdots & \text{parameter}_{1,24} \\ \vdots & \ddots & \vdots \\ \text{parameter}_{366,1} & \cdots & \text{parameter}_{366,24} \end{pmatrix}
 \end{array}
 }$$

Figure 3.7. Process of rearranging the dimension of wind speed and electric demand

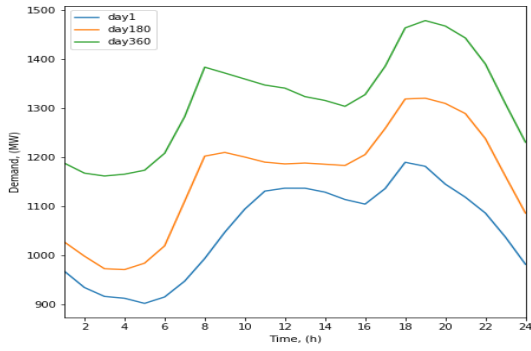


Figure 3.8. Demand profile for selected days

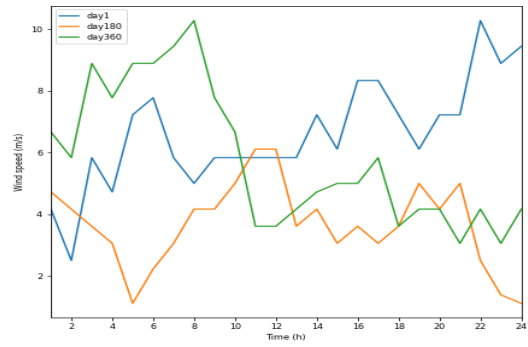


Figure 3.9. Wind speed profile for selected days

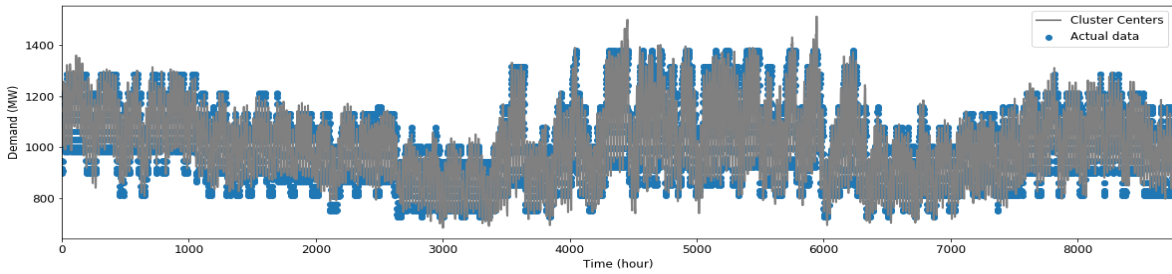


Figure 3.10. Actual electricity demand and its computed cluster centres for 1-year time horizon

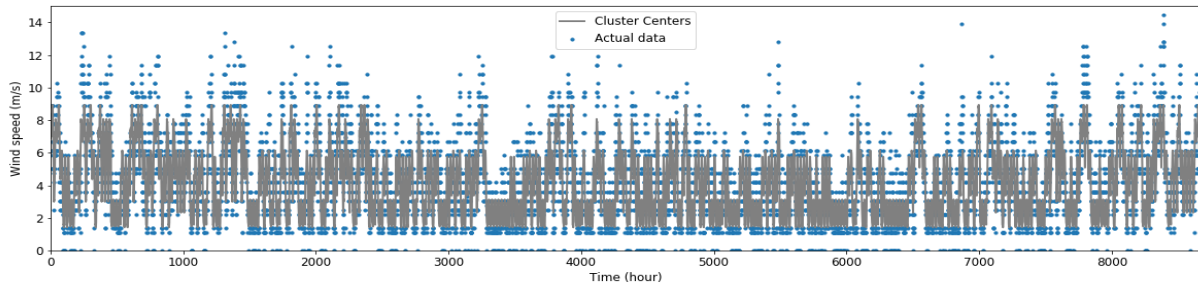


Figure 3.11. Actual wind speed and its computed cluster centres for 1-year time horizon

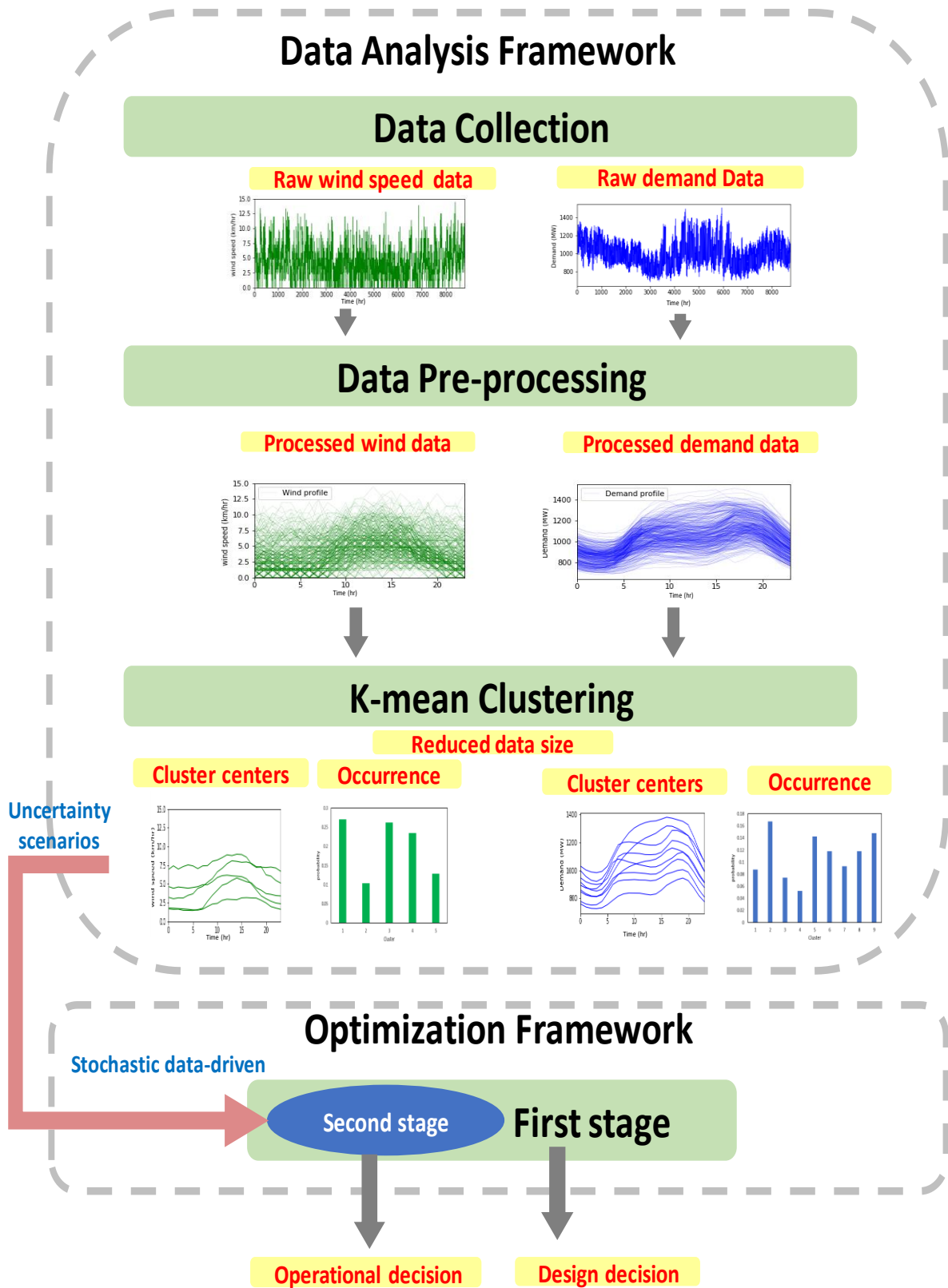


Figure 3.12. Scenario generation for the stochastic data-driven power generation planning model

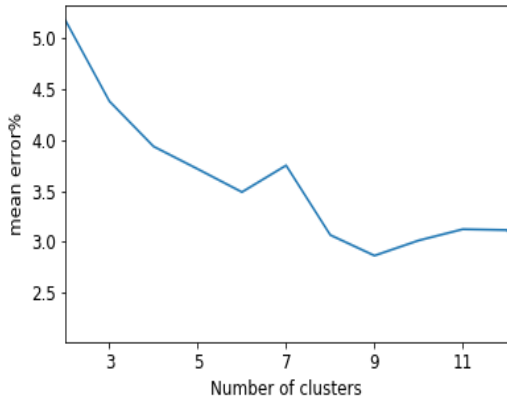


Figure 3.13. Effect of cluster number on the average error for electricity demand

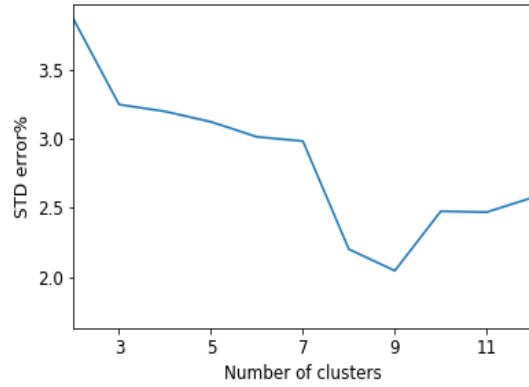


Figure 3.14. Effect of cluster number on the average standard deviation for electricity demand

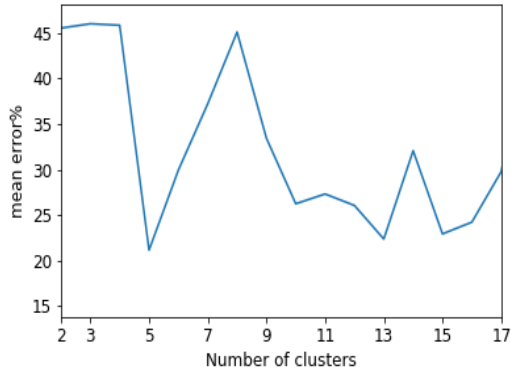


Figure 3.15. Effect of cluster number on the average error for wind speed

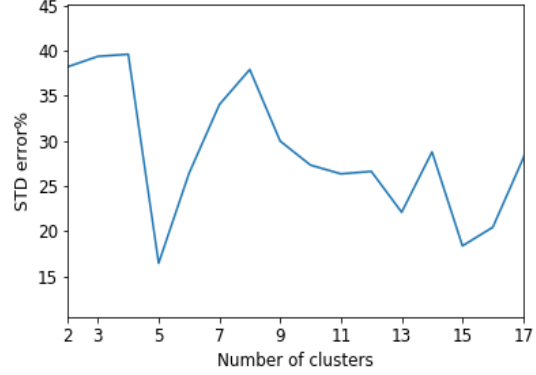


Figure 3.16. Effect of cluster number on the average standard deviation for wind speed

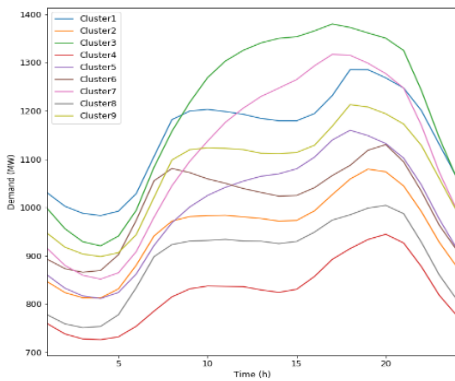


Figure 3.17. Electricity demand clusters

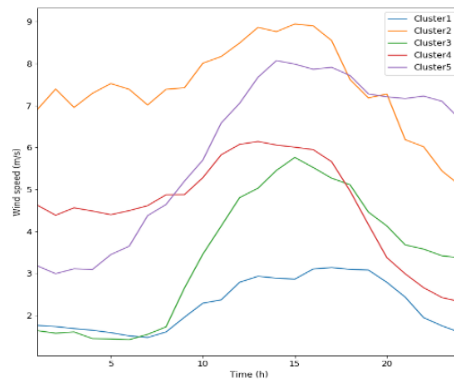


Figure 3.18. Wind speed clusters

### **3.3.3 Comparison Between Stochastic and Deterministic Results When There Are No Environmental Considerations**

It was discussed in the previous section that 9 clusters (i.e., scenarios) were chosen to represent the uncertain demand (Figure 3.17) and 5 clusters were chosen to represent the daily wind speed behaviour (Figure 3.18). Accordingly, the total number of scenarios for the stochastic model were 45 (i.e., 9 x 5). This stochastic model (equations (3.27)-(3.42) was implemented in GAMS [15]. The model was solved using the MILP solver CPLEX. The stochastic problem contains 10811 discrete variables (10810 are binary and 1 is integer) and 22681 continuous variables. The GAMS program executed successfully in 145.687 seconds.

Table 3.4 shows the design results and the objective function values of three different cases, namely, deterministic, stochastic and the worst-case scenario. The worst-case scenario was generated by assuming that there was no wind and tackling the maximum representative demand day for the whole year (extreme demand). It was calculated by taking the maximum demand value of each column (i.e., each time step; hour) of the processed matrix (Figure 3.7) for the whole period (365 days).

As it can be noticed in Table 3.4 that deterministic approach gives the lowest expensive solution whereas, the worst-case is the most expensive. However, the deterministic approach has the lowest reliability because it was designed for certain parameters (e.g., demand) which made it incapable of resolving most real-case scenario problems. The stochastic approach is more expensive than the deterministic because there is a price for the uncertainty. The stochastic approach is designed for the most likely scenarios while the worst-case is designed for the extreme demand which rarely occurs. The development of the system under the worst-case scenario will cause the system to be over-designed and therefore, the full capacity of the power generation will not be of use or rarely used.

Table 3.5 shows the difference if we are using the stochastic design solution with external electricity supply when the demand is extreme in the worst-case scenario. We assumed that the extra electricity required by the extreme demand will be supplied by an external power provider with very expensive price (the levelized cost was assumed to be 300 \$/MWh, which is double the levelized electricity cost reported by EIA for coal in 2018). If we assume that 20% of the year will be subjected to extreme demand, the extra cost that will be added to the stochastic solution is \$ 0.293 billion as it can be seen in Table 3.5. Therefore, the total cost of the stochastic solution with



external electricity needed for extreme demand is less than if we design the power generation plant for worst-case. We can say from this analysis that the design and operation under stochastic approach is more practical than designing under extreme case (i.e., worst-case scenario) that rarely happens.

Table 3.4. Comparison between deterministic, stochastic and worst-case solution of power generation model without environmental consideration

<b>Deterministic</b>		<b>Stochastic</b>		<b>Worst-case scenario</b>	
<b>Number of generating units</b>	<b>Capacity (MW)</b>	<b>Number of generating units</b>	<b>Capacity (MW)</b>	<b>Number of generating units</b>	<b>Capacity (MW)</b>
2	455	2	455	2	455
1	130	2	130	2	130
1	162	1	162	1	162
1	80	1	80	1	85
		2	55	1	80
				3	55
Total thermal generating units	5 with total capacity of 1282 (MW)	8 with total capacity of 1522 (MW)		10 with total capacity of 1662 (MW)	
Number wind turbine	0	0		0	
Objective function (net present cost)	Total cost: 5.56 billion \$ Capital: 4.16 billion \$ Operating: 1.40 billion \$	Total cost: 6.35 billion \$ Capital: 4.94 billion \$ Operating: 1.41 billion \$		Total cost: 7.37 billion \$ Capital: 5.40 billion \$ Operating: 1.97 billion \$	

Table 3.5. Comparison between stochastic solution with external electricity supply and worst-case scenario objective function of power generation model without environmental consideration

	<b>Stochastic solution with external electricity supply</b>	<b>Worse-case scenario</b>
Total cost	6.64 billion \$ 6.35 billion \$ (stochastic solution) 0.29 billion \$ (extra needed when 20% of year demand is extreme)	7.37 billion \$

### **3.3.4 Comparison between Stochastic and Deterministic Results When There Are Environmental Considerations**

In Table 3.6, the objective function of the deterministic and stochastic formulation of the power system model when the environmental constraint was active are reported. It was mentioned before that the value of CO<sub>2</sub> emission ( $\alpha$ ) was set to be 20% less than its upper limit. As it can be observed in Table 3.6, the stochastic solution is less expensive than the deterministic solution. Although considering that the demand uncertainty has a negative effect on the objective function (demand uncertainty should increase the price the objective function) as can be seen in Table 3.4 (stochastic solution is more expensive than deterministic solution) when there is no environmental consideration. This is because in the deterministic model only the expected values of wind speed were used while for the stochastic model the clustered wind speeds were used. It is clear that the average wind speed does not reflect the reality of the wind behaviour and therefore we are not able to fully utilize the benefits from this energy source. In other words, the average (i.e., expected values) wind speeds are not true representatives of the annual wind data. Consequently, the decisions that had been taken via deterministic formulation (i.e., optimization answers) are missing because it relies on a relatively small segment of information (average wind speed), that does not sufficiently explain the real wind speed behaviour. Therefore, it can be said that wind uncertainty has a stronger effect on the optimization solution when environmental regulations were considered, as seen in Table 3.6.

### **3.4 Application of the Proposed Clustering Approach to Generate Stochastic Scenario for Day-Ahead - Real Time UC Power Generation Planning Model**

The aforementioned clustering method was applied to reduce the number of wind scenario in a power system model that was developed by Schwele *et. al.*, [98]. The model developed by [98] was intended to solve for capacity expansion planning, while in this case the model is modified to solve for capacity planning, where the goal is to design (select generators) and schedule (UC and power output decisions) in day-ahead and real-time operation. The model is UC generation planning under day-ahead and real-time operations. The model is a two stage stochastic programming problem with fixed recourse. The first stage determines which power unit to install

(predetermined 4 generators are given and the optimization will choose which are the best), as well as daily day-ahead UC and dispatch decisions under uncertainty of actual wind realization. While in the second stage, the real-time operations of wind power imbalances are adjusted under each scenario. This allows corrective (i.e., recourse) measures to be taken considering the actual wind power generation. More specifically, in real-time operation, the power generator units that determined to be on in day-ahead operations, their power outputs will be adjusted according to the actual wind power generation (uncertainty realization of wind power). The renewable portfolio standard factor was set to 20%, which means 20% of energy mix should come from wind energy. More details on this model formulation can be found [98]. Equations representing this model are portrayed in Appendix A. The developed system by [98] was further simplified, and only 4 nodes are considered with four candidate generators and one load at node 4. The graphical representation of the understudy power system nodes is presented in Figure 3.21.

The 50 wind scenarios were clustered and the minimum clustering error, based on equation (3.43), was found to be at 6 clusters. Therefore, the 50 scenarios were reduced using k-mean clustering to 6, and both were used to solve the power planning model. Figure 3.19 displays the 50 per-unit wind realization factor for existing wind farm  $k1$ , while Figure 3.20 represents the reduced order per-unit wind realization factor. Table 3.7 below shows the objective function values, installed capacity of conventional generators and wind farm, for full-size model and reduced size model, that implemented clustering. It can be seen that both model results are very close, the reduced size model underestimated objective function by only less than 2%. All design decisions of reduced size model are in good agreement with the full-size model. Figure 3.22 shows the day-ahead dispatched decision of both models. As it can be seen from this figure that the operational decision of the reduced model is different by small margin than the full-size model. Although, there is slightly different in operational decisions, the overall performance of the proposed method is acceptable especially for long term planning. As the proposed method will capture the most probable trends of the uncertain data behaviour (pattern) and will transfer that information to the long term stochastic planning models. Therefore, decisions of stochastic design and operation models can benefit from the captured information on uncertainty, and perform satisfactory planning decisions at lower computational expenses.

Table 3.6. Comparison between deterministic and stochastic solution of power generation model with environmental consideration

Deterministic		Stochastic	
Number of generating units	Capacity (MW)	Number of generating units	Capacity (MW)
2	455	2	455
1	130	2	130
1	162	1	162
1	80	1	80
		2	55
Total thermal generating units	5 with total capacity of 827 (MW)	8 with total capacity of 1,522 (MW)	
Number wind turbine	2269	1522	
Objective function (net present cost)	Total cost: 9.30 billion \$ Capital: 8.13 billion \$ Operating: 1.15 billion \$	Total cost: 8.77 billion \$ Capital: 7.60 billion \$ Operating: 1.17 billion \$	

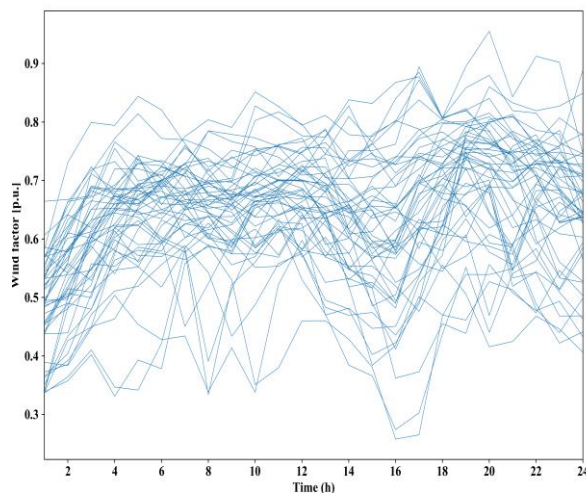


Figure 3.19. 50 scenarios per-unit wind realization factor for existing wind farm [98]

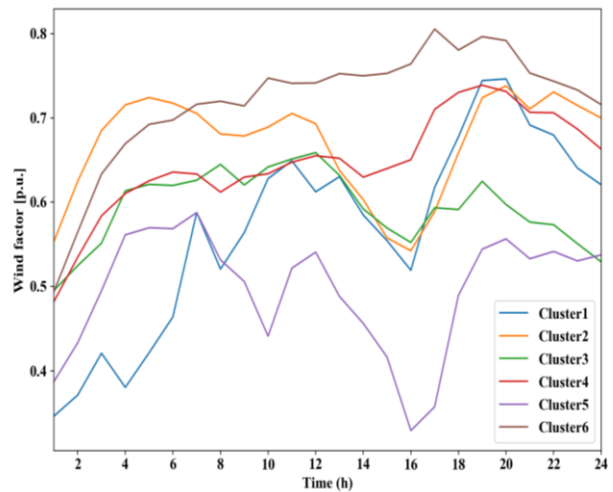


Figure 3.20. 6 clusters per-unit wind realization factor for existing wind farm

Table 3.7. Comparison between full-size and reduced size solution of stochastic power generation planning model in day-ahead and real time stages

	Full-size UC day-ahead real-operation	Reduced size day-ahead real-time
Total cost (Millions)	107.578	105.436
Installed capacity of all conventional generators (MW)	526.000	526.000
Selected generator	g1(450MW) ,g3(76)	g1(450MW), g3(76MW)
Installed capacity of wind farm (MW)	137.5	131.9
Calculated renewable portfolio	21%	20.04%

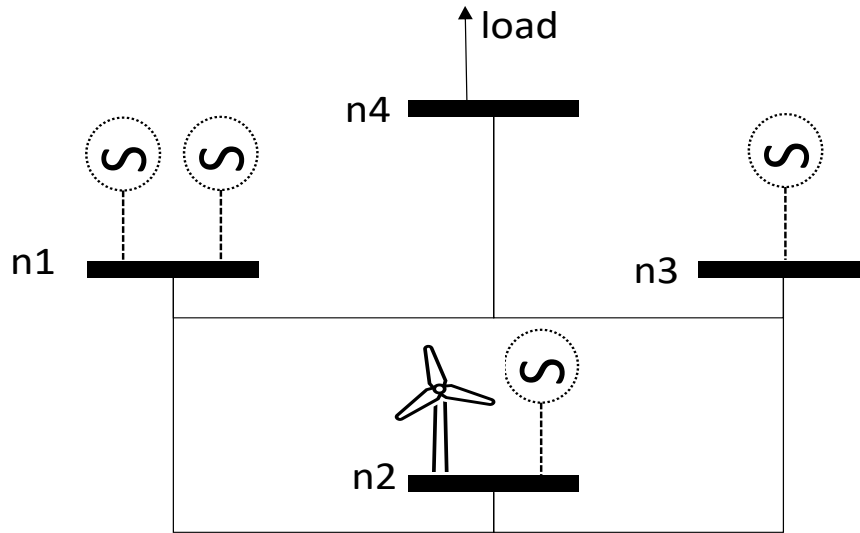


Figure 3.21. Schematic representation of the four-node power system

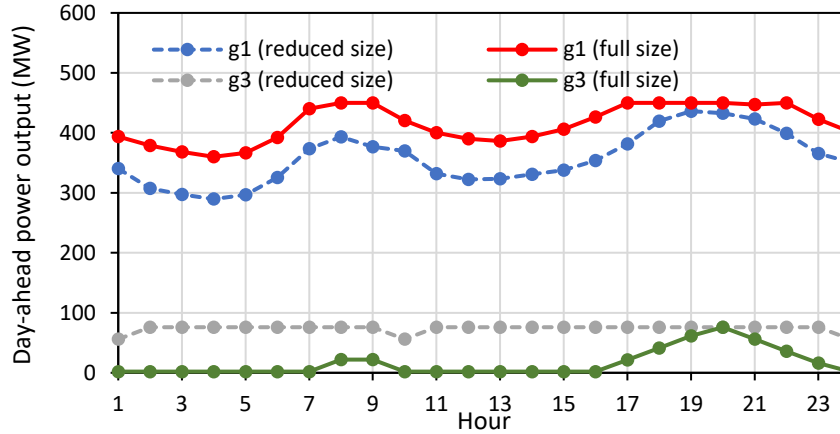


Figure 3.22. Dispatch day-ahead decisions for full-size and reduced size UC planning model

### 3.5 Conclusion and Future Work

A deterministic and stochastic data-driven power generation planning model were developed. The deterministic approach was solved based on a single population parameter (i.e., mean) from data, which does not perfectly cover the data behaviour, and consequently, its solution is unreliable. Renewable energy penetration was enforced by implementing a GHG (e.g., CO<sub>2</sub>) emission regulation. Under deterministic approach, the effect of including the average wind energy to the power generation mix has positive effect on the GHG emissions and conversely, negative effect on the overall net present cost. On other hand, the stochastic data-driven approach was based on more detailed information from the data without explicitly knowing its distribution and therefore, its solution was more reliable. In the stochastic data-driven approach, instead of using the whole set of available data or large number of realization scenarios, that will lead to expensive computational problem, clustering approach was applied to generate reduced size scenarios (i.e., clusters) from the historical available data. Therefore, important characteristics of these uncertain parameters were transferred to the stochastic planning model through these clusters.

It was shown that power generation design and operation under stochastic approach is more practical than designing under both; extreme case (i.e., worst-case scenario), and deterministic case, especially when environmental regulations were imposed. As planning the system under worst-case scenario will cause the system to be over-designed and planning it under deterministic approach will result in decisions that are lacking, because it relies on a relatively small segment of information. It was demonstrated from applying the proposed method to other UC power

generation model, in which operational decisions are made in day-ahead and real-time, that results under reduced size uncertain scenarios are very close to the full-size model. Consequently, satisfactory planning decisions can be achieved at lower computational expenses.

It can be concluded that reducing the uncertain data size by implementing k-means<sup>33</sup> clustering method, is an effective tool to tackle the computational tractability of considering many stochastic scenarios. In addition to that, the proposed method does not require a full understanding of the data behaviour. Meanwhile, it offers a simple framework that can give acceptable results. Therefore, the data-driven stochastic method is a trade-off between computational effort and data accuracy.

In the future, this power generation planning model can be further expanded to include different types of generation units, powered by different types of fuel, and investigate the effect on design and operational decisions. Moreover, the integration of solar energy can be examined. A storage block can be added and the benefits of adding energy storage system under different realization of intermittent renewable energy and demand can be studied. Moreover, a carbon capture unit can be added to the power generation plant, and the whole power and carbon emission capturing unit can be optimized under uncertain wind and demand data.

# **Chapter 4      Clustering Approach for the Efficient Solution of Multiscale Stochastic Programming Problems: Application to Energy Hub Design and Operation Under Uncertainty**

## **4.1 Introduction**

The increase in the share of renewable energy generation expands the implementation of distributed energy resource and encourages the move towards developing and modelling smart energy network and energy hubs [99]. As such, energy hubs are particularly useful for enabling the integration of intermittent renewable energy sources such as solar and wind. An energy hub is a multi-carrier energy system consisting of multiple energy conversion, storage, which is designed to meet different sources of demands such as electric, cooling, and heating demands [100]. As a result, different energy carries in an energy hub can be stored and transferred through different energy conversion units, which enables greater flexibility in energy delivery. The modelling concept of an energy hub quantify the relation between input and output energy flows and can be extended to determine the required capacity of each unit. The energy hub model is typically used as a platform/framework to optimally plan (design) and operate energy systems [101], [102]. Controllable and flexible energy system is established due to the ability of the energy hub system to integrate different types of distributed renewable energy resources (DRERs) and energy storage system (ESSs) [103]. However, due to uncertainty and variability of DRERs (e.g., solar photovoltaic and wind turbine), the advantages of energy hub system to supply flexible power could be limited and diminished [104]. Therefore, modelling the energy hub by considering the uncertainties associated with these sources is crucial.

Energy hub models can be applied to different spatial scales (i.e., from the level of a single building to a larger geographic region), as well as a different time scale. Particularly, energy hub modelling could be applied to different time scales from long term planning (e.g., designing and sizing energy conversion and storage unit) to mid/ short term planning (scheduling and operation). Typically, planning and scheduling are both performed separately even though they're interdependent. However, the integration between these different time scales is the key to improve efficiency and profit margins. As, the integration of planning (e.g., design) and scheduling (i.e., operation) improves decision level management which results in lower net cost. However, the computational



tractability arising from this integration makes it difficult to solve. For example, a very large and intractable problems will be formed if different time scales of the multiscale energy hub model are converting to the shortest planning period (i.e., detailed scheduling over a long duration). While relaxing some constraints or employing surrogate models or using averaging method might lead to infeasible operations (i.e., since detailed schedules cannot be obtained to meet the planned production targets) or inaccurate system design [97], [105].

Various modelling and solution approaches have been proposed to overcome this problem. Clustering arises as an effective and appropriate approach to handle such problems by aggregating similar inputs, such as: supply, demand or price; together. Input parameters typically are made up of multiple attributes like simultaneous electricity and heat demands. The task of clustering is to discover structure in unlabelled data sets by grouping the data into homogenous groups in which the similarity within-group-object (i.e., within one cluster) is minimized while the between-group-object dissimilarity is maximized (i.e., between different clusters). Therefore, application of clustering approach to tackle this problem can significantly reduce the model size and improve computational tractability while maintaining solution accuracy. For more than 50 years, clustering has been broadly studied throughout various disciplines. [106]. A crucial role in clustering algorithm developments is played by mathematical programming. Many research studies had purposed different clustering approach based on the mathematical programming concept such as [97], [107]–[111].

This chapter attempts to address the following challenges (multiscale decision making, uncertainty and variability of DRERs) associated with energy hub by:

1. Applying general mathematical programming-based clustering methods to reduce the multiple attribute demand data size.
2. Proposing a statistical method that models the uncertain behaviour of renewable energy sources.
3. Formulating the energy hub system as two stage stochastic optimization.

Many research studies have been devoted to optimal planning and scheduling of energy hub systems. Moghaddam *et al.*, (2016) [71] performed daily scheduling of an energy hub including different generation and storage technologies. In the studies conducted by (Majidi *et al.*, 2017; and Nojavan *et al.*, 2018) [72], [73], the operation and emission costs of energy hubs were optimized based on two-objective estimation problems. In another study [74], the daily operational costs, including costs of purchased electricity/gas and carbon emission costs were minimized using

deterministic MILP. In [75] paper, the authors proposed a multi-objective optimization framework for optimal operation of energy hub components considering uncertain behaviours of households. The operation of energy hub networks was explored in several papers; however, few studies carried out the design and operation of urban energy systems based on the energy hub concept [101]. The optimal design and operation of DERs has been studied by [112]. However, the study did not consider renewable energy technologies. Research done by [113] developed a deterministic MILP based approach for optimal energy hub operation to support the hydrogen economy. The design decision variables considered in this study are variables limited to hydrogen refuelling station (i.e., hydrogen production and storing facilities. In [101] study, the authors proposed a deterministic design and operation of distributed energy systems (DESs) in urban areas with renewable energy sources. In this study, economic and environmental considerations were investigated, however, the uncertainties of renewable resources were not considered. A study conducted by [114] presented a multi-objective optimization process to determine the optimal design and operation of combined heat and power (CHP) units hub taking into consideration both economic and environmental factors. In [99] study, a stochastic programming approach for the planning and operation of a power to gas energy hub was developed. This study focusses on assessing the benefits power-to-gas energy storage, while accounting for uncertainty in hourly electricity price, the number of fuel cell vehicles serviced, and the amount of hydrogen refuelled. Time series aggregation based on clustering algorithms was used to cluster demand, wind speed and solar irradiance in [105] study. These aggregated/ clustered input data were applied to different energy hub systems to tackle the design and operational optimization problems of these systems. The study shows that the application of clustering methods in energy hub optimization significantly reduce the model complexity.

The energy optimization models introduced in the aforementioned studies did not consider the uncertainties of DRERs, which subsequently could lead to inaccurate or incomplete decisions and results. However, some studies have considered uncertainties of DRERs in their developed optimization problems.

Optimal stochastic scheduling of energy hubs was presented by [76], which considered the uncertainties associated with wind turbine system output and electricity price. [115] Developed a two stage stochastic model for optimal design and operation of combined cooling, heat, and power (CCHP) unit. This study considered the uncertainties of loads and solar irradiation at different

seasons. However, energy storage system (ESS) was not considered in their study. A Stochastic operation and scheduling of energy hub considering the uncertainties of DRERs and different configurations of energy hub due to outage of sub-systems was developed by [116].

The current research studies the following aspects of energy hub modelling: 1. Optimal design, 2. Optimal operation; 2. Greenhouse gas (GHG) emission and mitigation; 4. DRERs; 5. ESS; 6. Uncertainty of renewable energy; 7. Demand size reduction.

Current literature reports have some similar individual features, but none of the literature has combined all the features of the analysis into one model. A knowledge gap exists in both:

1. Developing a stochastic optimization which comprehensively considers the design and operation planning, energy storage system and uncertainties of DRERs;
2. Applying efficient size reduction approach to large size multiple attributes demand data which can be used as an input to the stochastic energy hub model. Table 4.1 reports an overview of the literature review that has been revealed.

The first aim of the present work is to overcome the problem associated with integrating different scales of energy hub model by adopting generic clustering approach. The goal of the clustering approach is to represent the days in a year that exhibit a similar trajectory by a reduced-size typical day candidate (i.e., representative) of the operating year. In other words, the goal of clustering is to represent the yearly multiple attribute demand curves with a range of sufficiently representative curves. A sufficient number of representative curves mean the representatives are able to provide a close enough solution to the full size (high in accuracy) model while also maintaining solution tractability. Figure 4.1 shows a conceptual schematic of the clustering approach application to the multiscale decision-making problem. The development of the clustering algorithm in this chapter is based on the work done by [97] for single attribute and [117] for multiple attributes. Therefore, the multiple attribute clustering algorithm used in this study is formulated based on mathematical programming approach as a mixed integer programming problem. Furthermore, due to the computational difficulty of the mathematical programming-based clustering approach, a heuristic size-reduction approach that is derived from the general clustering approach is applied. It is worth noting that the clustering approach to reduce the size of multiscale decision-making problem is general and can be implemented to different planning optimization problem where the size of the multiscale input parameter is large and takes high computational effort to solve.

The second goal of this work is to develop two stage stochastic optimization model for the design and operation of energy hub system with hydrogen storage. Hydrogen storage system is selected due to its flexibility in offering different energy recovery pathways. For instant, hydrogen can be used to produce electricity through fuel-cell, supplied to hydrogen demand for hydrogen vehicle, injected and distributed into the existing natural gas infrastructure. Two case studies are considered to optimally design and operate the energy hub model, one is without restriction on green-house gas (GHG) emission, and another restricts the GHG emissions. A Weibull distribution statistical method is implemented to generate stochastic wind speed scenarios from wind speed data. To test the clustering efficiency, the cluster results are applied to the developed energy hub model and compared with the energy hub when the whole set of data is used. At the end of this chapter, the efficiency of the stochastic approach is assessed.

The rest of this chapter is organized as follows: Section 4.2 describes the methods used for the development and verification of the clustering algorithms and modelling the uncertainty of wind speed. Section 4.3 presents the stochastic formulation of energy hub system. In section 4.4, results of solving the stochastic energy hub model using the clustered electricity and heat demands and the full-size demand data are discussed. Additionally, the solution quality of using clustered demand and applying stochastic approach of the energy hub system are evaluated. Section 4.5 presents concluding remarks.

Table 4.1 Literature review summary on energy hubs optimization problems

study	year	Research aspects						
		Optimal design	Optimal operation	(GHG) emission saving	DRERs	ESSs	Uncertainty of renewable energy	Demand size reduction
[71]	2016	✗	✓	✗	✗	✓	✗	✗
[72]	2017	✗	✓	✓	✓	✓	✗	✗
[73]	2018	✗	✓	✓	✓	✓	✗	✗
[74]	2017	✗	✓	✓	✓	✓	✗	✗
[75]	2020	✗	✓	✗	✓	✓	✗	✗
[112]	2014	✓	✓	✗	✗	✓	✗	✗
[113]	2016	✗	✓	✓	✓	✓	✗	✗
[101]	2016	✓	✓	✓	✓	✓	✗	✗
[114]	2017	✓	✓	✓	✓	✓	✗	✗

[99]	2017	✓	✓	✗	✗	✓	✗	✗
[105]	2018	✓	✓	✗	✓	✓	✗	✓
[76]	2017	✗	✓	✗	✓	✓	✓	✗
[115]	2019	✓	✓	✓	✓	✗	✓	✗
[116]	2021	✗	✓	✗	✓	✓	✓	✗
Current work	2021	✓	✓	✓	✓	✓	✓	✓

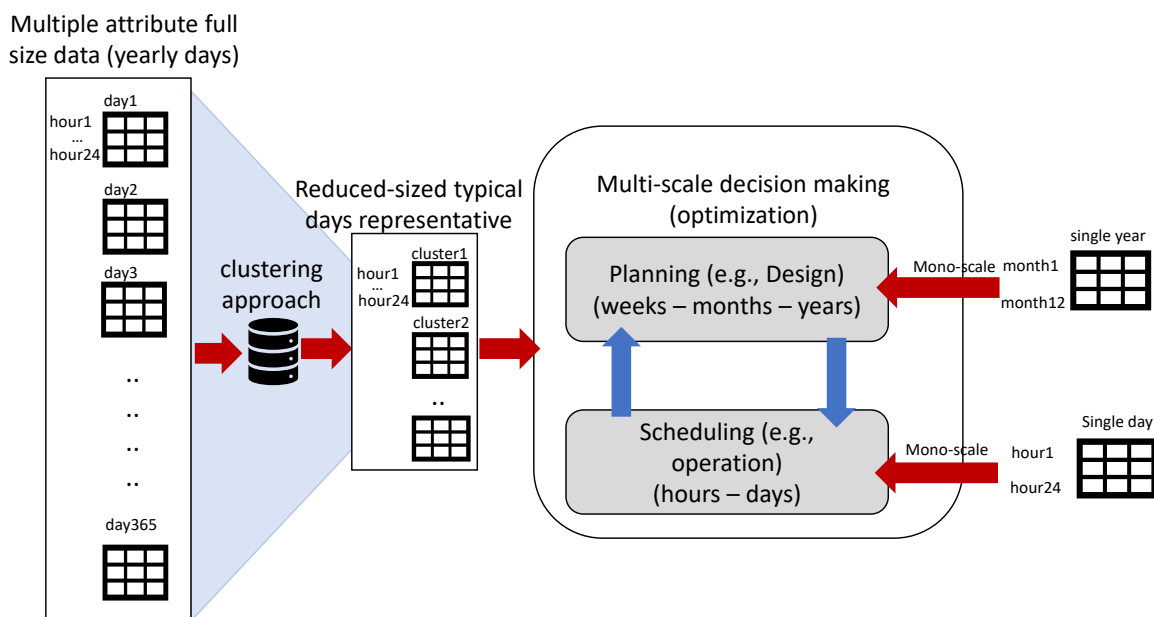


Figure 4.1. Application of the proposed clustering approach to the multiscale decision-making problem

## 4.2 Methodology: Clustering Algorithm and Stochastic Scenario Generation

Clustering has been constantly drawing attention because of their prospective applications in big data processing including time series data. The clustering approach used in this study is part of time series data, which can cluster multiple attribute data while simultaneously considering their shape-similarities and time-trajectories [118]. Therefore, applying the clustered data (reduced size time-series data) to the multiscale modelling problem can efficiently reduce the computational tractability. The  $L_1$ -norm [119]–[123] (least absolute value method) was used as the clustering measure similarity (objective function) to maintain linearity of the clustering approach model. A multiple attribute clustering algorithm is applied, as input parameters are generally consisted of

multiple attributes like the simultaneous electricity, heat and cooling demands. To tackle the issue of multiple attributes, the weighting method is selected to perform multi-objective optimization [52]. Figure 4.2 illustrates a Pareto front for a bi-objective problem. Different combinations of weight factors of both objective functions are compiled together to generate the Pareto frontier. The utopia point ( $OF^*$ ) corresponds to the optimum of both objective functions 1 and 2. However, there is typically no feasible solution at utopia point as demonstrated in the Figure 4.2. Consequently, the nearest point to feasible solution to the utopia point is considered as the best solution.

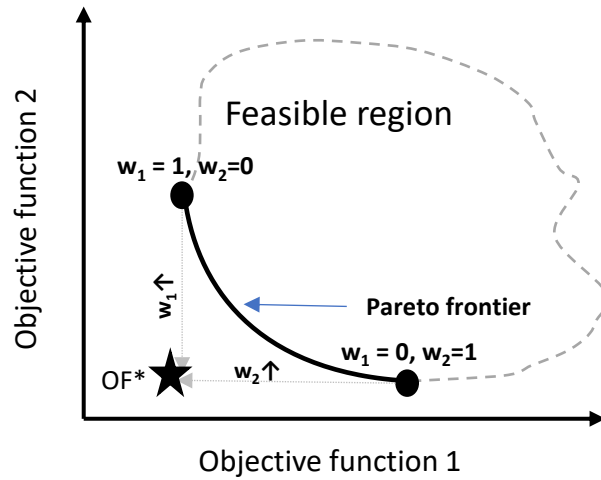


Figure 4.2. Conceptual representation for Pareto frontier

#### 4.2.1 General Clustering Algorithm Formulation

The aim of this approach is to allocate days to cluster which have minimum dissimilarity (gather days that have similar trajectory), given a set of demand (load) curves in  $D$  (days) and for  $H$  (hours) to be gathered in  $C$  clusters. Additionally, multiple attributes are included within the formulation represented by index ( $a$ ). This can be expressed in the following form:

$$\min \sum_a W_a IAE_a \quad (4.1)$$

$$\sum_{c=1}^c x_{d,c} = 1 \quad \forall d \quad (4.2)$$

Where equation (4.1) is the multi-objective performance criteria function to be minimized,  $IAE_a$  is the integral absolute error (L<sub>1</sub>-norm) that applied as similarity metrics for each attribute ( $a$ ). Equation (4.1) indicates the multi-objective performance criteria of different attributes ( $a$ ) under consideration, where each attribute is assigned to attribute  $a$ 's weighting factor ( $W_a$ , where  $W_a \geq 0$  and  $\sum_a W_a = 1$ ). Equation (4.2) represents the day assignment constraint where each day of the year should be allocated to a cluster curve  $c$ . The variable  $x_{d,c}$  is a binary variable that represents allocating loads (demands) for day  $d$  to cluster  $c$ . The IAE mathematical expression can be given as follows[118]:

$$IAE = \int_a^b |L(t) - C(t)| dt \quad (4.3)$$

where  $L(t)$  represents the load curve(s) and  $C(t)$  the clustered curve(s). Equation (4.4) is obtained by applying trapezoidal rule to equation (4.3) as follows:

$$IAE_a = \frac{\Delta}{2} * \sum_{d=1}^D \sum_{h=1}^{H-1} AD_{a,d,h} + AD_{a,d,h+1} \quad \forall a \quad (4.4)$$

where  $AD_{a,d,h}$  denotes the absolute difference between demand curve and clustered curve  $c$  for hour  $h$  in day  $d$  for attribute  $a$  which can be defined as follows (equation (4.5)):

$$AD_{a,d,h} \geq |DL_{a,d,h} - D_{a,c,h}| x_{d,c} \quad \forall a, h, d, c \quad (4.5)$$

$DL_{a,d,h}$  denotes  $a$ 's attribute demand load for hour  $h$  in day  $d$ ,  $D_{a,c,h}$  is the representative demand of attribute  $a$  for hour  $h$  hour in cluster  $c$ . It is worth noting that this model is flexible in terms of similarity measure (performance criteria), since, implementing the L<sub>2</sub>-norm as a replacement for the L<sub>1</sub>-norm can be easily done by incorporating the Euclidean distance in equation (4.3).

Furthermore, sequential clustering of demand data can be simply performed by including the following set of constraints (Equation (4.6)-Equation(4.8)) [124]. Sequence clustering can be significant to sustain flexible operations, such as on many circumstances continuous similar operations are desired to minimize the undesirable cost related to change-mode or set ups.

$$x_{d+1,1} \leq x_{d,1} \quad \forall d < D \quad (4.6)$$

$$x_{d+1,c} \leq x_{d,c} + x_{d,c-1} \quad \forall d < D, c > 1 \quad (4.7)$$

$$x_{D,c} \leq x_{D-1,c} + x_{D-1,c-1} \quad d = D, \forall c > 1 \quad (4.8)$$

Where  $D$  denote the total number of days. Equations (4.6)-(4.8) control the first, intermediate, and last clusters sequence of days, respectively.

The above-mentioned general formulation provides a general framework to perform both normal and sequence clustering because both are based on the same algorithmic structure. Nevertheless, this formulation is MINLP model because of the absolute value and multiplication between the variables  $D_{a,c,h}$  and  $x_{d,c}$  as can be seen in equation. (4.5). Simply, the absolute function can be linearized by applying linearization methods on the absolute function [125]. Furthermore, the bilinear term ( $D_{a,c,h} x_{d,c}$ ) can be further linearized by introducing a new continuous variable ( $RV_{a,h,d,c} = D_{a,c,h} * x_{d,c}$ ) called the relaxation variable through a set of constraints [126], further details on linearization approach can be found in [97]. In summary, the model for normal clustering is made up by equations (4.1) -(4.5); whereas sequence clustering is denoted by equations (4.1) - (4.5), and equations (4.6) -(4.8).

#### 4.2.2 Heuristic Approach for Multiple Attributes Data Size Reduction

Although, the abovementioned clustering approach is simple, its computational complexity is obvious as the time required to perform clustering task is very long. Therefore, the goal of this subsection is to apply heuristic size reduction algorithm to tackle this issue. By applying the heuristic approach, the previously presented MILP clustering model can be used to cluster data with long planning horizons, including multiple attributes, in computationally reasonable time. Nonetheless, the linearity and programming basis of the former clustering approach is maintained. The heuristic steps follow the k-means clustering algorithm [127] but clusters construction are derived from the previously mentioned mathematical programming-based clustering approach.

As it can be noticed from the former clustering mathematical model, it is composed of two classes of variables, namely, continuous variables ( $AD_{a,d,h}$ ,  $RV_{a,h,d,c}$  and  $D_{a,c,h}$ ) and discrete variables (the day assignment binary variable,  $x_{d,c}$ ). Accordingly, the proposed algorithm decomposes the original problem into a master problem and subproblem. the master problem is a Mixed Integer Programming (MIP) problem which solve the complicated variables (day assignment integer variable ( $x_{d,c}$ )) and fix them to given feasible integer. The subproblem is a linear programming (LP) problem that solves the resulting continuous problem (clusters curves  $D_{a,c,h}^n$ ) using those fixed integer variable values from the master problem. The upper bound of the objective would be obtained by solving the master problem (MIP), while the lower bound of the objective function



would be obtained by solving LP subproblem. In the master problem, the initial guess clusters are fed to the problem as parameter. The algorithm executes in iterative bases in which will keep iterating until the difference between upper and lower bound solution is within acceptable range. This type of heuristic approach has been utilized in the past and considered as an applicable and reliable technique to solve similar type of large-scale mathematical models [108], [128].

Figure 4.3 and Figure 4.4 show the flowchart of the proposed heuristic algorithm for multiple attributes. Figure 4.3 shows the execution of the heuristic algorithm for different weight factor combinations. Figure 4.4 depicts the execution of the heuristic algorithm using single weight factor combination. The Pareto frontier can be constructed by considering all scenarios for a given weight factor, then procedure goes to the next weight factor, and repeat these steps until all weight factors are considered. The procedure for a given attribute weight factor is given as follows:

1. **Initialization:** Set the number initial guess scenarios  $N$
2. **Generate random initial guess clusters scenarios:** The scenarios are generated using random uniform distribution between maximum and minimum demand of each hour for each attribute in the entire demand curves  $\{D_{a,c,h}^{n=1}, D_{a,c,h}^{n=2}, \dots, D_{a,c,h}^{n=N}\}$ .
3. **Initial scenario:** Consider scenario  $n = 1$ .
4. **Master problem solution:** Solve for day assignment integer variable  $(x_{d,c})$ , given fixed clusters curves in order to obtain upper bound objective function  $(Z_{UB}^n)_{iter}$
5. **Subproblem solution:** Solve for clusters curves  $(D_{a,c,h}^n)$ , given fixed day assignment integer variables  $(x_{d,c})$  from the master problem and obtain the lower bound objective function  $(Z_{LB}^n)_{iter}$
6. **Convergency check:** If  $|(Z_{UB}^n)_{iter} - (Z_{LB}^n)_{iter}| \leq Tol$  go to 7. Otherwise, feed the cluster curves  $(D_{a,c,h}^n)_{iter+1}$  obtained from solving the subproblem into the master problem and repeat step 4 to 6, keep iterating until convergence is achieved.
7. **Next scenario:** Record  $n$  scenario solution and then, consider the next scenario, and repeat step 4 to 6. If all scenarios have been considered go to the next step.
8. **Scenario with minimum objective function:** The solution of the problem (i.e., clusters) will be corresponding to the objective function with minimum value  $\min Z^n$ .

The model can be used for normal clustering equations (4.1), (4.2) and equations (4.4)-(4.5) or sequence clustering (equations (4.1)-(4.2), (4.4)-(4.5) and (4.6)-(4.8)) Generally, both types of clustering problems with multiple attributes can be performed using this formulation approach. The mathematical models were built in the General Algebraic Modelling System (GAMS) [15].

The total number of the continuous variables of the general formulation clustering approach for 2 attributes is  $(2 \times 24[D(1 + C) + C])$  and the total number of binary variables is  $(DC)$  where  $D$  and  $C$  are the total number of days and clusters, respectively. On the other hand, the total number of binary variables of the master problem of the heuristic clustering approach is  $(DC)$ , while the total number of continuous variables of the heuristic subproblem are  $(2 \times 24[D + C])$ .

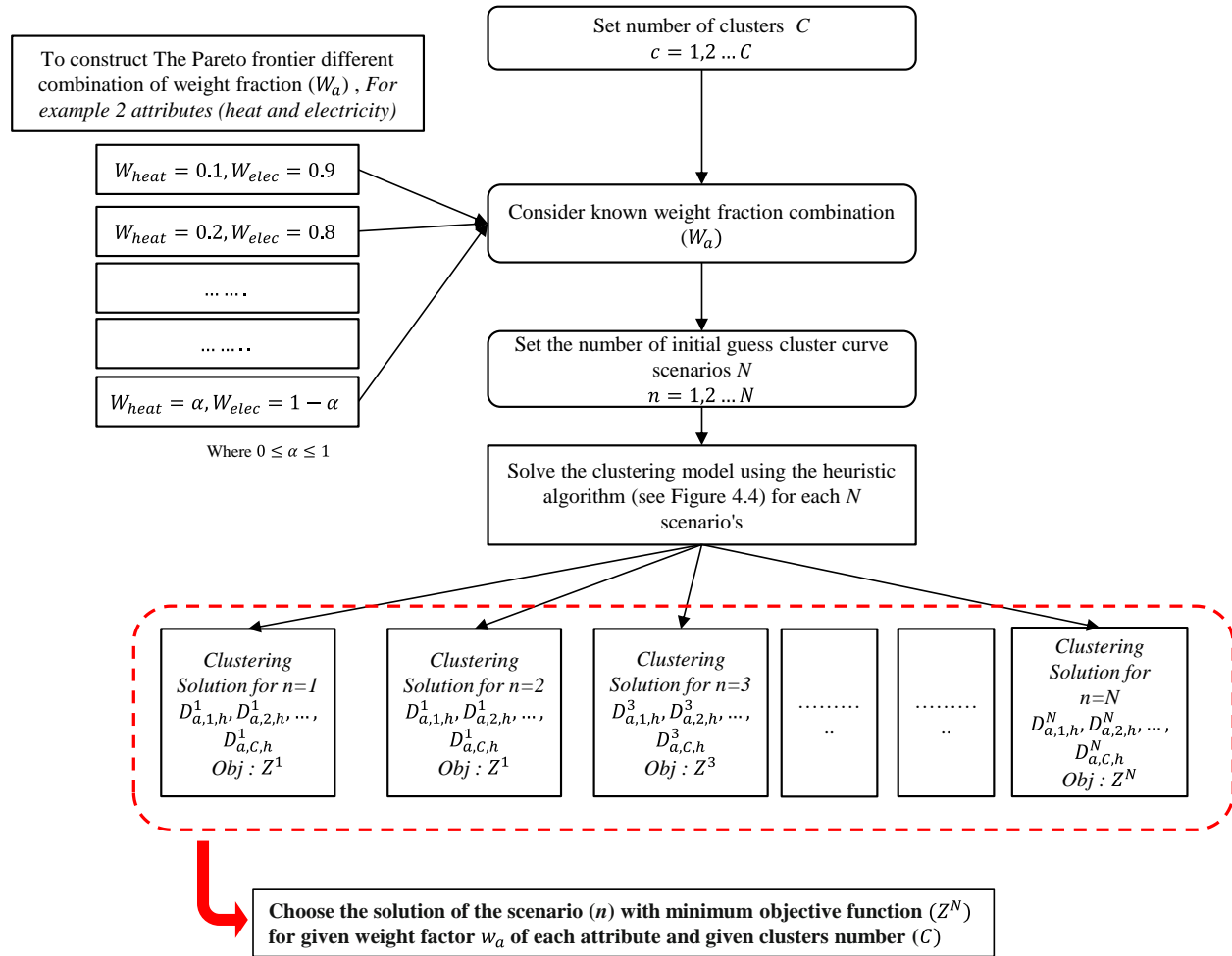


Figure 4.3. Graphical representation of the heuristic size reduction algorithm for multiple attributes.

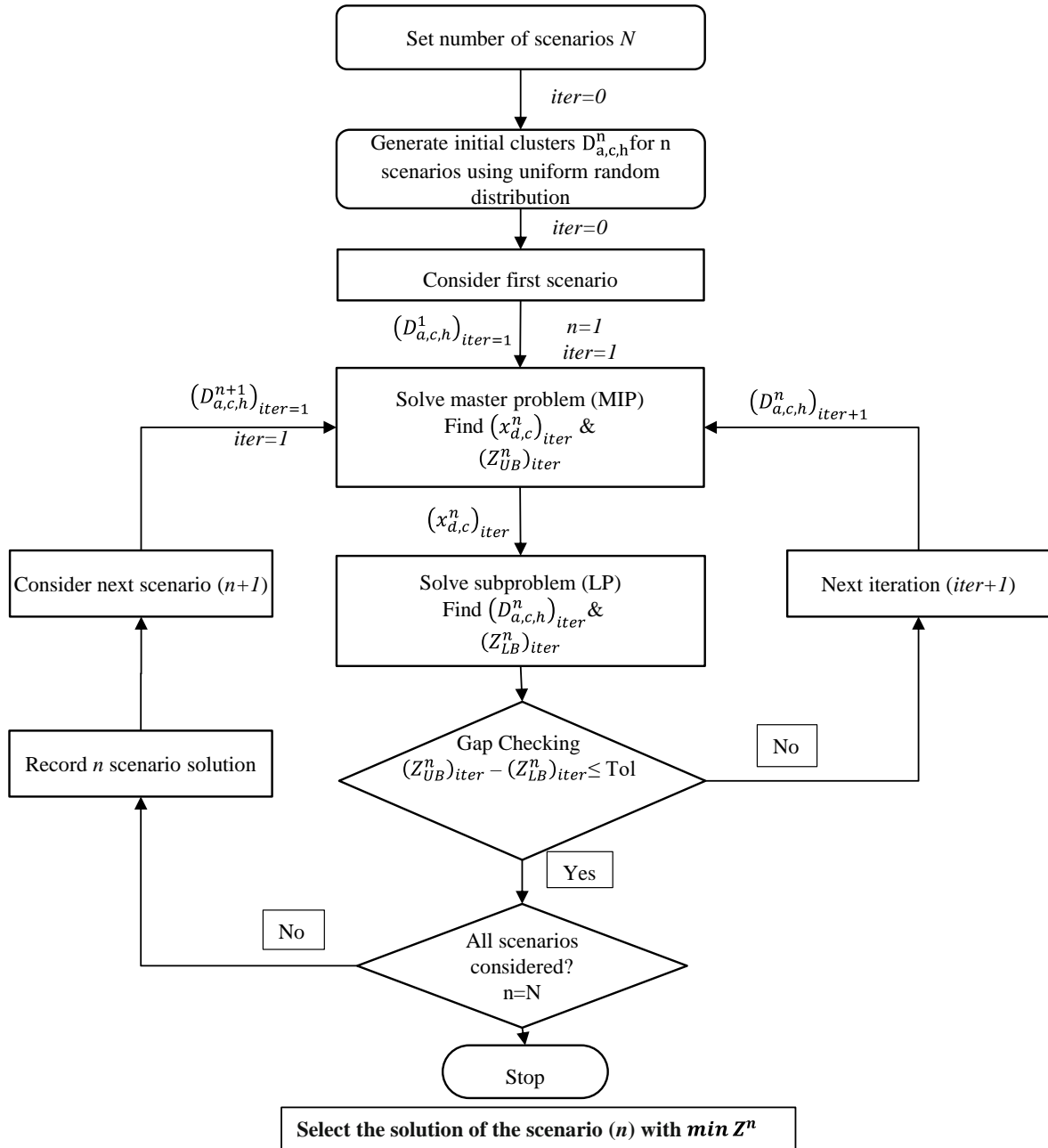


Figure 4.4. Graphical representation of the heuristic size reduction algorithm for multiple attributes for a single weight factor

### 4.2.3 Clustering Assessment

This section aims to assess the computational performance of the general formulation clustering algorithm (mathematical programming-based formulation model) and the heuristic clustering algorithm derived in the previous sections. In this case study, hourly heat and electricity demands data during a year for the energy hub system is employed for demonstration and evaluation purposes [129]. Figure B.1 and B.2 in Appendix B show the heat and electricity demands,

respectively. Weight factor combinations used to construct the Pareto frontier are listed in Table 4.2.

Table 4.2. Attributes weight factors for the multi-objective function (overall clustering similarity measure)

Weight Factor	Electricity	Heat
1	0.20	0.80
2	0.30	0.70
3	0.40	0.60
4	0.50	0.50
5	0.60	0.40
6	0.70	0.30
7	0.80	0.20
8	0.90	0.10

The solutions quality and solutions time for the mathematical programming-based (section 4.2.1) and the heuristic approach (section 4.2.2) clustering algorithm (size reduction) for multi-attributes were examined. It was noticed that the general clustering formulation cannot tackle the whole year heat and electricity demand data especially when normal clustering approach is used.

It is worth noting that, when normal clustering is applied for 1-year demand data, no solutions were returned after 48 hours using the general clustering formulation. Therefore, for the sake of comparison, the two algorithms were tested using reduced data set composed of 20 days. However, when sequence clustering of general formulation was used to cluster 1-year demand data, the solution time was reasonable. Therefore, the performance of the two algorithms was compared under sequence clustering using 1-year demand data. The runs for this comparison study (i.e., between heuristic and general clustering model) include 4, 5, and 6 clusters using a 20-day demand data for normal and 365 for sequence clustering (i.e., total 6 runs). Moreover, the weight factor was set to be 0.5 for both attributes in all runs. 30 initial guess scenarios were generated for each of heuristic formulation run. Table 4.3 shows, optimal objective function value and solution time using both clustering formulations.

Table 4.3 Computational performance of heuristic and general formulation clustering approaches

		<b>Objective function (MWh)</b>		<b>Solution time (min)</b>	
		heuristic formulation	general formulation	heuristic formulation	general formulation
<b>Normal clustering - 20 days</b>	4	10.836	10.836	1.05	50.25
	5	9.192	9.192	1.23	148.03
	6	8.356	8.34	1.38	434.75
<b>Sequence clustering - 365 days</b>	4	469.144	469.144	149.58	469.86
	5	430.696	430.969	500.42	2142.48
	6	404.9	404.54	770.42	11703.16

As it can be noticed from this table that the heuristic approach needs much less time to solve the clustering problem than the general formulation method model (mathematical programming-based). Although at larger number of clusters, the heuristic approach objective function differs from the general mathematical programming-based clustering approach. However, the difference in IAE values is minor. Therefore, one can say with certainty that the heuristic approach outperforms in terms of solution time especially when the dataset is large with close proximity to the solution form the general formulation model. Consequentially, the heuristic approach was implemented to generate clusters curves for our case study of energy-hub model presented in the next section.

Furthermore, in this section, the outputs of the multiple attributes using the proposed heuristic clustering algorithm (section (4.2.2)) were assessed. The heuristic clustering algorithm was used to cluster an entire year (365 days) demand data into 4, 5 and 6 clusters using normal and sequence clustering approaches. The weight factor combinations (Table 4.2) were considered to generate the Pareto frontier of each cluster run. Moreover, 30 scenarios were generated per run. The GAMS/CPLEX [15] solver was used to perform the runs on an Intel(R) core i7 (R) 4.0 GHz, 16 GB RAM personal laptop. The algorithm tolerance was set to  $10^{-3}$ . The average solution time for these runs is reported in Table 4.4. It can be noticed this table that the solution time for sequence clustering is slightly shorter than normal clustering due to the additional constraint sets included in sequence clustering which shrink the feasible region size, resulting in less solution time. It is clear from table that, increasing the number of clusters results in bigger model size and thus longer solution time. In general, the model is challenging to solve even with a small number of binary variables.

Pareto frontiers for normal and sequence clustering are illustrated in Figure 4.5. The Pareto frontiers considered all weight factor combinations shown in Table 4.2 for all runs. As depicted in the figure, increasing the number of clusters has a positive effect on the value of objective function (IAE) for both: normal and sequence clustering.

Table 4.4. Solution time of heuristic clustering approach under different runs

Average solution time per scenario (min)	Normal clustering			Sequence clustering		
	4	5	6	4	5	6
	7.03	18.3	28.6	5.98	16.68	25.6

A relative error function is used as evaluation measure between the cluster curves and the load (demand) curves to attain insightful conclusions as follows:

$$RE_{h,d,c} = \frac{D_{h,c} - DL_{d,h}}{DL_{d,h}} \quad (4.9)$$

where  $RE_{h,d,c}$  is the relative error between the cluster and load curves. Table 4.5 displays the average error using weight factor equal to 0.5 for all cluster runs. This metric basically represents the integral absolute error (IAE) scaled by the cluster curve, to evaluate performance independently from the scale of the data set and allow comparisons when the demand curves are significantly differed in magnitude. This error measurement is broadly applied in utility forecasting studies[130]. In order to measure the spread of error between cluster curve and loads belong to this cluster, the error standard deviation was also calculated. Average results of relative error and standard deviation using weight factors of 0.5 are presented in the same table.

Results in Table 4.5 present that normal clustering performs better than sequence clustering in terms of objective function value, relative error average, and standard deviation. This is due to the extra sequence restriction (constraints) that might be a requirement in certain process decision making. Furthermore, as it can be noticed in Figure B.1 (annual heat demand) that heat demand undergoes significant fluctuation and reach zero values or very close to zero in certain periods. Moreover, the average error and standard deviation values associated with heat demand are relatively bigger compared to the electricity, due to the high fluctuation in heat demand because of season change (low heat demand in summer months between May and July).

Table 4.5. Summary of relative error statistics for normal and sequence clustering using weight factor 0.5 (365 days-4,5 and 6 clusters)

Clusters		Electricity			Heat		
		Average relative error	Standard deviation of relative error	IAE (MWh)	Average relative error	Standard deviation of relative error	IAE (MWh)
Normal	4	0.073	0.076	200.2	7.667	36.510	460.3
	5	0.070	0.077	191.0	6.155	2.475	402.5
	6	0.059	0.063	175.0	1.469	7.372	370.1428
Sequence	4	0.084	0.094	271.7	11.414	76.977	661
	5	0.080	0.087	241.8	6.601	29.838	627.5
	6	0.072	0.084	231.8	5.149	23.637	574.5

Figure 4.6 shows the actual heat and electricity demand data used in this case study and the corresponding representative cluster curves of the data using both normal and sequence clustering approach (4, 5 and 6 normal and sequence clusters). The weight factor used to generate the corresponding representative cluster curves is 0.5 for both heat and electricity attributes. As it can be seen in this figures, clustered results are in good agreement with the actual demand data. However, for heat demand data we can say that cluster curves have slightly larger discrepancy, but mostly follows the tendency of the actual demand, due to the high fluctuation in the actual heat demand. Also, it can be concluded from this figure that normal clustering curves match better with actual demand data than the sequence clustering curves. Despite the slight error associated with the cluster curves, the purpose of performing clustering is to use a reduced size set of demand data that is well representative and can reflect the underlying trends. Moving forward, these cluster curves reduced-size demand data will be used as an input for planning, designing, and operating the energy hub model and will serve to improve the tractability of the solution.

Figures B.5 to B.10 in Appendix B show the clusters and day assignments of normal and sequence clustering for weight factors 1 and 8 along with 4, 5, and 6 clusters. The weight factor 1 prioritizes heat demand, whereas the priority for weight factor 8 is electricity demand. It is clear from those figures that cluster curves are slightly affected by the weight factors. The advantageous of using the weight factor approach in the current clustering algorithm is allowing to perform clustering with the emphasizes on one/or more attributes than the other. It was also noticeable that many clusters of electricity demand, especially the sequence clusters, are the same yet correspond to different days. This is because they represent different representative days (clusters), in which the

clusters of heat demands for those days are different, and therefore, they cannot be merged into the same cluster. Therefore, due to the advantage of normal clustering over sequence, the use of normal clustering to reduce computational effort and handle large scale models is recommended when the applications do not require sequencing.

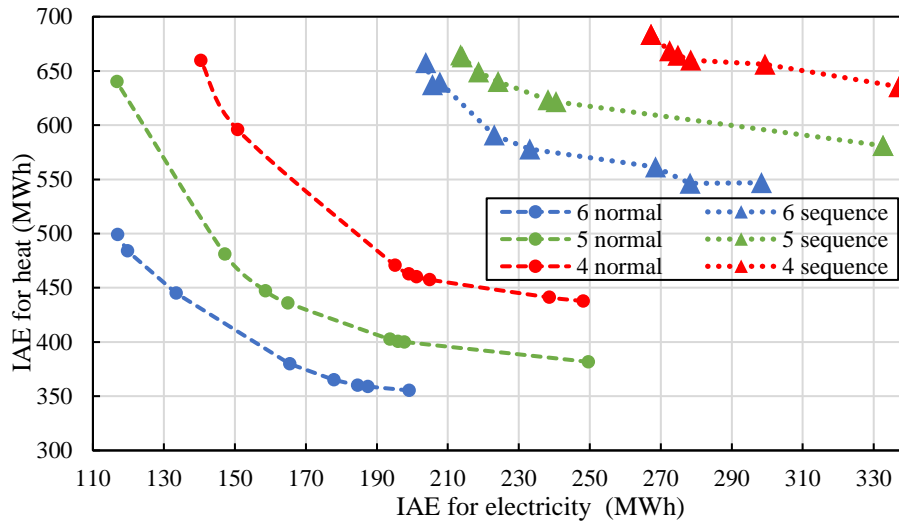


Figure 4.5. Pareto frontiers for normal and sequence clustering using different number of clusters

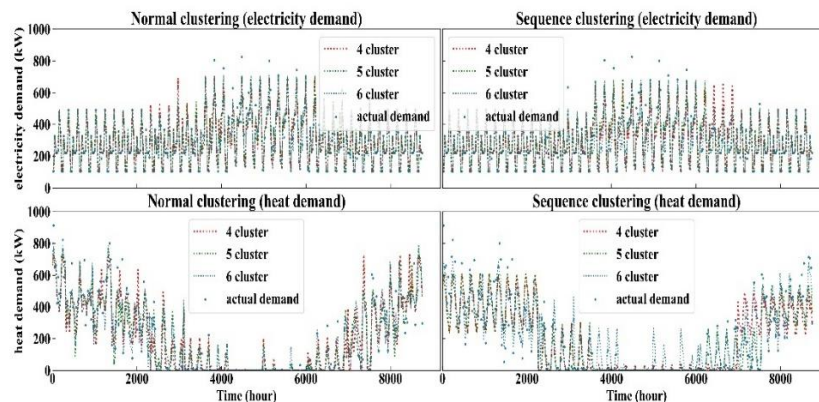


Figure 4.6. Actual electricity (top) and heat demand (bottom) and its computed cluster curves (4,5 and 6 clusters) using normal (left) and sequence clustering approach (right) for 1-year time horizon

#### 4.2.4 Uncertainty Modelling of Wind speed

Unfortunately, the wind is notoriously fickle, varying substantially throughout a day, from season to season and even from year to year. Weibull distribution is favourable to describe the fluctuation in wind during anytime interval using two parameters [131]. It is a method that has been widely



used both in industrial and academic studies. This statistical tool reflects how often winds of different speeds will be seen at a certain location. Therefore, in this study wind speed data were collected for one year and then fitted into Weibull distribution using equation (4.10) [131]. The collected wind data reflected the measured wind speeds in 2018 from the Waterloo region, courtesy of The National Solar Radiation Data Base (NSRDB) [132]. The maximum likelihood method (MLM) was used to fit Weibull distribution on the measured wind speed data [30]. By using MLM, the best-fit Weibull distribution for the available data was obtained and presented in Figure 4.7. Also, the graph shows the probability which the variable wind speed falls within different bins.

$$prob(v) = \frac{k}{c} \left(\frac{v}{c}\right)^{k-1} \exp\left[-\left(\frac{v}{c}\right)^k\right] \quad (4.10)$$

Where  $v$  is the wind speed,  $k$  and  $c$  denote the shape and scale parameter of Weibull distribution, respectively. From Weibull distribution, the probability of wind speed occurrence can be estimated. By doing this, stochastic scenarios can be generated, each scenario has a probability as shown in Figure 4.8. The probability at which the wind speed is between two limits is given by equation (4.11):

$$prob_s(v_s^u > v > v_s^l) = \int_{v_s^l}^{v_s^u} \frac{k}{c} \left(\frac{v}{c}\right)^{k-1} \exp\left[-\left(\frac{v}{c}\right)^k\right] dv = \Phi(v_s^u) - \Phi(v_s^l) \quad (4.11)$$

Where  $\Phi$  is the cumulative distribution function,  $v_s^u$  and  $v_s^l$  are the upper and lower wind speed limit of each stochastic scenario ( $s$ ), respectively. Accordingly, the inverse of cumulative Weibull distribution ( $\Phi^{-1}$ ) returns the wind speed at given probability of occurrence. Therefore, the upper and lower wind speed limit of each stochastic scenario can be calculated as follows:

$$\Phi^{-1}(prob[v_s \geq v]) = v_s \quad (4.12)$$

In order to obtain scenarios with equal probabilities (equal areas under the probability density function curve), each scenario is represented by probability that is equal to  $\left(\frac{1}{S}\right)$ , where  $S$  is the total number of scenarios. The upper and lower limits for wind speed of each scenario can be calculated as follows (equations (4.13) and (4.14)):

$$v_s^u = \Phi^{-1}\left(\frac{S}{S}\right), v_s^l = \Phi^{-1}\left(\frac{S-1}{S}\right), \quad \forall s < S \quad (4.13)$$

$$v_s^u = \Phi^{-1}(0.99), v_s^l = \Phi^{-1}\left(\frac{s-1}{S}\right), \quad s = S \quad (4.14)$$

To avoid infinities, equation (4.14) was used to impose 99% confidence interval because when  $s = S$ ,  $\Phi^{-1}(1) = \infty$ .

In the following case study, 10 scenarios were generated from the fitted Weibull distribution curve as shown in Figure 4.8. As it can be seen from this figure, each shaded area under the probability distribution function curve represents a single stochastic scenario with probability of (1/10).

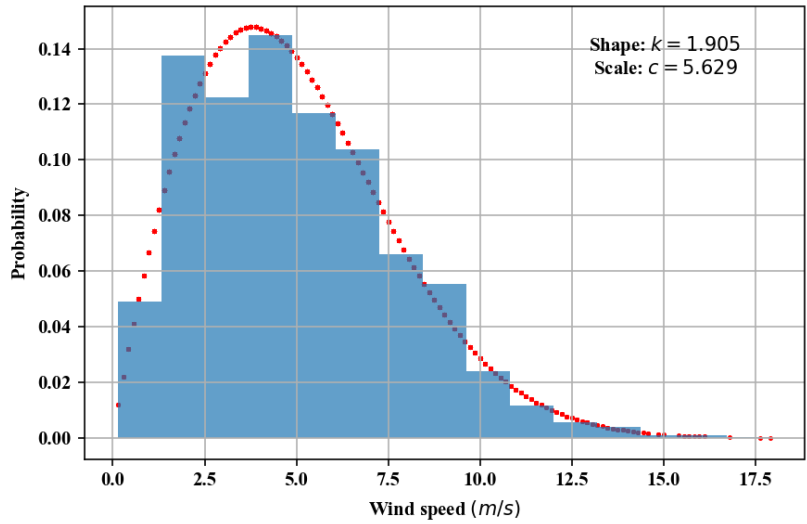


Figure 4.7. Actual and best fit distribution wind speed profile

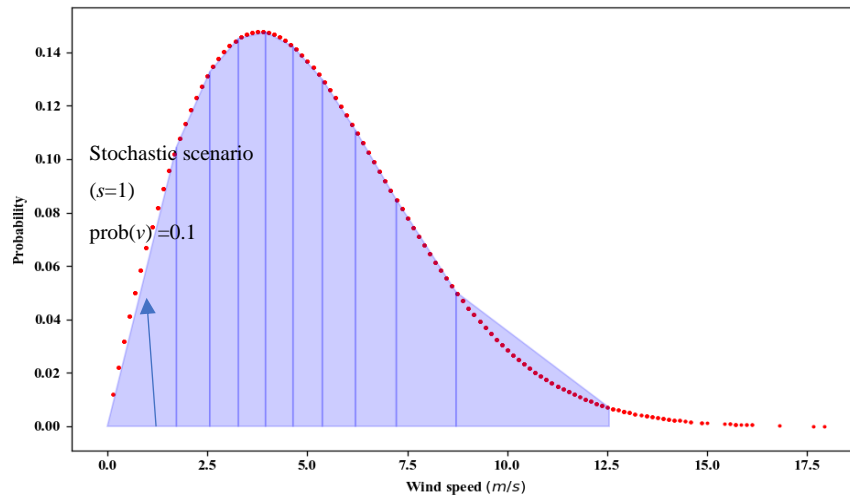


Figure 4.8. Wind speed stochastic scenarios

### **4.3 Application of the Multiple Attribute Clustering Approach to Energy Hub**

As mentioned in the introduction section, this study showcases an application of, 1. clustering algorithms to multiple attributes demand data, and 2. data-driven statistical method to represent the intermittent behaviour of uncertain wind speed data 3. reformulating the design and operation of energy hub with hydrogen storage as multiscale model with multiple attributes by agglomerating demand data with similar profiles and generating stochastic scenarios for a two stage stochastic model based on uncertain wind data. Moreover, in this section, the impact of clustering on the accuracy of the optimal energy hub decisions using clustered and full-size demand data will be investigated. In other words, the solution of the energy hub planning model under uncertain wind with multiple demand attributes using the outputs of the normal and sequence clustering algorithms against the full-size demand data (without clustering), will be evaluated. The design and operation of energy hub problem is strategic (long-term) and medium-term decision level, aimed to minimize the total annual cost of designing (installing and sizing) and operating energy hub units while meeting demands. There are several models for the energy hub problem available in the literature, ranging from heuristics to mathematical programming. The energy hub problem adopted in this section is modelled as MILP [129] model. The following subsection presents the energy hub model formulation.

#### **4.3.1 Energy Hub Model Formulation**

This subsection discusses the stochastic modelling for the design and operation of energy hub system under uncertain wind speeds utilizing clustered demand data (heat and electricity).

The present energy hub system model aims to minimize the annual operational and maintenance cost, as well as the capital cost while meeting electricity and heat demands within the units' operating capacities and physical constraints.

In this case study, both DRERs and conventional DERs (Distributed Energy Resources) based on fossil fuel are considered. The current energy hub system includes variety of conventional energy conversion technologies powered by natural gas such as combined heat and power (CHP) units and boilers, and a non-conventional energy conversion technology (i.e., wind turbines) powered by renewable energy resources. Additionally, it utilizes a hydrogen production and storage system, from electricity utilizing electrolyzer, hydrogen tank and fuel cell as ESS (energy storage system).

Hydrogen storage system is chosen, because it can play a role in both storing energy and supply hydrogen demand for hydrogen vehicles. Figure 4.9 shows the energy hub layout with considered energy technologies and input data handling (wind speed, electricity and heat demand).

The optimization program will decide the number of each unit and the respective capacity within the energy hub system, as well as the operating points for the electrolyzers, hydrogen tanks, fuel cells, boilers, and CHP generators at each time point. Particularly, in this chapter, the discrete size of each technology is considered in the optimization, which makes this work more realistic. The number and type of each technology chosen is a design decision variable while the operating variables are related to how energy hub units are running. The main outputs of the optimization model can be summarized as follows:

1. Type and number of energy conversion and storage technologies within the hub.
2. Design and the operation of optimal energy hub under intermittent wind energy availability, and based on multiple attributes aggregated demand or full-size demand data.
3. Economic cost of the system including capital, operation and maintenance and fuel consumption.
4. Environmental impact of the system through measuring the GHG emissions (mainly CO<sub>2</sub>).

Natural gas is fuel for both: the boiler and CHP. As illustrated, the electricity demand is met by means of the CHP generators, wind turbines and fuel cell; whereas heat is met by the boilers and CHPs. The list of the energy generation technologies and its technical and economical properties are given in Table 4.6. This model is a general framework for microgrid/energy hub system where different technologies can easily be added or removed, according to the problem that needs to be solved.

The mathematical model was formulated as two stage stochastic with recourse, where the first stage decisions decide the design of the system which includes the number of each unit and the respective capacity within the energy hub. While the second stage decisions schedule the operation of the system including the operating points for the electrolyzer, fuel cell, CHP units and boilers at each time point. The two stage stochastic recourse (we refer to it as recourse problem (RP)) formulation is basically a bi-level optimization formulation whose inner optimization problem mimic the second stage planning process. Due to special structure, two stage stochastic programs can be naturally reformulated into an equivalent single level optimization problem. Therefore, the

single level optimization formulation of RP for design and operation of energy hub system can be directly written as follows:

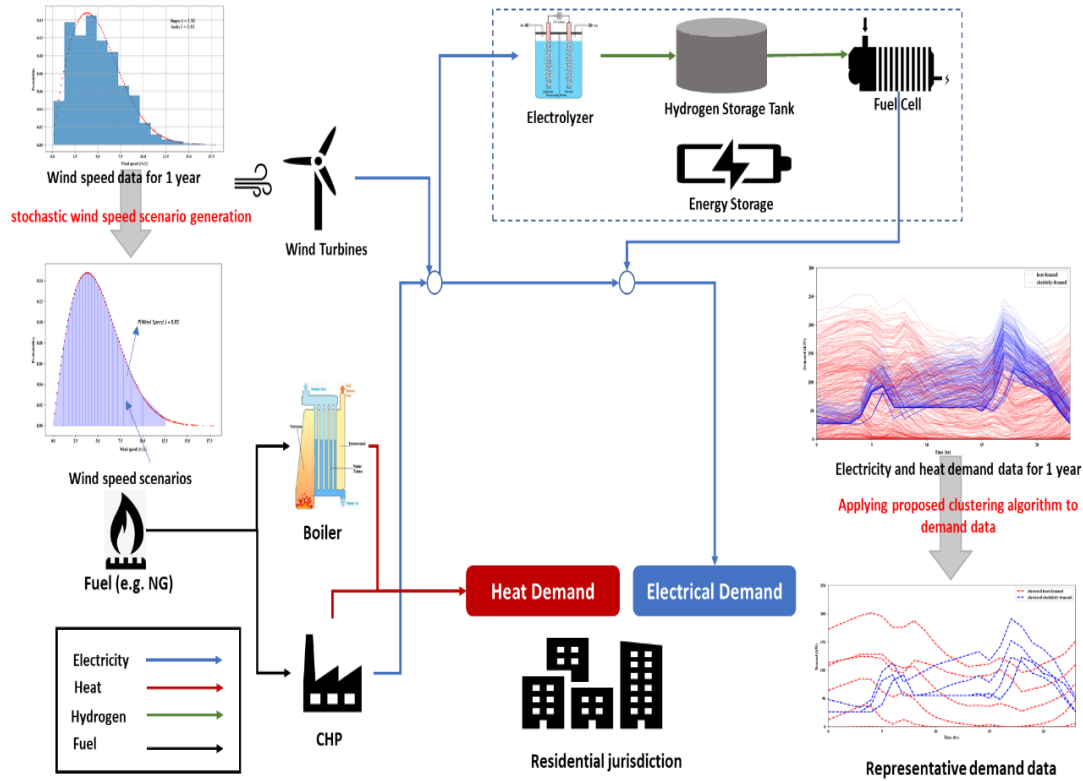


Figure 4.9. Understudy energy hub architecture

Table 4.6. Technical and economic information of energy conversion and storing technologies

unit	rated capacity (kW)	Input energy form	Output energy form	efficiency	Capital cost	Operating & maintenance cost (\$/kW)
Boiler [101]	530			0.82	100 (\$/kW)	0.027
	300	kW fuel HHV	kW heat	0.9	120 (\$/kW)	0.027
	100			0.8	150 (\$/kW)	0.027
CHP [101]	300	kW fuel HHV	kW power	0.26	900 (\$/kW)	0.016
			kW heat	0.44		
	100	kW fuel HHV	kW power	0.35	1080 (\$/kW)	0.016
			kW heat	0.5		
	60	kW fuel HHV	kW power	0.31	1200 (\$/kW)	0.0111
		kW heat	0.56			

Wind Turbine [133]	20	kW available by air	kW power	0.4	2200 (\$/kW)	0.008
	30	kW available by air	kW power	0.42	1906 (\$/kW)	0.008
<hr/>						
Storing units						
Electrolyzer [101]	290	kW power	kg of H <sub>2</sub>	0.0193	155051\$ / unit	0.06
Fuel Cell [134]	250	kg/hr of H <sub>2</sub>	kW power	16.5	210630\$/unit	0.06
<hr/>						

The objective function in equation (4.15) represents the total annual cost including capital cost of the energy hub units and their operating cost including fuel consumption, operation and maintenance costs. The first part of the equation represents the capital cost of the energy hub units (i.e., first stage decision of the stochastic programming). The second part of the equation denotes the annual net cost from operating the energy hub, which is basically operational and maintenance and fuel consumption, that depends on the scenario of wind speed uncertainty realization  $s$  with probability  $\beta_s$ .

$$\begin{aligned}
& \min \text{CRF} \left[ \sum_u y_u \text{CAP}_u E_{max}^u + \sum_{wt} y_{wt} \text{CAP}_{wt} \sum_{st} y_{st} \text{CAP}_{st} \right] \\
& + \sum_s \beta_s \left[ \sum_{h,d} \left( \sum_u [P_{i,d,h,s}^u \text{OM}_u + \text{NG}_{d,h,s}^u \text{Price}_{ng}] + \sum_{st} P_{i,d,h,s}^{st} \text{OM}_{st} \right) \right]
\end{aligned} \tag{4.15}$$

Where  $u$ ,  $st$  and  $wt$  are sets of fossil fuel based power and heat generation units, storing units, and wind turbines units, respectively.  $y$  denotes the integer design variable that represents the number of each unit needed to be installed.  $P_{i,d,h,s}$  is the operational decision variable that represents the amount of energy flow ( $i$  denote the type of energy heat or electricity) consumed or produced by each energy hub unit at  $s$  scenario and  $h$  hour of the  $d$  day.  $\text{CAP}$  and  $\text{OM}$  represent capital, and operational and maintenance cost parameters, respectively.  $\text{Price}_{ng}$  symbolizes the natural gas price (0.325 \$/m<sup>3</sup>) [129].

The capital cost of each unit of the energy hub is obtained by summing the number of installed unit  $y$  multiplied by their unit capital cost  $\text{CAP}$  (\$/unit) (in the case of power and heat generation units (e.g., CHP and boilers) the capital cost is defined as (\$/kW<sub>installed</sub>), so it is additionally multiplied by its rated capacities  $z_{\text{rated}}^u$ ) and converting the present value of the capital cost to

annuity (\$/yr) by means of the capital recovery factor (CRF). Capital recover factor (CRF) is calculated using  $CRF = \frac{r(r+1)^{life}}{(r+1)^{life+1}}$ . Where  $r$  (8%) and  $life$  (25 years) denote the interest rate and the life time of the energy hub, respectively.

The electricity and heat demands are satisfied at any  $s$  scenario and  $h$  hour of day  $d$  through the following energy balance equations using equations (4.16) and (4.17). Electricity output is fixed to meet demand while heat output is allowed to exceed demand if necessary due to excess heat from CHP units.

$$\sum_u P_{elec,d,h,s}^u + \sum_{wt} P_{elec,s}^{wt} - P_{elec,d,h,s}^{Elyzr} + P_{elec,d,h,s}^{Fuelcell} = L_{elec,d,h} \quad \forall h, d, s \quad (4.16)$$

$$\sum_u P_{heat,d,h,s}^u \geq L_{heat,d,h} \quad \forall h, d, s \quad (4.17)$$

where  $L_{elec,d,h}$  (kW) and  $L_{heat,d,h}$  (kW) are the hourly electricity and heat demands, respectively. The optimization problem is further constrained by various physical requirements. Each energy hub unit takes in a certain type of energy or mass flow and outputs a different kind of energy or mass flow. Thermodynamic efficiencies are used in the following set of equations (4.18)-(4.20) to calculate the converted utilities produced by energy hub units such as storing units (i.e., electrolyzer and fuel cell) power units (i.e., CHP) and heat generation units (i.e., boilers). The efficiency of the system depends on the condition and operating regime of the unit, however, for simplicity, efficiencies are assumed to be constant for all operating conditions in this study.

$$P_{i,d,h,s}^u = NG_{d,h,s}^u \eta_i^u b \quad \forall h, d, s, u = \{CHP1, CHP2, CHP3, boiler1, boiler2, boiler3\}, i = \{elec, heat\} \quad (4.18)$$

Where  $b$  represents the conversion factor for the natural gas flowrate (10.7 kWh/m<sup>3</sup>).

$$H_{d,h,s}^{Elyzr} = P_{elec,d,h,s}^{Elyzr} \eta_{H_2}^{Elyzr} \quad \forall h, d, s \quad (4.19)$$

$$P_{elec,d,h,s}^{Fuelcell} = H_{d,h,s}^{Tank} \eta_{elec}^{Fuelcell} \quad \forall h, d, s \quad (4.20)$$

Where  $H_{d,h,s}^{Elyzr}$  and  $H_{d,h,s}^{Tank}$  is the mass flow rate of hydrogen gas produced by electrolyzer and leaving the hydrogen tank, respectively, in (kg/hr) at  $s$  scenario and  $h$  hour of the  $d$  day. The wind turbine, however, is not modelled using previous equations. The power delivered by wind turbine to the electricity grid can be calculated using the following equation [131]:

$$\mathbb{P}_s^{wt} = \begin{cases} 0 & , v_s < v_{cut\_in}^{wt} \\ C_p \frac{1}{2} \rho_{air} (v_s)^3 A \eta_{wt} & , v_{rated}^{wt} > v_s \geq v_{cut\_in}^{wt} \\ \mathbb{P}_{s,rated}^{wt} & , v_{cut\_out}^{wt} > v_s \geq v_{rated}^{wt} \\ 0 & , v_s \geq v_{cut\_out}^{wt} \end{cases} \quad (4.21)$$

Where  $wt$  is a the represents the wind turbines types considered in this case study, two wind turbines type where considered namely Vergent (20 kW) and Fuhrlander (30 kW), the characteristic of these wind turbines can be found in [133] (Figure B.11 in Appendix B).  $\mathbb{P}_s^{wt}$  is a parameter denotes the electrical power generated by one wind turbine in (kW) of type  $wt$  at scenario  $s$ .  $v_s$  is the actual wind speed in (m/s) at scenario  $s$ . The wind speed scenarios as well as its corresponding probabilities from section (4.2.4) are used to calculate the power produced by single wind turbine  $\mathbb{P}_s^{wt}$ .  $v_{cut\_in}^{wt}$  is wind turbine specific characteristic representing the cut-in-speed, the minimum wind speed at which the turbine blades overcome friction and begin to rotate. Rated output wind speed ( $v_{rated}^{wt}$ ), for this speed and above, the wind generator is limited to its maximum design output power cut-out-speed ( $v_{cut\_out}^{wt}$ ) it is a wind speed where braking system is employed to bring the rotor to a standstill to prevent the wind turbine from damage.  $\eta_{wt}$  is the wind generator efficiency. The rotor swept area and the air density are represented by  $A^{wt}$  and  $\rho_{air}$  respectively.  $C_p$  describes the fraction of the power in the wind that may be converted by the turbine into mechanical work. The maximum achievable value of  $C_p$  is 16/27. The factor 16/27 is known as the Betz limit or Betz efficiency, The Betz limit applies to any type of wind-driven machine [131].

Furthermore, Equations (4.22)-(4.25) determine the number of units that need to be installed (designed) in order to satisfy demand. Also, they ensure that operation of any energy hub unit at any time are within their corresponding capacities as follows:

For boilers and CHP units:

$$P_{i,d,h,s}^u \leq y_u z_{rated}^u \quad \forall h, d, s, u = \{CHP1, CHP2, CHP3, boiler1, boiler2, boiler3\}, i = \{elec, heat\} \quad (4.22)$$

For electrolyzer (Elyzr):

$$P_{elec,d,h,s}^{Elyzr} \leq y_{Elyzr} z_{rated}^{Elyzr} \quad \forall h, d, s \quad (4.23)$$

For fuel cell:



$$P_{elec,d,h,s}^{Fuelcell} \leq y_{Fuelcell} z_{rated}^{Fuelcell} \quad \forall h, d, s \quad (4.24)$$

For hydrogen tank:

$$HL_{h,d} \leq y_{Tank} z_{rated}^{Tank} \quad \forall h, d, s \quad (4.25)$$

Where  $z_{rated}$  is a parameter represent the rating capacity of each energy hub unit (Table 4.6).  $HL_{h,d,s}$  is the amount of hydrogen stored in hydrogen tank in (kg) at the  $h$  hour of the  $d$  day. From previous equations all energy hub unit output such as power, heat or hydrogen must be less than or equal to the unit rating capacity. The number of wind turbines needed of each  $wt$  (wind turbine type) to be installed can be determine using the following equation. In this equation (4.26), the power that can be harvested by wind turbines at each scenario ( $s$ ) is limited by upper and lower power of single wind mill ( $\mathbb{P}_s^{wt}$ ) multiplied by the total number of number wind turbines ( $y_{wt}$ ). The upper ( $\mathbb{P}_s^{wt}(v_s^{up})$ ) and lower ( $\mathbb{P}_s^{wt}(v_s^{lo})$ ) power of single wind turbine at each scenario corresponds to the upper and lower limits of wind speed of each scenario.

$$y_{wt} \mathbb{P}_s^{wt}(v_s^{lo}) \leq P_s^{wt} \leq y_{wt} \mathbb{P}_s^{wt}(v_s^{up}) \quad (4.26)$$

Hydrogen gas flows from the electrolyzer to the hydrogen tank, where it is stored, until it is directed to the fuel cell when there is need for power generation. In order to keep track of the amount of hydrogen stored at each time, a discretized dynamic mass balance on hydrogen entering and leaving the tank is applied as described in the following equations (4.27)-(4.28). These equations were designed such that for a given scenario, if the hydrogen production is high due to an excess in wind energy, the excess hydrogen will be stored for use at different scenarios that have low hydrogen production as a result of low wind power. The hydrogen level is not stochastic (not function of uncertain scenarios) but it accounts for all possible uncertain wind speed realization scenarios.

$$HL_{h,d} = HL_{h-1,d} + \sum_s \beta_s (H_{d,h-1,s}^{Elyzr} - H_{d,h,s}^{Tank}) \quad , 1 < h < 24, \forall d \quad (4.27)$$

$$HL_{d,h} = HL_{d-1,h} + \sum_s \beta_s (H_{d-1,h,s}^{Elyzr} - H_{d-1,h,s}^{Tank}) \quad , h = 1, d > 1 \quad (4.28)$$

The second equation is added to link between the first hour of the latter day with last hour of the former day. It can be noticed from this equation that the input and output hydrogen flow rates is

weighted and summed by the probability of each stochastic scenario to account for all possible scenarios.

Since the energy storage technology cannot be charged and discharged simultaneously binary variables ( $ch_{d,h,s}$  charging status,  $dis_{d,h,s}$  discharging status) are introduced to track the on/off status for the electrolyzer (i.e., works as charging unit), and fuel cell (i.e., works as discharging unit) at each  $s$  scenario and  $h$  hour of the  $d$  day. In the following equations, the big-M formulation [135] is used to ensure no hydrogen and power flow leave out the electrolyzer and fuel cell when they are off.

$$H_{d,h,s}^{Elyzr} \leq ch_{d,h,s} M \quad \forall h, d, s \quad (4.29)$$

$$P_{elec,d,h,s}^{Fuelcell} \leq dis_{d,h,s} M \quad \forall h, d, s \quad (4.30)$$

Where  $M$  is a big number,  $ch_{d,h,s}$  and  $dis_{d,h,s}$  are binary variables that represent the on and off states of electrolyzer and fuel cell units at each at each  $s$  scenario and  $h$  hour of the  $d$  day, respectively. In order to prevent the electrolyzer (charging status) and fuel cell (discharging states) from running at the same time, the following constraint is added (4.30)

$$ch_{d,h,s} + dis_{d,h,s} \leq 1 \quad \forall h, d, s \quad (4.31)$$

#### 4.4 Results and Discussions

The full-size electricity and heat demand along with the cluster curves generated in section 4.2 are employed as inputs for the present energy hub planning model. The objective cost function is multiplied by a frequency parameter referred to as  $\gamma_d$  (as presented in equation (4.32)) that allows comparing the original demand dataset which has a 1-year time horizon and the clustered cases. The parameter  $\gamma_d$  represents the number of repetitions (frequency) for corresponding  $d$  day or cluster. The parameter  $\gamma_d$  is equal to 1 when the original (full-size) demand data is used, and is equal to the number of days that represent a cluster when the representative cluster curves are used.

$$\begin{aligned} & \min CRF \left[ \sum_u y_u CAP_u E_{max}^u + \sum_{wt} y_{wt} CAP_{wt} \sum_{st} y_{st} CAP_{st} \right] \\ & + \sum_s \beta_s \left[ \sum_d \gamma_d \left( \sum_h \left( \sum_u [P_{i,d,h,s}^u OM_u + NG_{d,h,s}^u Price_{ng}] + \sum_{st} P_{i,d,h,s}^{st} OM_{st} \right) \right) \right] \end{aligned} \quad (4.32)$$

For the rest of this chapter, we will call the energy hub that considers the original (i.e., full-size) hourly heat and electricity demands for 365 days the original energy hub model (original model), whereas the energy hub that take into account 4, 5 and 6 hourly loads clusters from previous section (section 2) (clusters are considered as days) the clustered energy hub model (clustered model). The energy hub optimization problem was developed in GAMS and solved using CPLEX (MILP) solver [15] with relative optimality criterion solver equal to  $10^{-5}$ . The simulations were carried on an Intel(R) core i7 (R) 4.0 GHz, 16 GB RAM personal laptop.

#### **4.4.1 Results Without GHG Emissions Constraint**

Figure 4.6 shows the objective function values along with the relative error of the clustered energy hub model cases in comparison with the original (i.e., optimal) energy hub model. As it can be shown in Figure 4.10, all clustered cases underestimated objective function value comparing to the original case. The objective function values of the normal clustered energy hub cases are closer to the original optimal energy hub solution than the sequence clustering cases. The relative error of the objective functions of the clustered energy hub model compared to the original energy hub model is ranged between -4 % and -10 %. Additionally, the higher the number of clusters the better the solution quality is, for both normal and sequence clustering as the solution gap between the clustered cases and the original case become smaller. Additionally, the average absolute relative error of all weight factor is also presented in Figure 4.10. In other words, higher number of clusters implies more representativeness of the actual data. These errors are inversely proportional to the number of clusters. It can also be concluded that the objective function values for the clustered cases does not vary considerably as a function of the weight factors because both heat and electricity exhibit partially similar temporal trajectory. Inset in Figure 4.10 also illustrates the advantages of clustering applications in terms of solution time. The bar chart in Figure 4.10 displays the average solution time of all weight factor for each clustered case run and the solution time of the original energy hub model. As can be seen from the bar chart with a vertical logarithmic scale that, solving the clustered energy hub models are tremendously faster than the original energy hub model. The average (i.e., between ~50 to 100 second) time required to solve the clustered cases energy hub is shorter by 2 order of magnitude than solving the original energy hub model (i.e., ~7000 second).

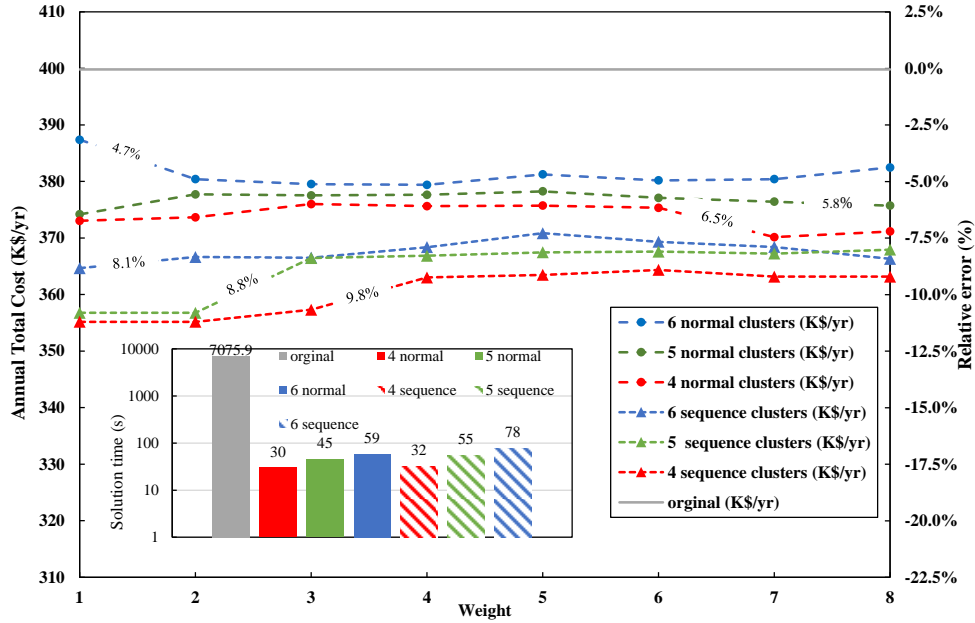


Figure 4.10. Comparison between original and clustered energy hub solution in terms of solution quality and time

In order to examine the multiscale clustering approach of demand data effect on the design results of the energy hub model, Figure 4.11 was generated. Figure 4.11 shows a comparison in terms of decision variables solution between original energy hub solution and clustered cases for weight factor 1, 4 and 8. As can be seen in this figure, the higher the number of clusters, the more closely are the design decision results of the clustered cases to the original case. Moreover, the installed generation capacity is following the same trends. It is worth noting that, the results of weight factor 1 shows slightly better performance, when normal clustering is applied, than other weight factor since it tends to align more with the heat demand. Due to higher inconsistency and fluctuation of the heat demand, a better design decision variable (i.e., closer to the original case) was achieved by prioritizing the heat demand. Additionally, it can be illustrated that in the clustered cases, the decisions of installed capacity for power and heat generation are generally underestimated. Specifically, the power generation capacity design decisions are underestimated by a lower margin than the heat capacity design decision, when compared to the original model. This is because of total heat production rate in equation (4.17) is allowed to exceed demand, if necessary, while an equality constraint is imposed on the power balance to satisfy the electricity demand.

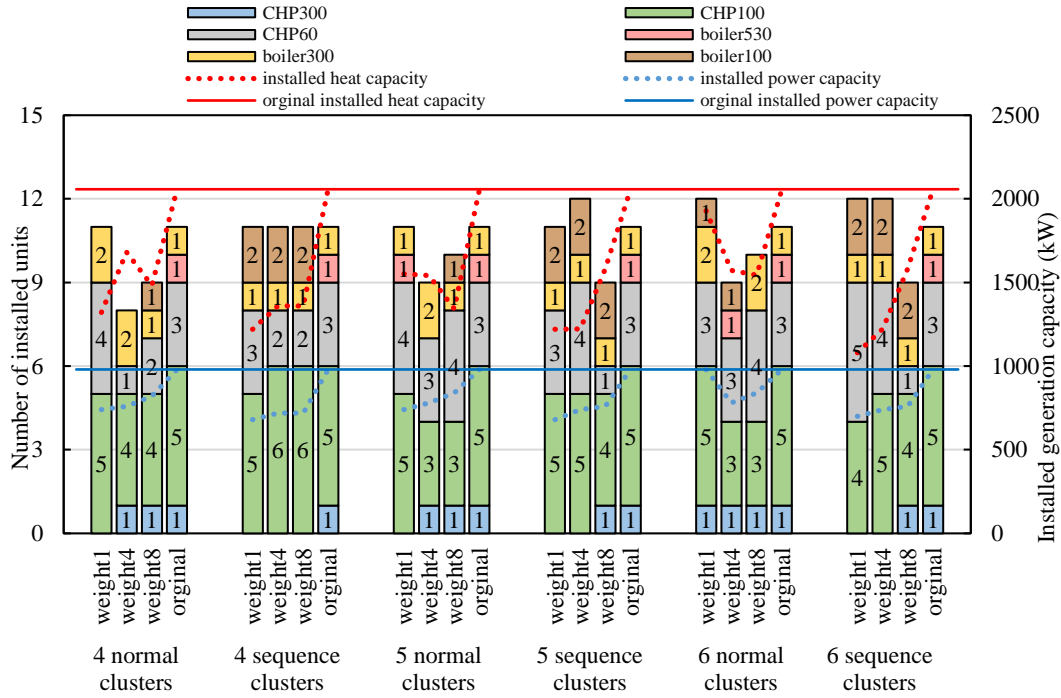


Figure 4.11. Design results comparison between original and clustered energy hub cases

The effect of multiscale clustering approach of the demand data on the energy hub operational decision are depicted in Figure 4.12. Figure 4.12 presents the total energy hub utility production rates for the normal and sequence clustering cases for weight factors 1, 4 and 8. The figure displays the error associated with the total utility production using clustered cases relative to the original case. The figure indicates that the errors in the total production rate from boilers are higher than the errors associated with electricity and heat production from CHP. This is due to equation (4.17) that allows the heat production to be greater than the demand whereas electricity output is fixed to meet the demand. On the other hand, electricity production rate using clustered model is very close to the original model. It can be concluded that using the clustering approach is an effective tool to reduce the size of the original model while maintaining good results. Similarly, one can notice that increasing the number of clusters improves the solution quality, as it closes the gap between the original (i.e., non-clustered cases) and clustered cases. Moreover, as was expected, normal clustering has a better solution quality (i.e., closer design and operational decisions to the full-size energy hub model) than sequence clustering because of the additional restriction added by sequence clustering. The clustered cases energy hub model underestimated the installed design capacities of boilers since they are cheaper (have less effect on the objective function) and as a

result of that, the total heat rate generated by reboilers using the clustered case energy hub model are less than the original case.

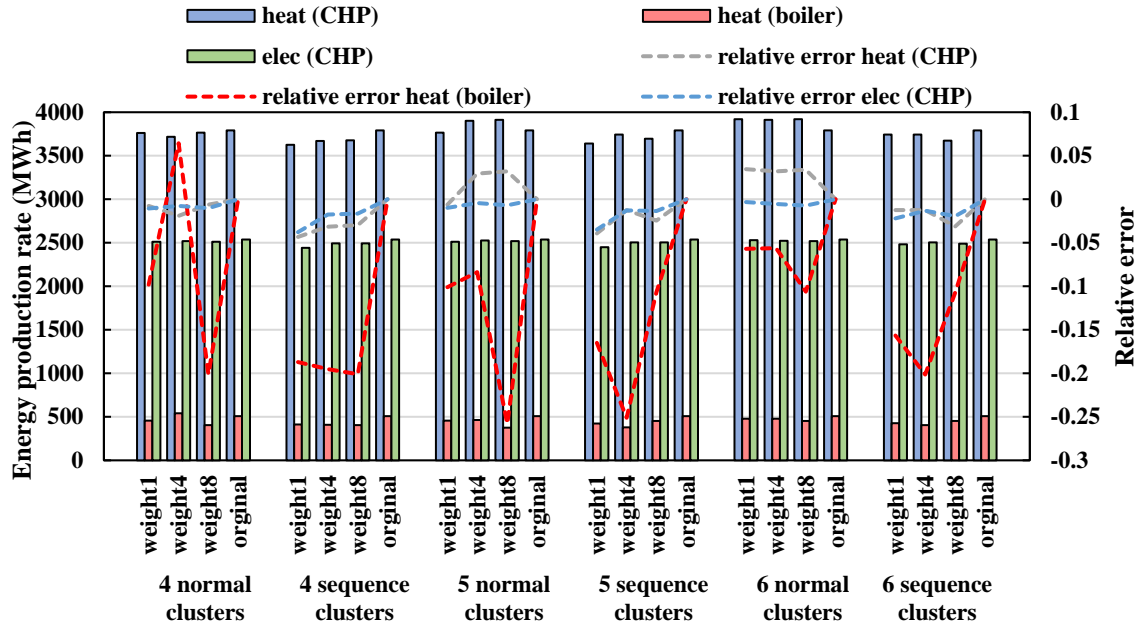


Figure 4.12. Energy hub's utility production rates comparison between original and clustered cases

#### 4.4.2 Environmental Considerations (CO<sub>2</sub> emission regulation)

It can be seen from previous results that according to the current parameter, the optimization program emphasised to have neither single wind turbines nor storing units installed. This is because renewable energy is usually more expensive than traditional fossil fuels. However, renewable energy resources are cleaner alternatives to fossil fuel, and have been widely integrated with current energy hubs and microgrid systems to mitigate GHG emissions (i.e., CO<sub>2</sub>, CH<sub>4</sub>, NO<sub>x</sub>). It is also clear that the essence of this study is to integrate the power system with wind energy. Therefore, in order to force the optimization program to select some wind turbines and energy storage units, CO<sub>2</sub> emission constraint is introduced and imposed onto the energy hub mathematical model as shown in equation (4.33):

$$Em = \sum_s \beta_s \left[ \sum_d \gamma_d \left( \sum_{u,h} (\delta * b * NG_{d,h,s}^u) \right) \right] \leq \alpha \quad (4.33)$$

Where  $Em$  denotes the total equivalent mass of CO<sub>2</sub> emissions from the energy hub system per year.  $\delta$  is the emission factor associated with Ontario's natural gas and it is assumed to be (0.187 kg/kWh)[113].  $\alpha$  is the limits that enforce on the CO<sub>2</sub> emissions. In this case study, only the emissions from fossil fuel units (boilers and CHP) are considered while, emissions associated with renewable energy generation units (i.e., wind turbines) and storage units are ignored since they are relatively negligible compared to the fossil fuel generation units.

A sensitivity analysis on the objective function variable and the CO<sub>2</sub> emissions was done to check the validity of our mathematical problem and see if the optimization will force to install some wind turbines and energy storing facilities. Figure 4.13 shows the change of the total annual cost and the number of wind turbines that need to be installed as a function of CO<sub>2</sub> emission ( $\alpha$ ). The figure was generated using a clustered case energy hub model with 6 normal and sequence clusters and of weight factor 4 (50% emphasis on heat and 50% emphasis on electricity data) since they are a better representative (have lower IAE) of the whole year demand data. As it can be noticed that there are upper limits for ( $\alpha$ ) which occur at the highest total annual cost. This happens when the emission constraint is not active (same solution of section 4.4.1) and the number of wind turbines that need to be installed are zero. Following this, when the value of ( $\alpha$ ) decreases, the objective function (total annual cost) increases and the optimization program forces the installation of wind turbines at the same time. The greater the reduction in the amount of CO<sub>2</sub> emissions, the higher is the number of wind turbines that are picked by the model to be installed at more expensive total annual cost. It is worth noticing that the trend of results from the energy hub optimization model when using both normal and sequence clusters are nearly the same as function of the CO<sub>2</sub> emissions reduction. Furthermore, at higher level of CO<sub>2</sub> emission, reducing the CO<sub>2</sub> emission has a lower effect on the objective function; but at lower CO<sub>2</sub> emission level, reducing the CO<sub>2</sub> emission comes with additional cost that arise from installing more storage units to help dispatch the wind power more efficiently.

It is proposed in this study to reduce the CO<sub>2</sub> emission ( $\alpha$ ) by 20% from its upper limit (the CO<sub>2</sub> at the lowest cost when there is no limit to CO<sub>2</sub> emissions). All the following case studies in the remaining sections will use the above emission guideline.

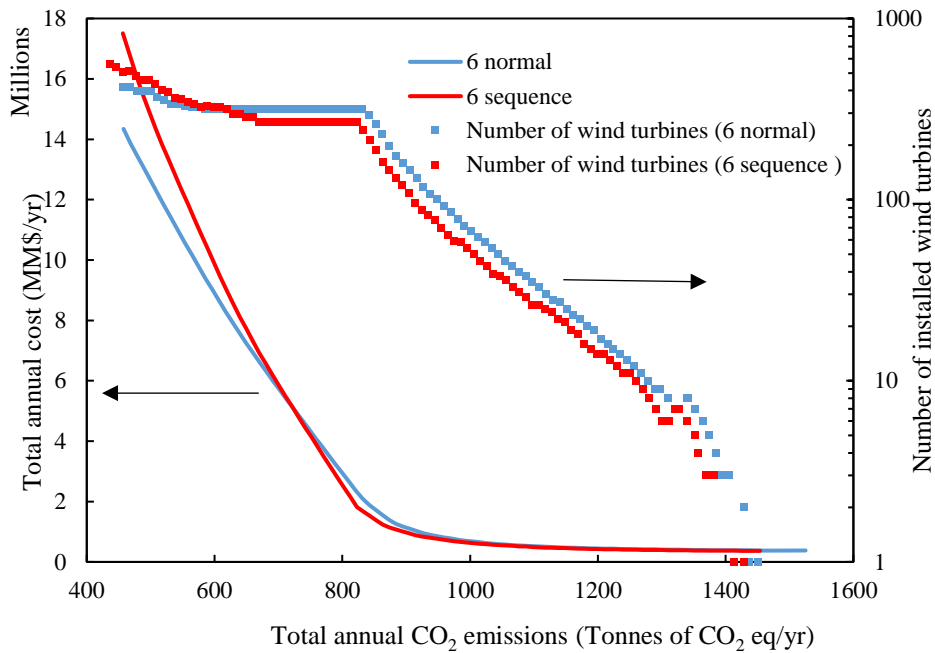


Figure 4.13. The effect of CO<sub>2</sub> emission regulation on the objective function (lines) and number of wind turbine needed to be installed (square marker)

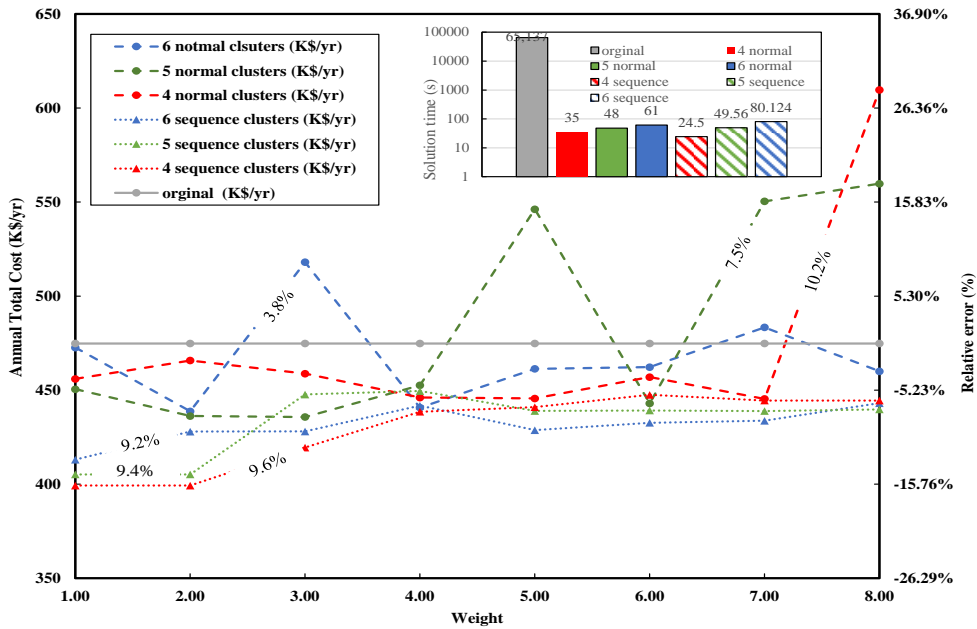


Figure 4.14. Comparison between original and clustered energy hub solution in terms of solution quality and time under CO<sub>2</sub> emissions restriction



#### 4.4.2.1 Results with GHG Emission Constraints

The effects of weight factor and number of clusters on the objective function value are illustrated in Figure 4.14 when the CO<sub>2</sub> emission constraint is active. The objective function values along with the relative error of the clustered and original energy hub cases are shown in this figure. When clusters emphasis more on heat demand (at weight factor 1), the results of the clustered cases energy hub (i.e., total annual cost) are much closer to the original energy hub case. It can also be seen from Figure 4.14, that for normal clustering at weight factor 8 (when prioritizing the electricity demand) the highest deviation (highest relative error) from the original case is occurred. In between weight factor 1 and 8 there is no clear relation between the weight factor and the solution quality of the clustered cases. Sequence clustering results exhibit less variability as priority switches from heat to electricity. Furthermore, the average of the absolute relative error for all weight factors are also presented in Figure 4.14. The average relative error values are converging towards each other (i.e., reducing by higher number of clusters), and normal clustered cases have slightly less average error than the sequence clustered cases. The average solution time for all weight factor of each cluster case run (i.e., 4, 5 and 6 clusters) along with the original energy hub solution time are displayed in the same figure (inset Figure 4.14). it can be realized from the bar chart that the time required to solve the clustered cases energy hub model are tremendously shorter than the original energy hub model. The solution time of the original energy hub (i.e., ~ 65137) case is greater by 3 orders of magnitude than the average solution time of clustered cases energy hub (i.e., between ~50 to 100 second). Also, one can observe that solving the energy hub model with considering the carbon emission regulation (Figure 4.10) is much faster than if there is no environmental consideration (it is less by 1 order of magnitude). When the GHG emissions constraint is active, the optimization program decided to install storing and wind turbines units in order to keep the carbon emissions within the desirable level. As a results of that, larger number of non-zero variables (e.g., continuous variables associated with power flow to/or from the storing units, hydrogen flow rates, power directed from wind turbines and binary on/off variables for charging and discharging storing unit) are handled by the optimization problem, hence, the degree of complexity is boosted. On the other hand, there is no significant differences in solution times of the clustered cases energy hub when the environmental constraint is considered or not

The effects of number of cluster and weight factors on the design decision variables of the energy hub model when the GHG emissions constraint is active are demonstrated in Figure 4.15- Figure

4.17. Figure 4.15 shows the design variables solution of the fossil fuel units as a function of all clustered cases runs with weight factor 1, 4 and 8 along with the results of the original energy hub model. As it can be seen from this figure that the weight factor has no significant effect on the design decision results. The higher number of clusters the closer the design decisions of the clustered cases to the original (e.g., 5 normal clusters have same number of CHP100 units as the original case and the 6 normal clusters with weight factor 1 has the exact same design of the original case). This can be better demonstrated by Figure B.12 in Appendix B which shows the installed capacity of power and heat generation for all clustered runs with weight factor 1, 4, and 8 along with original case. Furthermore, in all clustered cases, the optimization program avoided installing any CHP300 and boiler530 units which aligns with the same results suggested by the original energy hub, as these two units are the largest units that powered by natural gas which can correspond to the highest carbon emission.

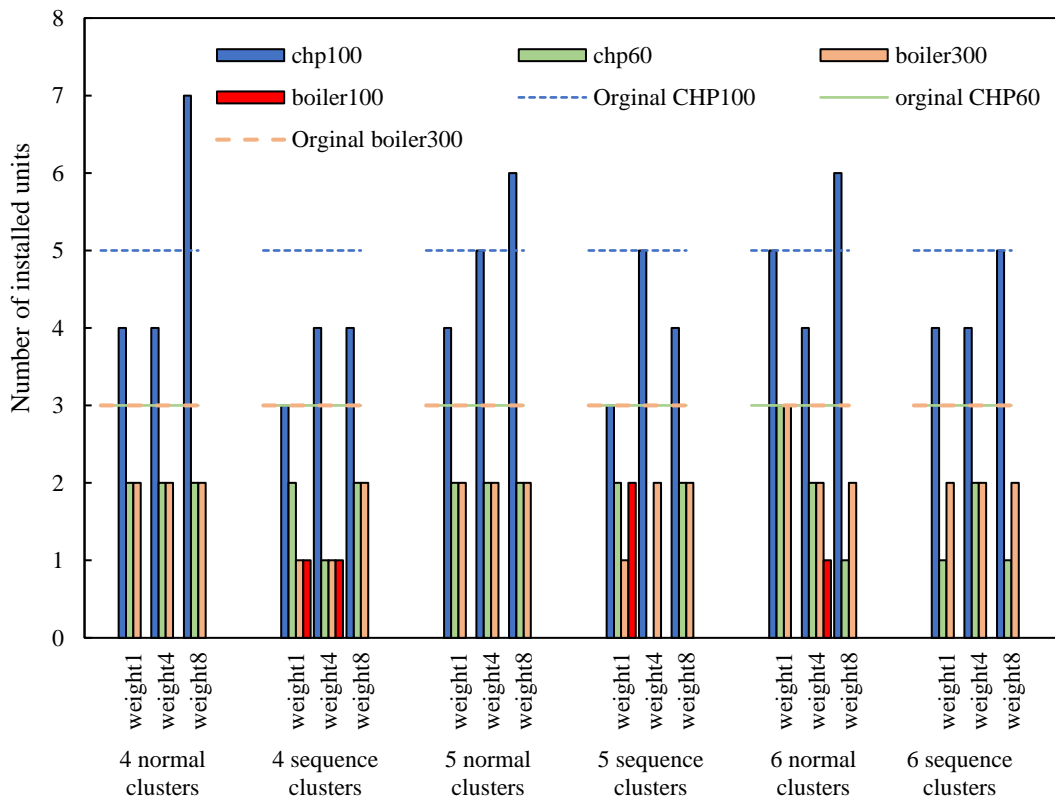


Figure 4.15. The number of energy hub units powered by fossil fuel that are installed under CO<sub>2</sub> emissions regulation

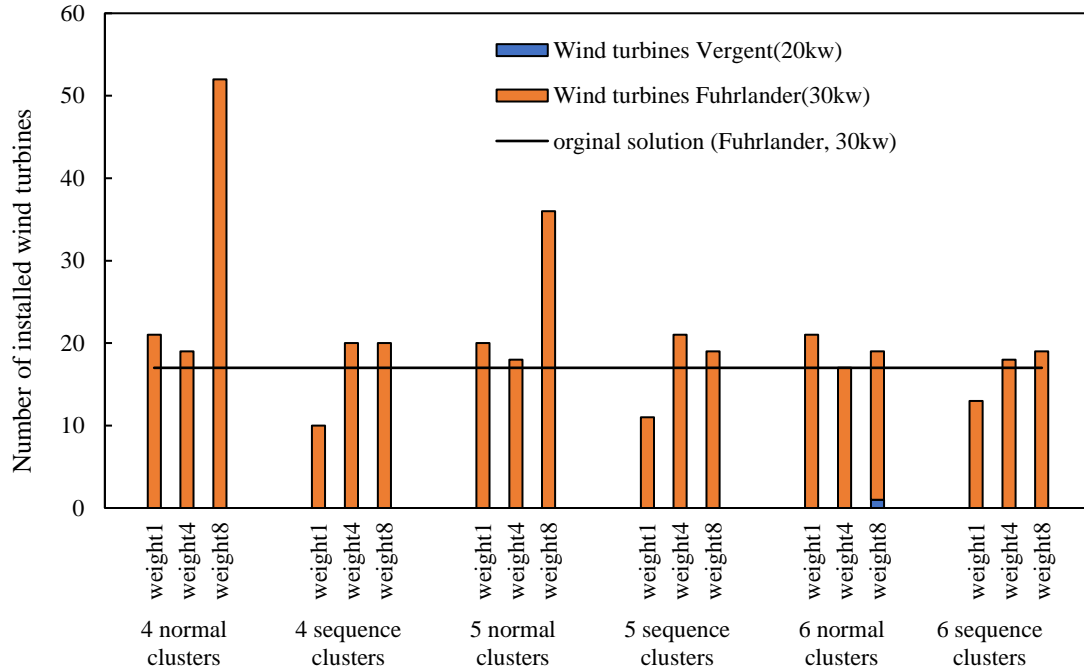


Figure 4.16. Number of installed wind turbines suggested by original and clustered cases under CO<sub>2</sub> emissions regulation

The number wind turbines that needed to be installed for all clustered cases runs with weight factor 1, 4 and 8 along with original case model result is displayed in Figure 4.16. The figure indicates that, as the number of clusters increase, the gap between the number of wind turbine suggested by cluster cases and original case is reduced. At weight factor 8 (prioritizing electricity demand) for 4 and 5 normal, the optimization model overestimated the number of wind turbine by a large margin. This can explain the high error presented in Figure 4.14 for the value of the objective function of those two cases.

Figure 4.17 shows optimal number of storing units of the energy hub model under CO<sub>2</sub> emissions regulation using clustered and original data. As it can be seen from this bar chart most of clustered cases (i.e., sequence and normal clustering) storing units results are in very good agreement the original energy hub model results. It also can be noticed that some of sequence clustering results are overestimating the number of hydrogen tank needed.

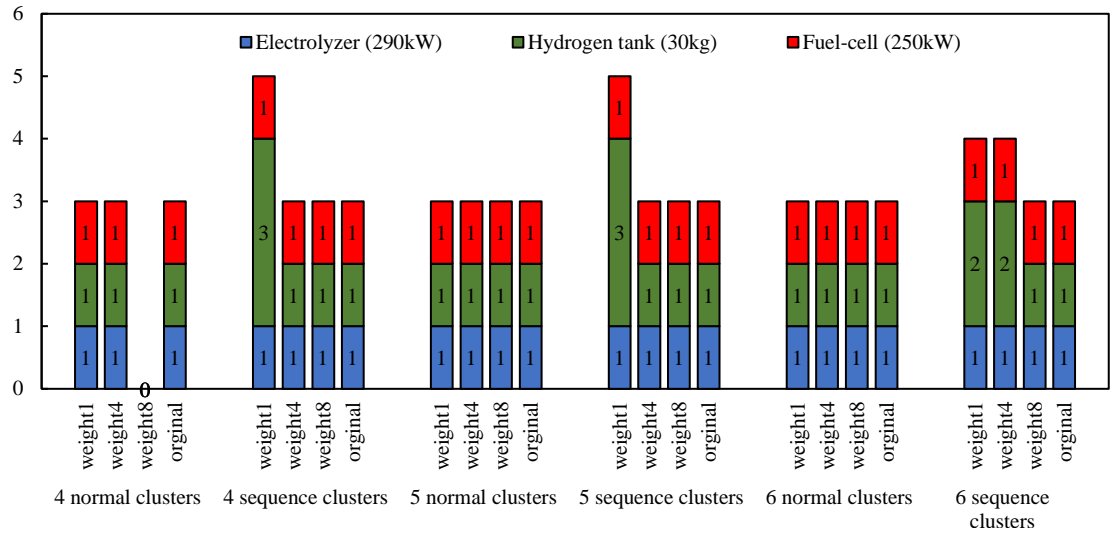


Figure 4.17. Number of installed storing facilities suggested by original and clustered cases under CO<sub>2</sub> emissions regulation

To examine the operational decisions solution quality of applying the multiscale clustering approach of the demand data on the energy hub under CO<sub>2</sub> emissions regulation, Figure 4.18 and Figure 4.19 are generated. The figures illustrate the total energy hub's utilities production rate of each unit including heat, electricity and hydrogen for clustered and original case. The total utility production of each unit is calculated by summing up the unit's production over the year for each stochastic scenario and taking the weighted probability sum of all scenarios. Figure 4.18 depicts the utilities production of units powered by fossil fuel (i.e., CHPs, and boilers); while Figure 4.19 displays the total utilities produced by wind turbines and storing units (i.e., electrolyzer and fuel cell). The figures also show the relative error in the total utilities production using the clustered cases with respect to the original case. Figure 4.18 shows that all utilities production rates of clustered cases are in very good agreement with utilities produced from energy hub when the full-size demand data is used. There is no significant variation between the amount of heat and electricity produced by CHPs for all cluster and original case. However, when using sequence clustering, the relative error associated with boilers heat production is high. In Figure 4.19, there is a larger degree of deviation between clustered cases and original case results in the total amount of electricity produced from wind turbines and fuel cell. Furthermore, this deviation from the original case results is even bigger for sequence clustering as noticed in the same figure. Despite these errors, the proposed clustering approach can still be considered as a powerful size reduction tool. This is because the design decision variables of clustered cases are close to the original, and

the total production rate of heat and electricity (Figure B.13 in Appendix B) from all clustered cases are very close to the original and their relative error does not exceed 20%.

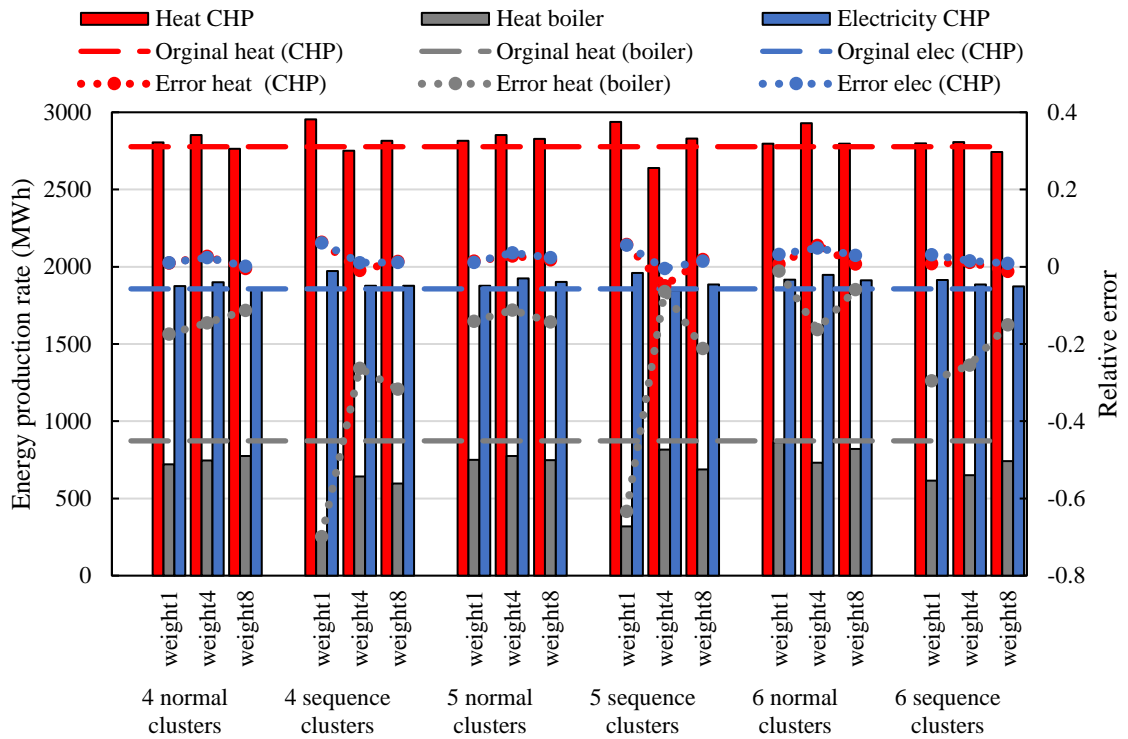


Figure 4.18. Comparison between original and clustered cases utilities production rates of energy hub units powered by fossil fuel under CO<sub>2</sub> emissions regulations

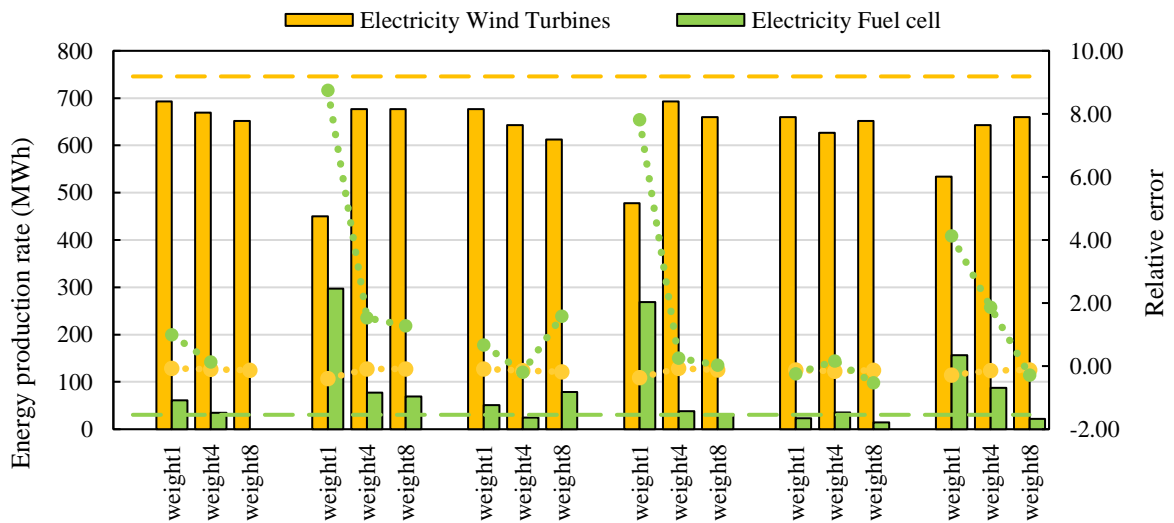


Figure 4.19. Total utilities produced by wind turbines and storing units for clustered cases and original cases under CO<sub>2</sub> emissions regulation

### 4.4.3 Stochastic Energy Hub Formulation Assessment

In order to assess the benefit of the current energy hub model formulation and its ability to store energy under different wind scenarios, the average power sent to the electrolyzer and the average power received from the fuel cell for each stochastic scenario are displayed in Figure 4.20 and Figure 4.21, respectively. For simplicity, the yearly average power of each hour of the day for each scenario is displayed (the average power flow of each hour with respect all year days for each scenario). The energy hub model with the 6 normal clusters and weight factor 4 is used to generate the results in these two figures. Using clustered case data for this assessment is easier since its solution produces smaller size of data than the original case; where these clustered cases mimic and follow the behaviour of the original case solution. The usage of the clustered energy hub model in the current assessment shows a direct applicability of implementing the proposed clustering method. From Figure 4.20, the rate of charging (i.e., power directed to electrolyzer to produce hydrogen) is higher with higher wind speed scenarios, which means that more energy is stored when the availability of wind energy is high. As an increase in scenario number indicates an increase in the wind speed as well. Furthermore, at times when demand is relatively low the optimization model stores more energy. On the other hand, it is an evident from Figure 4.21, that the rate of discharge from the fuel cell is inversely proportional to the wind speed scenario number. Additionally, most of the discharging power from the fuel cell happens when the demand is the highest.

To examine the efficiency of the stochastic programming method, the value of stochastic solution (VSS) is calculated. [20] stated that the VSS helps in determining whether it is beneficial to fix the first stage decision variables in the stochastic optimization problem based on the solution obtained from the expected value (EV) problem. In other words, VSS represents the extra cost paid by the decision maker for not considering stochastic programming method (not considering uncertainties). In order to estimate the VSS, the solution to the (EV) problem needs to be determined. The EV problem in our case is the solution of the deterministic energy hub optimization problem by utilizing the expected value (i.e., mean) of the uncertain parameter (i.e., wind speed). In the next step, the first stage decision variables (design decision in our case study), obtained from the expected value problem (EV), are used as input parameter and fixed in the two stage stochastic energy hub optimization recourse problem (RP). Subsequently, solving the resultant RP upon fixing the first stage decision variables called expected result of using the EV

solution (EEV). The EEV gives the solution to the second stage decision variables when the first stage decision variables have been fixed. VSS is represented by the difference between EEV and the RP.

Table 4.7. Values of objective function for the RP, EV and EEV problems

	Results	EV	EEV	RP	VSS
Objective function: total annual cost (\$/yr)	Without environmental consideration	379411.5	379411	379411	0
	with environmental consideration	438717.6	455561.6	440729	14832.66371

Table 4.7 presents the solution of the EV, EEV and RP for the energy hub model without and with environmental constraint (GHG emission regulation). The results of this table were obtained using the 6 normal clusters with weight factor 4 as demand data for the energy hub model since they are a better representative (have lower IAE) of the whole year's demand data. From this table, it is clear that when there is no environmental consideration, no benefit is gained by solving the stochastic programming problem (VSS = 0).

In contrast, when the emission constraint was active, the VSS is estimated to be 14832 \$/yr (VSS = EEV-RP). The positive VSS value proves that considering uncertainty in the modelling of the energy hub is beneficial. Additionally, although the EV (deterministic solution) has the lowest objective function, deterministic formulation solution is insufficient because it relies on a relatively small segment of information (average wind speed) that does not sufficiently explain the real wind speed behaviour (i.e., not true representatives of the annual wind data). Therefore, it can be said that wind uncertainty has a strong effect on the optimization solution when environmental regulations are considered as proven by the value of VSS.

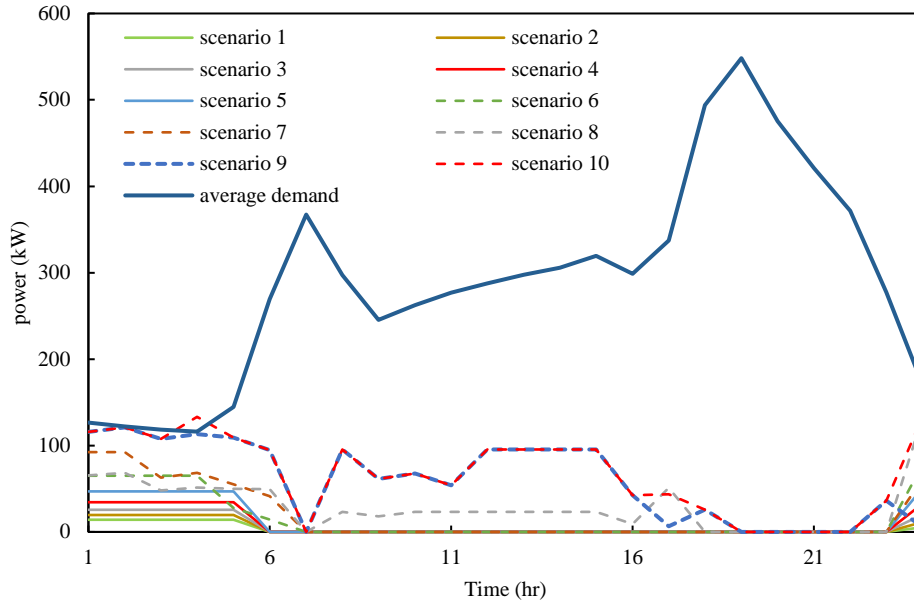


Figure 4.20. Average charging power for each stochastic scenario

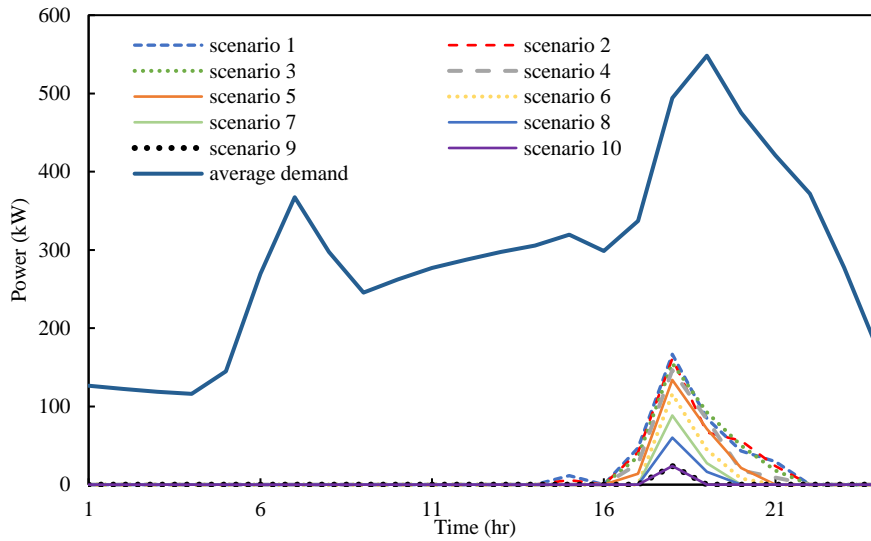


Figure 4.21. Average discharging power for each stochastic scenario

## 4.5 Conclusion and Future Work

The present work targets the multiscale stochastic energy hub modelling using a clustering approach. The clustering approach was employed to reduce the model size by representing the yearly days by “typical” days representatives of the operating year, as considering shorter time



periods (e.g., hour) for the whole year data, results in a larger and intractable model. The clustering problem with multiple attributes was modelled based on mathematical programming approach which resulted in a multi-objective optimization problem. The weighting method approach was utilized to tackle this problem. Although the approach was simple, the computational complexity of the clustering algorithm was evident as its computational (i.e., running) time was long. Therefore, heuristic size reduction approach based on the general formulation clustering approach was employed to cluster the given demand data in shorter times. Results shows that heuristic approach can reduce the clustering running time by 2 orders of magnitude than the general clustering approach, and generates very close clusters (i.e., close clustering measure) to the original clustering approach. The present clustering algorithm (include heuristic approach derived from it) features many unique characteristics that gives it advantages over other clustering approach. One of these features is the ability to attain normal and sequence clustering. Another feature is its flexibility to change the internal clustering measure, therefore, different type of clustering measures can be applied. A further feature is the ability to tune attribute weights which offer to the decision maker the ability to prioritize attributes that are more important. For example, in this case study, it was concluded that giving priority to the demand data with higher variability enhanced the solution of the energy hub model (i.e., closer to the solution of energy hub model under full-size demand data).

A Weibull distribution was used to model the intermittent behaviour of wind speed data. The design and operation of energy hub system was modelled as stochastic problem under uncertain wind speeds utilizing clustered demand data (heat and electricity).

The multiple clustered demands applied to the stochastic energy hub model to reduce the model size. Results when there was no CO<sub>2</sub> emission regulation, indicated that the relative errors of the reduced size energy hub objective functions with respect to the full-size mode were ranges between -4% and -10%. It should also be stated that the time required to solve the clustered energy hub model was shorter by 2 orders of magnitude than solving the full-sized energy hub planning model. The effect of the clustering approach on the design and operational decision of the energy hub model was assessed. It was concluded that the long-term decisions (i.e., design decision variables) of clustered cases were in very good agreement with the full-size energy hub mode for both cases of GHG emissions regulation. Similarly, most of the operational decisions represented by the total production rate of utilities using clustered model, were close to the full-size energy hub model

when the GHG regulation was not active. On the other hand, a larger degree of deviations was noticed when GHG emission constraint was active. Regardless of these deviations, the total production rate of heat and electricity from all clustered cases were very close to the original case (full-size model) and their relative error did not exceed 20%. Furthermore, the results show that a closer objective function to the full-size model was achieved when the number of clusters increases for both normal and sequence clustering. Normal clustering results were found to be better than sequence clustering in terms of both objective function and multiscale decision variables. It can be concluded that using the clustering approach is an effective tool to reduce the size of the original model while maintaining good results.

It was demonstrated from the example of adding GHG emissions regulation to the full-size stochastic energy model, that stochastic model complexity can be boosted by adding extra constraints or considering more stochastic scenarios. Therefore, the reduction of multiscale stochastic energy hub model size by applying the multiscale clustering approach become crucial. Applying the suggested demand reduction method will allow decision maker to study different cases of energy hub model (e.g., using different hub architecture, changing the number of stochastic scenarios, and adding more storing) and obtain satisfactory solution at reasonable time. As it was proven from this study that the solutions (design and operational decisions) of solving energy hub model with reduced size demand are very close to the solution of full-size energy hub model.

The developed stochastic energy hub model showcases the advantages of the current formulation where the model suggestions (solutions) considered the uncertain behaviour of wind energy. Additionally, it can be stated from the assessment done on stochastic formulation model that wind uncertainty has a strong effect on the optimization solution, when environmental regulations were considered, as proven by the positive VSS. As the VSS indicates the extra cost that the decision maker has to pay for not using the stochastic programming method.

Future works can include the application of the proposed clustering approach to different multiscale planning problem. The stochastic energy hub planning model can be extended to include capacity expansion planning decisions to satisfy multiple attributes demand. It would be interesting to use forecasted demand data to plan energy hub system, as this case study was limited to implement historical demand data into multiscale modeling. Therefore, forecasting techniques can be employed to forecast the future demands; clustering approach will be applied to reduce the

size of these multiple attributes demand where they can be used as an input to the energy hub planning or capacity expansion model. Another example of future work is that the multiscale clustering approach can be applied to superstructure modelling approach to design new chemical or power plants. Therefore, instead of solving the superstructure model for a 1-day profile that represents the whole year, it can be solved for several representative days that are more likely to reflect the real behaviour of demand. Furthermore, the clustering approach that was proposed in chapter 3 to generate stochastic scenarios can be used in this case study to generate reduced size wind speed scenarios.

# **Chapter 5      Machine Learning Approach for Modeling and Optimization of Complex Systems: Application to Condensate Stabilizer Plant**

## **5.1 Introduction**

Gas condensate is a valuable liquid product that is recovered from natural gas. Condensate is present in raw natural gas from many natural gas fields as low-density mixture of hydrocarbon in the form of both liquids and gaseous components. After it is recovered from natural gas, it can be converted to different petroleum fuel product (i.e., jet fuel and gasoline) or used to dilute the heavy crude oil [136]. Raw condensate can be separated from natural gas using a multiple phase separator, however condensate in its natural gas form cannot be stored or exported. Therefore, gas condensate must undergo treatment, where it will turn into commercially acceptable form for storage, exportation, and transmission purposes. Gas condensate treatment typically includes separation of dissolved light hydrocarbon gases components (i.e., methane and ethane) along with lowering its sulfur contents (i.e., hydrogen sulfide, mercaptans, etc...), reducing water and salt contents to the desired standard levels [137]. Additionally, the vapour pressure of the processed condensate should be within certain recommended range to prevent condensate from forming a separate gas phase in pipelines and storage tanks as light component tend to escape the processed condensate. Reid vapour pressure (RVP) is commonly used as a measure of the volatility of the condensate and other petroleum products (e.g., gasoline), so the higher the RVP the more volatile components are in the condensate. Therefore, raw condensate should undergo stabilization to meet the required specifications. In stabilization process, the light end components are stripped out from the heavier hydrocarbons which will reduce vapour pressure of condensate along with their RVP and hence, the formation of the vapour phase will be avoided when transferring them to atmospheric tanks. Typically, flash vaporization or fractionation processes can be utilized to stabilize the condensate, However, condensate stabilization by fractionation is more common choice in industry because it can produce wide range of condensate specification (i.e., required vapour pressure) with proper operating condition in single tower [12]. Stabilization process also involve reducing the sulfur content of the stabilized condensate into environmentally safe levels.

As the presence of sulfur component within condensate (mostly H<sub>2</sub>S) leads to significant corrosion problems, as well as it is considered a very toxic gas.

Reliable and accurate stabilization process modelling can predict product specification under different conditions, and help studying different scenarios of operating conditions. Also, it can be used to optimize operating conditions with the objective to minimize the operational costs. However, stabilization is a complicated chemical process, and modelling it based on detailed mass and energy balances, will require significant effort, and is computationally expensive to solve. Moreover, it would be even more complicated to solve these detailed models when they are combined with optimization routine [10], [11].

Modelling of stabilization process can also be developed using available commercial simulation software in which they can obtain accurate results. However, these commercial software are not open-source and compiling them with an optimization framework is a challenging procedure to apply [12].

The recent advances in machine learning methods have made input-output modelling approach more usable as approximation surrogate models using plant data or data generated from commercial software. As machine learning models have proven their ability to generate accurate alternative models. Additionally, the availability of plant data is another factor that let machine learning gain more significant attention due to its ability to deal with massive amount of data. Nevertheless, real data should be handled with caution as it isn't devoid of missing points, outliers and faulty measurement, and using them without pre-processing could lead to inaccurate prediction models.

Several studies had been conducted in which machine learning methods were applied to model different chemical process. For example, Kazerooni [51] developed an ANN model that predicts H<sub>2</sub>S content and the RVP of condensate stabilizer plant based on plant data. In this study, a small set of data that include only two features was used to train the ANN. Another study implemented SVM regression to predict the condensate RVP and sulfur content based on real plant data [12]. Nevertheless, both of these studies did not perform noise removal from plant data nor process optimization. Salooki *et al* [138] implemented ANN to predict outputs of the regenerator column in a gas sweetening plant using experimental data. Design experiment method along with statistical regression analysis were applied by [139] in their study to generate different surrogate models for natural gas treatment based on data obtained from commercial software. Afterwards, these models

were used into optimization model to perform process optimization. However, only few data points were used to generate linear and second order polynomial that were more likely to not capture the underlying plant behaviour. In a recent study conducted by Shalaby *et. al.*, [140] developed a machine learning approach to predict CO<sub>2</sub> post-combustion capture unit and performed optimization over the developed models, however data were generated from commercial software (gPROMS) [140].

To best of our knowledge there were no studies that implemented the integration of machine learning models based on plant data into process optimization modelling in comprehensive way. Therefore, the main goal of this chapter is to construct input-output machine learning approach models that can predict condensate stabilizer behaviour, and be used in condensate stabilizer process optimization. In this study, large size plant data are used to build different machine learning models. Before building the data-driven models, different outlier's detection methods are implemented and the one that corresponds to the best linear regression score is used to clean the data. After that, clean data are undergone feature selection procedure, to test if removing some input variables will improve prediction accuracy. Then, cleaned data are used to train different machine learning models that includes linear (Lasso and Ridge regression) and nonlinear (SVM and deep ANN) models. Detailed model developments that include tuning models' parameters are presented. Comparison between these models is conducted and the best model is selected to be integrated into process optimization model. The best model can serve as an accurate and more convenient replacement of detailed first principle models or plant data. An optimization framework based on trust-region constraint algorithm is proposed. Firstly, the selected machine learning model that predict condensate specification (i.e., RVP, water content and H<sub>2</sub>S content) are integrated as constraints into the optimization framework. The purpose of the optimization is to minimize the energy consumption represented by reboiler flowrate while satisfying the condensate desired specification. Then further developments are added to the optimization framework where another machine learning model to predict the steam flowrate is developed and integrated within the optimization objective function. Figure 5.1 shows schematic of the integration of data-driven prediction models (machine learning) into the process optimization of condensate stabilizer. The development of surrogate-based optimization in this chapter can serve as a general framework that can be applicable to a wide range of chemical process.

The rest of this chapter is organized as follows: A description of the stabilization process is presented in the following section (section 5.2). After that, data preprocessing that include outlier removal, feature selection and normalization is presented in the third section (section 5.3). The fourth section includes different machine learning models developments and model validation (section 5.4). The proposed process optimization framework is explained in the fifth section (section 5.5) followed up by a conclusion in the sixth section (section 5.6).

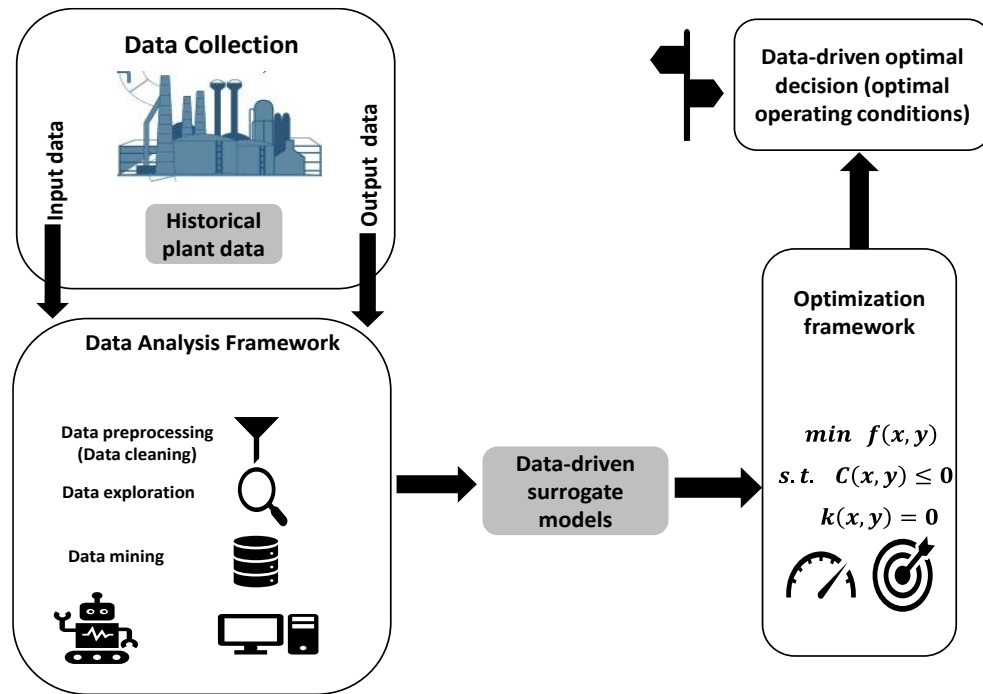


Figure 5.1. Schematic representation of the proposed data-drive surrogate-based optimization framework

## 5.2 Condensate Stabilizer Process

Natural gas originating from the wellbore, is a multiphase mixture containing solids, water and mixture of hydrocarbons that includes volatile light hydrocarbons (i.e., condensate) which can cause dangerously high vapour pressures in upstream processing equipment. The condensate stabilization process is a natural gas treatment process that recover light hydrocarbon from natural gas to store it or use it as a fuel. The stabilization process takes place at early stage of natural gas processing. A schematic representation of condensate stabilization process is shown in Figure 5.2. Firstly, the coming gas from the field enters slug catcher followed by one or more multi-phase separation units where primary gas/condensate/water/solid separations are taking place. Water

and solids from this stage are processed in a separate water treatment plant for disposal. While the condensate from the separator is fed to the stabilizer feed drum to provide the feed of the stabilization tower. After that the liquid is stabilized through a stabilizer column. The stabilizer feed typically enters the top of a packed or a tray-type reboiler absorber column. As liquid falls through the column from tray to tray, heavier hydrocarbons are stripped out from the gas and absorbed by the liquid. Therefore, falling liquid becomes leaner in light components and richer in heavy components. Heat is added to the bottom of the column through a reboiler that is powered by low pressure steam. Column liquid is circulated through the reboiler where it evaporates, and the formed vapour is returned back to the bottom of the column. This circulation process provides a series of stage flashes which drives the separation process. The bottom product (condensate) of the column is cooled to prevent flashing of vapours and sent to the storage. The overhead gas leaving the top of the column is either used as a fuel (be sent low-pressure fuel gas system) or recompressed and combined with the sales gas or fuel gas.

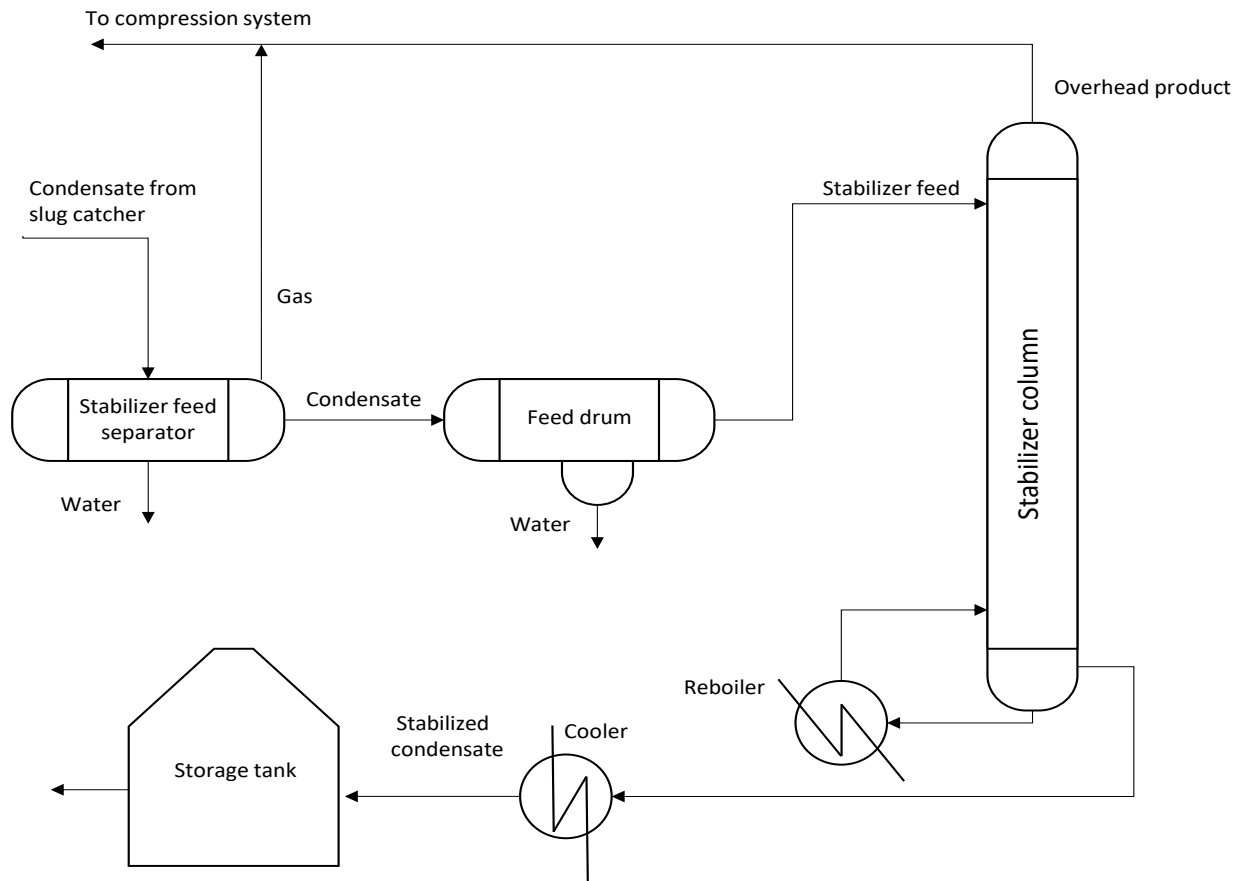


Figure 5.2. Schematic representation of condensate stabilizer process



### 5.3 Data Pre-processing

The stabilizer plants considered in this study are part of two different natural gas treatment complexes located under study in southwestern part of the Emirate of Abu Dhabi in the United Arab Emirates (UAE). The two natural gas complexes are known by Hab. 5 and Hab. 3, respectively. The understudy data were collected for two different stabilizer plants at two different natural gas complexes. The plants are named plant 1 (for Hab. 5 complex) and plant 2 (for Hab. 3 complex) for the rest of the study. The data for the two locations were collected in hourly basis for 1 year.

For Plant 1 data, two set of data were considered. The first set of data were collected during a span of one year at hourly basis. While the second set of data were also collected for 1 year but, in daily basis. Since the output variable of this data set (second one) was the lab measurement of water content which was measured once a day. So, every single measurement of water content was corresponding to the average of the input process variables for the same day (hourly process variable data (24 hour) were averaged for 1 day). Both set of data has the same features (process variables), while the response variables for the first set were RVP (psi) and H<sub>2</sub>S content in (ppm), while the response variable for the second is the water content in volume percentage (vol%). Accordingly, plant 1 has two data sets which we will call them hourly (the first one) and daily set (the second one that will be used to predict water content). Table 5.1 reports a preliminary list of identified variables of Plant 1 where X holds the process variables and y holds the performance (output) variables. Moreover, Table 5.2 and Table 5.3 report a summary of plant 1 hourly and daily data sets respectively. Figure 5.3 and Figure 5.4 display scatter matrix that can show the variation (distribution) of each data variable and the relation between every pair of variables for plant 1 hourly and daily data sets respectively.

For Plant 2 data, two identical stabilizer plant that work in parallel mode located in Abu Dhabi Hab. 3 complex plant were considered. The data for the two plants were merged because the two plants are identical and operate under the same conditions. Additionally, a larger data set for machine algorithm training is more favourable. In most cases, using larger data set for training machine learning models results in more accurate and robust models. Plant 2 list of input and output data are shown in Table 5.4. The inputs and outputs and their limits are listed in Table 5.5. Scatter matrix plot for plant 2 data is shown in Figure 5.5.

Table 5.1. List of input and output variables of plant 1

<b>Process/feature variables (predictors) - X</b>	<b>Performance/target variables (response) - y</b>
Inlet gas flowrate, temperature, and pressure	For the first set of data RVP, H <sub>2</sub> S content, of stabilized condensate
Reboiler temperature and steam flow rate	For the second set of data H <sub>2</sub> O content of stabilized condensate
Column temperatures and pressures	
Overhead product flow rate, pressure and temperature (gas top product stream)	
Condensate flowrate, temperature (liquid bottom product stream)	

Among all these data points there were some missing values and some negative reading for non-negative measurements which are obviously wrong. False data readings were first removed from the data sets. After that, data visualizations were performed to see if there are any pre-processing (adjustment or modification) needs to be done. It was noticed that 2 series of data were given for the flowrate of the condensate leaving the stabilizer column of plant 1. After inspection, it was realized that these two scenarios representing the flow rate of two pumps that works in alternating manners. Therefore, the summation of these two flowrates represents the total condensate flowrate that leaves the bottom of the column. Figure 5.6 shows the variation in condensate flow rate for pump A and pump B and the total flow that leaves the column.

Table 5.2. Summary of ‘plant 1’ hourly raw data set

<b>Variables</b>	<b>Unit</b>	<b>Variable detail</b>	<b>Count</b>	<b>Mean</b>	<b>Std</b>	<b>Min</b>	<b>Max</b>
<i>Response variables (output variables)</i>							
<b>RVP</b>	psi	RVP	8016.00	2.13	1.12	0.01	7.25
<b>H<sub>2</sub>S content</b>	cf	H <sub>2</sub> S_content	8016.00	10.32	5.34	0.00	20.00
<i>Process variables (input variables)</i>							
<b>Feed flowrate</b>	m <sup>3</sup> /h	Inlet flowrate	8016.00	7.82	3.21	0.14	26.52
<b>Feed temperature</b>	°C	Inlet temperature	8016.00	31.91	6.58	13.76	48.41
<b>Column temperature</b>	°C	1	8016.00	130.92	4.37	108.63	147.82
	°C	2	8016.00	120.44	5.11	62.71	134.77
	°C	3	8016.00	102.71	8.46	50.33	123.22
	°C	4	8016.00	84.82	10.85	23.74	115.43
<b>Column pressure</b>	barg	A	8016.00	2.43	0.14	2.07	2.98
	barg	B	8016.00	2.53	0.14	2.14	3.06

<b>Condensate flowrate</b>	m <sup>3</sup> /h	A	8016.00	5.47	5.92	0.03	29.13
	m <sup>3</sup> /h	B	8016.00	9.73	5.95	-0.02	31.95
<b>Condensate temperature</b>	°C		8016.00	103.42	6.69	74.91	126.05
<b>Reboiler temperature</b>	°C	Reboiler temperature	8016.00	158.36	0.98	152.46	164.14
<b>Steam flowrate</b>	kg/h	Steam flowrate	8016.00	625.71	281.91	108.06	1954.22
<b>Overhead gas from top</b>	m <sup>3</sup> /h	Flowrate gas from top	8016.00	539.93	481.76	-0.80	3228.48
<b>Overhead temperature</b>	°C	Temperature of gas from top	8016.00	80.34	11.44	32.23	113.44
<b>Overhead pressure</b>	barg	Pressure of gas from top	8016.	2.42	0.13	2.05	2.77

Table 5.3. Summary of ‘plant 1’ daily raw data set

<b>Variables</b>	<b>Variable detail</b>	<b>Unit</b>	<b>Count</b>	<b>Mean</b>	<b>Std</b>	<b>Min</b>	<b>Max</b>
<i>Response variables (output variables)</i>							
<b>Water content volume percent</b>	Water content	vol %	334	0.0108	0.0049	0.0040	0.0371
<i>Process variables (input variables)</i>							
<b>feed flowrate</b>	Inlet flowrate	m <sup>3</sup> /h	334	7.82	2.96	3.33	22.87
<b>Feed temperature</b>	Inlet temperature	°C	334	31.91	5.61	19.50	40.46
<b>Column temperature</b>	1	°C	334	130.92	3.81	116.34	140.37
	2	°C	334	120.44	4.42	102.40	129.37
	3	°C	334	102.71	7.25	71.50	118.57
	4	°C	334	84.82	9.58	43.56	101.47
<b>Column pressure</b>	A	barg	334	2.43	0.13	2.28	2.76
	B	barg	334	2.53	0.14	2.38	2.88
<b>Condensate flowrate</b>	Condensate flowrate	m <sup>3</sup> /h	334	6.98	2.77	3.65	21.92
<b>Condensate temperature</b>	Condensate temperature	°C	334	103.42	5.61	83.43	121.32
<b>Reboiler temperature</b>	Reboiler temperature	°C	334	158.36	0.91	156.51	160.32
<b>Steam flowrate</b>	Steam flowrate	kg/h	334	625.71	264.81	219.83	1591.55
<b>Overhead flowrate</b>	Flowrate Gas from Top	m <sup>3</sup> /h	334	539.93	451.53	6.39	2579.16
<b>Overhead temperature</b>	Temperature Gas from Top	°C	334	80.34	10.44	41.02	100.81
<b>Overhead pressure</b>	Pressure Gas from Top	barg	334	2.42	0.13	2.28	2.70

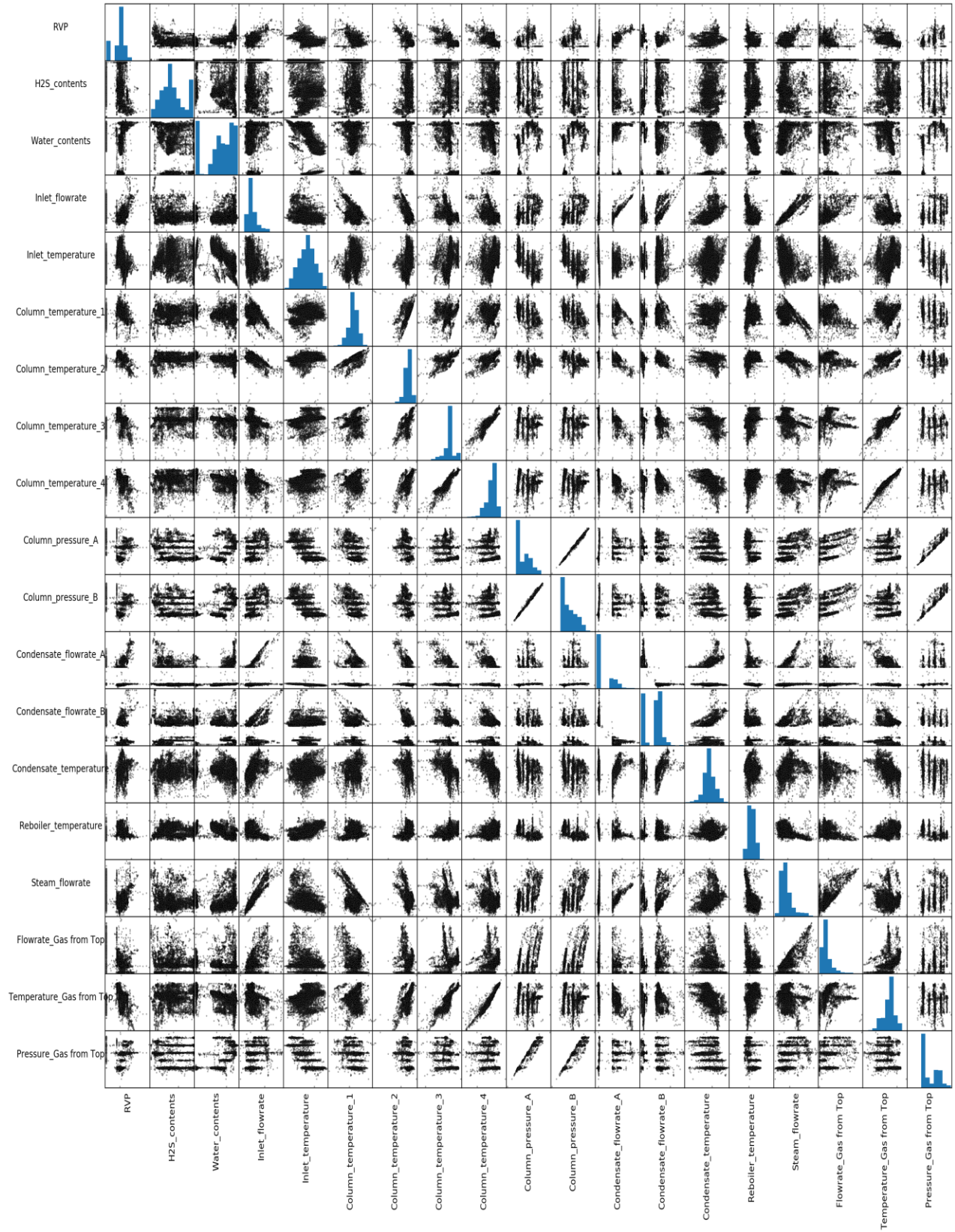


Figure 5.3. Scatter matrix plot of 'plant 1' hourly data set

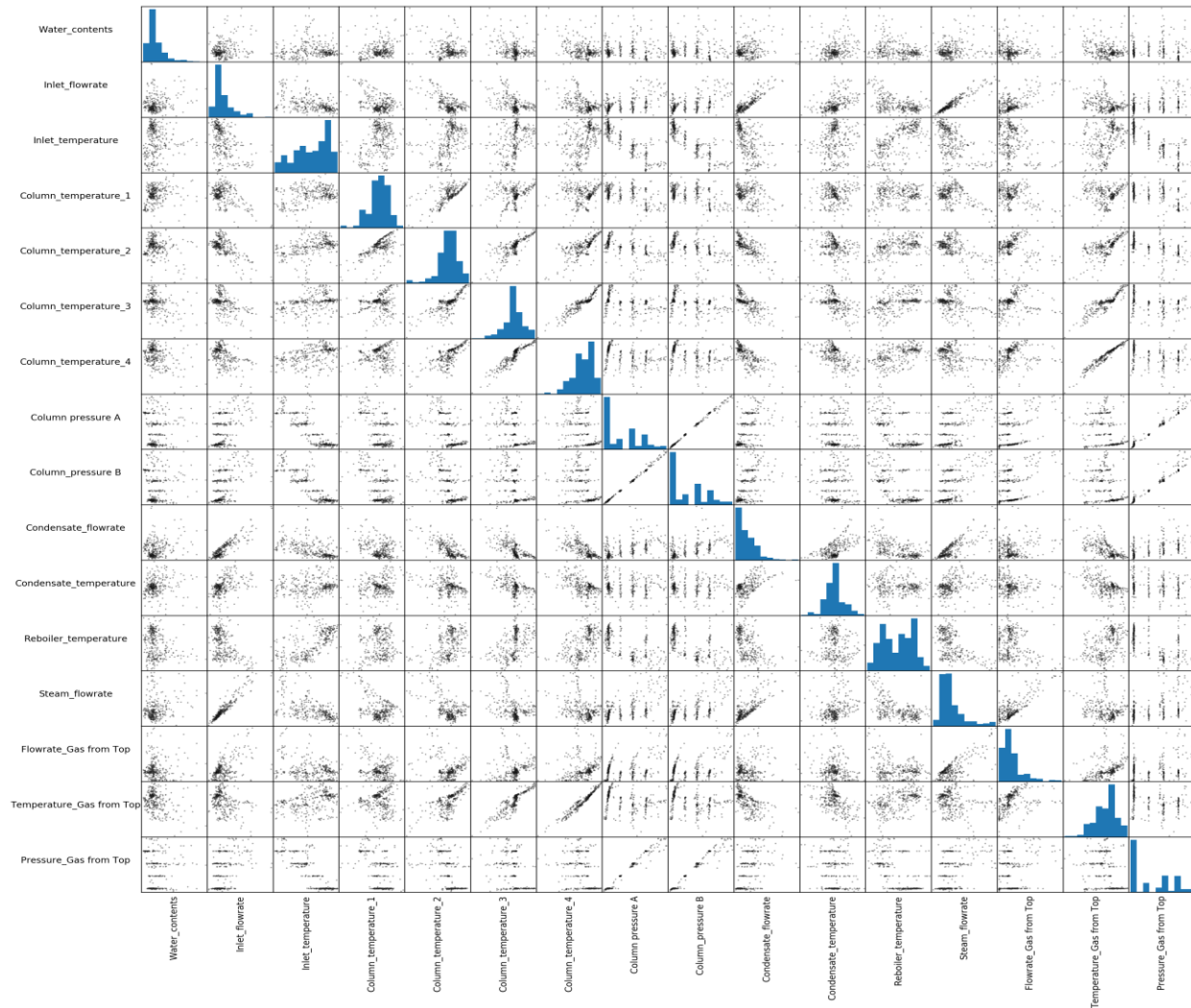


Figure 5.4. Scatter matrix plot of 'plant 1' daily data set

Table 5.4. List of input and output variables of plant 2

Process/feature variables (predictors) - X	Performance/target variables (response) - y
Inlet gas flowrate, temperature, and pressure	RVP and water content, of Stabilized condensate
Reboiler temperature and steam flow rate	
Column temperatures and pressures	
Overhead product flow rate, pressure and temperature (gas top product stream)	
Condensate flowrate, temperature (liquid bottom product stream)	

Table 5.5. Summary of ‘plant 2’ raw data set

Variables		Unit	Count	Mean	Std	Min	Max
<i>Response Variables (output variables)</i>							
<b>RVP</b>	RVP	psi	13617	6.56	1.92	3.76	23.20
<b>Water content</b>	Water content	mg/kg	13617	56.92	11.35	33.00	87.00
<i>Process Variables (input variables)</i>							
<b>Feed flowrate</b>	Inlet Flowrate	m <sup>3</sup> /h	13617	353.66	41.96	1.69	547.14
<b>Feed temperature</b>	Inlet Temperature	°C	13617	118.95	7.27	22.84	159.77
<b>Column temperature</b>	1	°C	13617	198.58	9.91	25.33	231.43
	2	°C	13617	167.76	8.87	26.52	200.28
	3	°C	13617	162.38	8.82	26.40	193.43
	4	°C	13617	144.31	8.43	26.16	175.87
	5	°C	13617	128.07	7.50	25.99	162.29
	6	°C	13617	91.14	5.55	25.78	124.62
	7	°C	13617	83.20	5.56	25.54	123.57
<b>Column pressure drop</b>	Column Pressure Drop	bar	13617	0.08	0.02	0.00	0.21
<b>Column pressure</b>	Column Pressure	barg	13617	9.83	0.44	2.32	11.90
<b>Feed flowrate</b>	Condensate Flowrate	m <sup>3</sup> /h	13617	291.86	34.58	2.17	409.74
<b>Feed temperature</b>	Condensate Temperature	°C	13617	198.03	9.90	24.99	231.20
<b>Reboiler</b>	Temperature (Inlet Shell)	°C	13617	173.69	8.55	25.36	215.46
<b>Reboiler</b>	Temperature (Outlet Shell)	°C	13617	201.48	8.58	25.80	232.22
<b>Feed flowrate</b>	Steam Flowrate	kg/h	13617	16605.48	2808.45	20.06	29976.26
<b>Overhead flowrate</b>	Flowrate of Gas from Top	m <sup>3</sup> /h	13617	11503.92	1588.87	0.37	18761.45
<b>Overhead temperature</b>	Temperature of Gas from Top	°C	13617	71.99	5.36	24.61	123.27
<b>Overhead pressure</b>	Pressure of Gas from Top	barg	13617	9.85	0.41	2.31	11.85



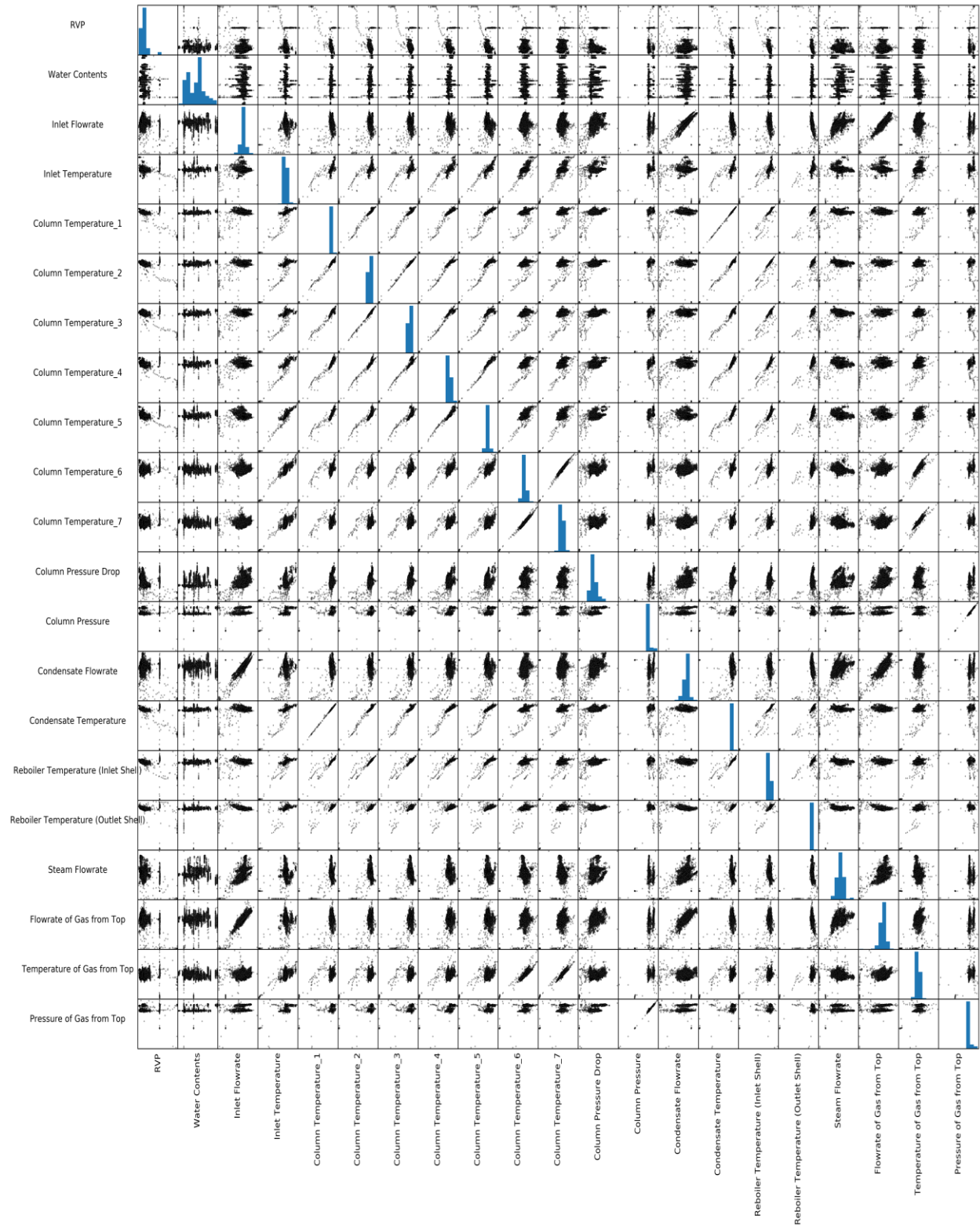


Figure 5.5. Scatter matrix plot of 'plant 2' data set

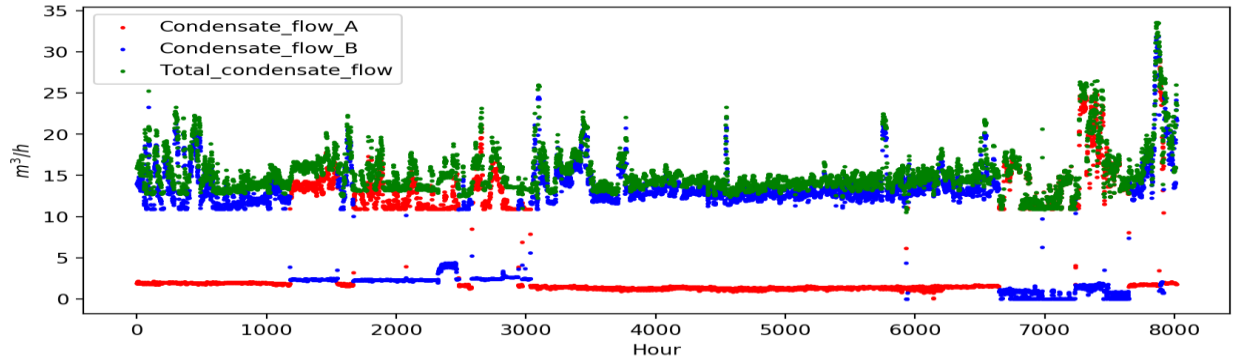


Figure 5.6. Variation in condensate flow rate for pump A and pump B and the total flow that leaves the column

### 5.3.1 Outlier Removal

It is well-known that perfect real data without any outliers is almost nonexistence. Therefore, cleaning data from outliers is very important step in data-driven modelling development. Outliers are observations that do not follow bulk pattern of the data points and are unlikely observation of data [141]. Commonly, outliers are incorrect measurements that can be recognized immediately and removed from the data set. However, sometimes it is challenging to recognize outliers by inspection and visualizing data set. Data set can be composed many input variables defining a high-dimensional feature space. Hence, visualizing them in two dimensions is not possible.

There are different methods that can be used to identify outliers. Some of these methods are based on univariant statistical methods such as simple univariate statistics, like: standard deviation and interquartile range; and the others are based on unsupervised machine learning methods such as one-class classification support vector machine (OCSVM). In this study, several methods were used to remove outliers as follows.

#### 5.3.1.1 Interquartile Range Method (IQR)

IQR is a good statistic tool for measuring the statistical dispersion. The IQR of a set of values can be calculated as the difference between the upper and lower quartiles. For a given even ( $2n$ ) or odd ( $2n+1$ ) set of sample data, the number of values first quartile  $Q_1$  is equal to the median of the  $n$  smallest values. While the third quartile  $Q_3$  will equal to median of the  $n$  largest values. The second quartile  $Q_2$  is the same as the ordinary median [142]. The IQR is calculated as the difference between the third and first quartiles ( $IQR = Q_3 - Q_1$ ). After that Outliers are identified by defining



limits on each feature sample values that are a factor  $k$  of the IQR below  $Q_1$  (first quartiles) or above  $Q_3$ . The common  $k$  factor value is 1.5 which is the value used in this study.

The IQR is a method that depends on statistical measurements, however, there are unsupervised machine learning methods that can be implemented to automatically detect outliers (automatic outlier detection). Generally, outliers are referred as anomalies where the rest of data is normal. In machine learning, anomaly detection problem can be effectively tackled by the advances of the one-class classification. Typically, one-class classification is referred to a subfield of machine learning that focuses on the problem of detecting outlier or anomaly. The goal of one-class classifier is to capture patterns in the underlying training instances, to differentiate between them and potential outliers [143]. One-class classification is subfield of machine learning focused on the problem of detecting outlier or anomaly.

There are a variety of automatic model-based methods for identifying outliers in data. Three methods were considered in this study namely: Local Outlier Factor (LOF), Isolation Forest (IF) and One Class Support Vector Machine Classification (OCSVM). Following is a brief description of these methods.

#### **5.3.1.2 Local Outlier Factor**

It is an unsupervised anomaly detection method that identifies outliers by detecting samples that are located far from the other samples in the feature space. It adopted the idea of nearest neighbours for identifying outlier by assigning a score of how isolated the object/sample with respect surrounding neighbours. In other words, it computes the local density deviation of a given data point with respect to its neighbours. Therefore, the outliers can be identified when its density is much smaller than the densities of its neighbours (where LOF is inversely proportional to the local reachability density, so that  $LOF \gg 1$ ), which means the point is far from dense areas. More explanation on LOF can be found in [144].

#### **5.3.1.3 Isolation Forest (IF)**

Isolation Forest (IF) another unsupervised machine learning tree-based algorithm that performs efficient outlier detection. It works on the principle of recursion. In this algorithm, partitions are recursively generated on the data set by random selection of: 1- feature and 2- a split value between the maximum and minimum values of the selected feature. Outliers can be identified when it needs fewer random partitions to be isolated compared to the normal data points. In other words, the

random partitioning generates noticeably shorter paths for anomalous differentiating them from the normal set of the data. Further details on IF algorithm can be found in [145].

#### **5.3.1.4 One Class Support Vector Machine (OCSVM)**

OCSVM is also an unsupervised machine learning algorithm that is used to identify outliers. It is a modification on the general SVM model that learns a decision boundary (separation hyperplane) and that maximizes the separation between the majority of the data from the origin. Data are projected into higher dimensional space in OCSVM using implicit transformation function that can be defined by kernel. Only small portion of data points are allowed to be located on the other side of the boundary separation hyperplane, where these points are considered to be outliers [146]. The performances of these outlier detection methods on understudy data sets, were evaluated using linear regression. In order to do so, the aforementioned methods were first used to remove outliers from our original data sets (remove outliers). For automated outlier detection, the data sets were scaled based on statistics that are robust to outliers and then the outlier detection methods were performed. Robust scaling removes the median and scales the data according to the quantile range of each feature (the range between the 1<sup>st</sup> quartile and the 3<sup>rd</sup> quartile) [37], [38]. All outlier detection methods were set at their default recommended settings.

Once data sets were cleaned, linear regression models were trained using the original and the cleaned data sets to predict our output variables (response variables e.g., RVP, H<sub>2</sub>S content, and water content) of both plants. The cross-validation techniques were used to evaluate the linear regression models for different outlier detection techniques for both plants data sets. The outlier technique that corresponds to the best cross-validation score (lowest MSE, highest R<sup>2</sup> value) score would be used as the outlier removal method for that plant data set. Table 5.6 to Table 5.8 below list the cross-validation scores of the plant 1 and plant 2 data sets using different outlier detection methods. As it can be seen in this table that IQR method return the best average (for both response variables: RVP and H<sub>2</sub>S content) cross-validation (lowest MSE and highest R<sup>2</sup>) value for the hourly data set of plant 1. In Table 5.7, it can be observed that LOF outperforms other outlier detection methods for plant 1 daily data set. While for plant 2 data set, as it can be seen in Table 5.8. IF method performance was better than other outlier identification methods. Therefore, IQR and LOC would be used to remove outliers from hourly and daily data set of plant 1 respectively. While, IF would be used to remove outliers from original plant 2 data set. Although the performance of IF and OCSVM methods on plant 2 data are close, yet, IF is preferred because it

removes a smaller number of data point compared to OCSVM, as a bigger data size is more preferable in developing machine learning model. The input and output data statistical summaries of the cleaned data sets are reported in Tables (Table 5.9-Table 5.11)for ‘plant 1’ hourly data set, ‘plant 1’ daily data set, and ‘plant 2’ data set respectively. After outliers are removed from all data sets, the cleaned data will be used to construct machine learning model.

Table 5.6. Outlier methods performance comparison for ‘plant 1’ hourly data set

Score meter	R <sup>2</sup>			MSE		
Outlier detection techniques	RVP	H <sub>2</sub> S Content	average	RVP	H <sub>2</sub> S Content	average
Original	0.2557	0.1794	0.2175	0.0179	0.0586	0.0382
IQR	0.7404	0.1933	0.4668	0.0019	0.0592	0.0306
LOF	0.2829	0.1905	0.2367	0.0198	0.0569	0.0383
IF	0.1708	0.1850	0.1779	0.0168	0.0542	0.0355
OCSVM	0.1816	0.2606	0.2211	0.0316	0.0471	0.0394

Table 5.7. Outlier methods performance comparison for ‘plant 1’ daily data set

Score meter	R <sup>2</sup>	MSE
Outlier detection techniques	Water content	
Original	-0.0534	0.0166
IQR	0.1309	0.0332
LOF	0.3240	0.0128
IF	0.1618	0.0160
OCSVM	-0.1159	0.0384

Table 5.8. Outlier methods performance comparison for ‘plant 2’ data set

Score meter	R <sup>2</sup>			MSE		
Outlier detection techniques	RVP	Water content	average	RVP	Water content	average
Original	0.4790	0.2037	0.3414	0.0051	0.0352	0.0201
IQR	0.5855	0.2347	0.4101	0.0148	0.0392	0.0270
LOF	0.5011	0.2264	0.3638	0.0126	0.0344	0.0235
IF	0.6283	0.2525	0.4404	0.0043	0.0285	0.0164
OCSVM	0.6435	0.2201	0.4318	0.0105	0.0232	0.0169

Table 5.9. Summary of ‘plant 1’ cleaned hourly data set

		Unit	Count	Mean	Std	Min	Max
<i>Response Variables (output variables)</i>							
	RVP	psi	6547	2.60	0.56	0.80	7.25
	H <sub>2</sub> S contents	ppm	6547	10.15	5.40	0.08	20.00
<i>Process Variables (input variables)</i>							
<b>Feed flowrate</b>	Inlet flowrate	m <sup>3</sup> /h	6547	7.42	2.87	0.14	26.52
<b>Feed temperature</b>	Inlet temperature	°C	6547	33.31	6.00	13.76	48.41
<b>Column temperature</b>	1	°C	6547	131.43	4.23	108.63	147.82
	2	°C	6547	121.06	5.26	62.71	134.77
	3	°C	6547	103.60	8.71	50.33	123.22
	4	°C	6547	85.89	11.26	29.00	115.43
<b>Column pressure</b>	A	barg	6547	2.40	0.12	2.07	2.98
	B	barg	6547	2.50	0.12	2.14	3.06
<b>Condensate flowrate</b>	Condensate flowrate	m <sup>3</sup> /h	6547	15.19	2.77	10.54	33.58
<b>Condensate temperature</b>	Condensate temperature	°C	6547	103.34	6.79	74.91	126.05
<b>Reboiler temperature</b>	Reboiler temperature	°C	6547	158.53	0.95	152.46	164.14
<b>Steam flowrate</b>	Steam flowrate	Kg/h	6547	584.20	233.38	108.06	1954.22
<b>Overhead flowrate</b>	Flowrate Gas from Top	m <sup>3</sup> /h	6547	485.64	390.07	0.88	3228.48
<b>Overhead temperature</b>	Temperature Gas from Top	°C	6547	81.34	11.98	32.23	113.44
<b>Overhead pressure</b>	Pressure Gas from Top	barg	6547	2.39	0.12	2.05	2.77

Table 5.10. Summary of ‘plant 1’ cleaned daily data set

		Unit	Count	Mean	Std	Min	Max
<i>Response Variables (output variables)</i>							
<b>Water content</b>	Water content		300	0.0107	0.0048	0.0040	0.0371
<i>Process Variables (input variables)</i>							
<b>Feed flowrate</b>	Inlet flowrate	m <sup>3</sup> /h	300	7.75	2.73	4.00	16.93
<b>Feed temperature</b>	Inlet temperature	°C	300	31.94	5.69	19.78	40.46
<b>Column temperature</b>	1	°C	300	131.17	3.37	121.29	140.37
	2	°C	300	120.55	4.17	102.40	128.55
	3	°C	300	102.48	7.29	71.50	118.57
	4	°C	300	84.53	9.75	43.56	101.47
<b>Column pressure</b>	A	barg	300	2.43	0.13	2.28	2.76
	B	barg	300	2.53	0.13	2.38	2.88
<b>Condensate flowrate</b>	Condensate flowrate	m <sup>3</sup> /h	300	6.97	2.57	3.65	17.39
<b>Condensate temperature</b>	Condensate temperature	°C	300	103.91	5.21	88.55	121.32
<b>Reboiler temperature</b>	Reboiler temperature	°C	300	158.34	0.91	156.51	160.32
<b>Steam flowrate</b>	Steam flowrate	kg/h	300	615.70	247.21	270.03	1591.55
<b>overhead flowrate</b>	Flowrate Gas from Top	m <sup>3</sup> /h	300	516.91	432.58	6.39	2579.16
<b>overhead temperature</b>	Temperature Gas from Top	°C	300	79.91	10.52	41.02	100.81
<b>overhead pressure</b>	Pressure Gas from Top	barg	300	2.41	0.13	2.28	2.70

Table 5.11. Summary of ‘plant 2’ cleaned data set

		Unit	Count	Mean	Std	Min	Max
<b>RVP</b>	RVP	psi	12170	6.30	1.12	4.01	14.43
<b>Water content</b>	Water content	mg/kg	12170	56.22	10.56	33.00	87.00
<b>Feed flowrate</b>	Inlet Flowrate	m <sup>3</sup> /h	12170	358.57	28.65	235.59	497.99
<b>Feed temperature</b>	Inlet Temperature	°C	12170	118.26	3.47	94.27	129.92
<b>Column temperature</b>	1	°C	12170	198.18	3.45	187.22	208.03
	2	°C	12170	167.35	4.56	155.94	179.37
	3	°C	12170	161.95	4.69	149.16	175.57
	4	°C	12170	143.54	4.69	127.99	157.40
	5	°C	12170	127.30	4.05	114.33	141.69
	6	°C	12170	90.63	3.86	78.31	104.39
	7	°C	12170	82.71	4.27	70.80	100.17
<b>Column pressure drop</b>	Column Pressure Drop	bar	12170	0.08	0.02	0.01	0.13
<b>Column pressure</b>	Column Pressure	barg	12170	9.75	0.24	9.30	11.04
<b>Condensate flowrate</b>	Condensate Flowrate	m <sup>3</sup> /h	12170	294.05	24.88	126.24	391.87
<b>Condensate temperature</b>	Condensate Temperature	°C	12170	197.64	3.49	186.88	207.23
<b>Reboiler temperature</b>	(Inlet Shell)	°C	12170	173.24	4.27	161.05	183.92
	(Outlet Shell)	°C	12170	201.31	3.08	190.74	211.91
<b>Steam flowrate</b>	Steam Flowrate	kg/h	12170	16981.09	2413.30	10541.69	27370.59
<b>Overhead flowrate</b>	Flowrate of Gas from Top	m <sup>3</sup> /h	12170	11618.97	1149.28	7615.07	16819.40
<b>Overhead temperature</b>	Temperature of Gas from Top	°C	12170	71.55	4.46	59.55	92.19
<b>Overhead pressure</b>	Pressure of Gas from Top	barg	12170	9.77	0.22	7.64	10.99

### 5.3.2 Feature Dependency and Selection

After outliers were removed from data set, F-test were performed on the cleaned data set for each response variable. F-test is a univariate statistical method that measure the degree linear

dependency between two random variables [38]. Therefore, F-test was used to explore the linear correlation between all features and each output variable for all given data sets. Additionally, F-test value was used to calculate the P-value which is probability that corresponds to the accepting of the null hypothesis. Therefore, the lower the P-value the stronger the argument of rejecting the null hypothesis (i.e., no linear correlation) and the more possibly that there is a correlation between the two corresponding variables (e.g., input and output variable). The larger the F-value, the stronger is the linear dependency. F-test and P-value statistics are usually performed to see whether there is a possible relationship between process and response variables. F-test values for each feature with respect of each response variables are shown in Figure 5.7 and Figure 5.8 for plant 1 and plant 2, respectively. Table 5.12 and Table 5.13 report the P-value of each feature with respect each response variable for plant 1 and plant 2 data set, respectively. As it can be demonstrated from these figures, that most of the input variables have strong correlation with the certain response variables, and some have weak correlation. In general, most P-value results imply that the null hypothesis is rejected, and possible correlations do exist between most of inputs and output variables.

A further investigation on the effect of the features space (feature selection) using F-test value for prediction accuracy for each response variable was performed. More specifically, a linear regression was constructed to see how the model accuracy would be enhanced if some of the input (features) with low F-test value were removed from the training data set (i.e., subset of input variables will be used to train the model). Figure 5.9 and Figure 5.11 show the cross-validation MSE as a function of the number of features that kept to train (develop) the linear model using plant 1 and plant 2 data sets, respectively. Figure 5.10 and Figure 5.12 are generated to see the same effect but with  $R^2$  as a cross-validation score. As it can be seen from these figures that best cross-validation score (lowest error and highest  $R^2$  value) is always achieved for all response variable for both plants when all features were used for training.

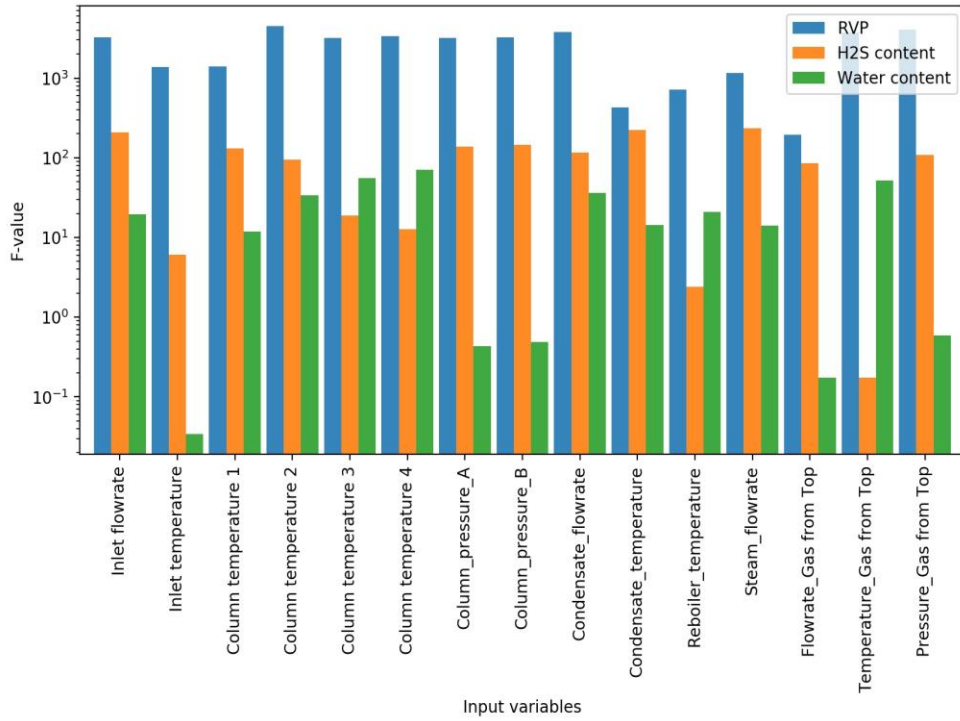


Figure 5.7. F-test values for plant 1 features vs target variables

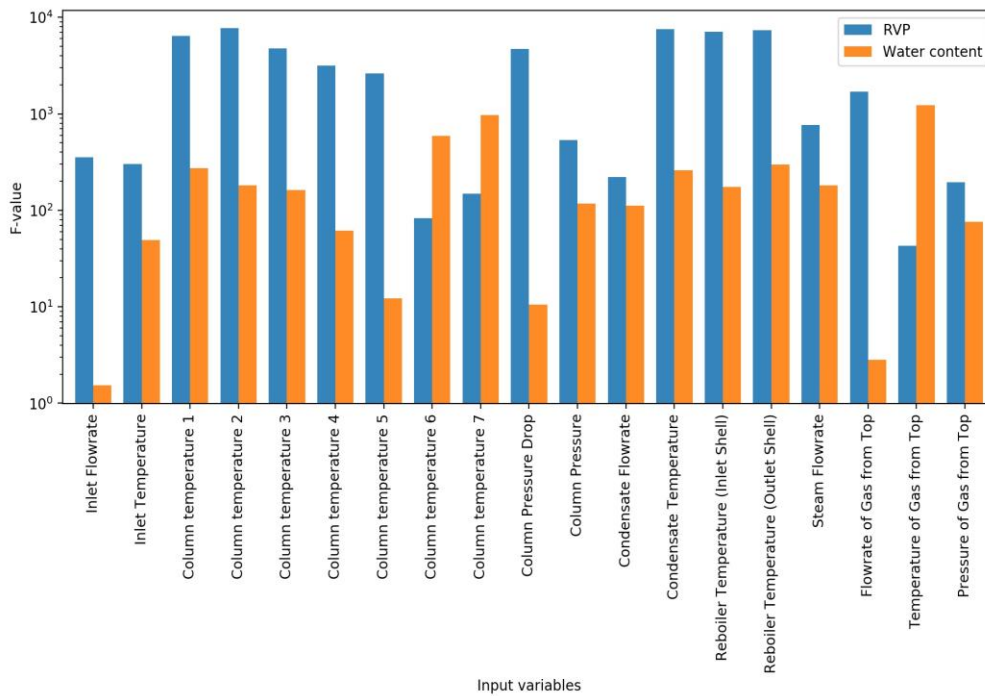


Figure 5.8. F-test values for plant 2 features vs target variables



Table 5.12. P-values of plant 1 features against response variables

	<b>RVP</b>	<b>H<sub>2</sub>S content</b>	<b>Water content</b>
<b>Inlet flowrate</b>	0.00000000	0.00000000	0.00001374
<b>Inlet temperature</b>	0.00000000	0.01338270	0.85276167
<b>Column temperature 1</b>	0.00000000	0.00000000	0.00063676
<b>Column temperature 2</b>	0.00000000	0.00000000	0.00000002
<b>Column temperature 3</b>	0.00000000	0.00001541	0.00000000
<b>Column temperature 4</b>	0.00000000	0.00038679	0.00000000
<b>Column pressure A</b>	0.00000000	0.00000000	0.51039629
<b>Column pressure B</b>	0.00000000	0.00000000	0.48531952
<b>Condensate flowrate</b>	0.00000000	0.00000000	0.00000001
<b>Condensate temperature</b>	0.00000000	0.00000000	0.00019662
<b>Reboiler temperature</b>	0.00000000	0.12102039	0.00000772
<b>Steam flowrate</b>	0.00000000	0.00000000	0.00022153
<b>Flowrate Gas from Top</b>	0.00000000	0.00000000	0.67657968
<b>Temperature Gas from Top</b>	0.00000000	0.67544459	0.00000000
<b>Pressure Gas from Top</b>	0.00000000	0.00000000	0.44139847

Table 5.13. P-values of plant 2 features against response variables

	<b>RVP</b>	<b>Water content</b>
<b>Inlet Flowrate</b>	0.00000000	0.2169081
<b>Inlet Temperature</b>	0.00000000	0.00000000
<b>Column temperature 1</b>	0.00000000	0.00000000
<b>Column temperature 2</b>	0.00000000	0.00000000
<b>Column temperature 3</b>	0.00000000	0.00000000
<b>Column temperature 4</b>	0.00000000	0.00000000
<b>Column temperature 5</b>	0.00000000	0.0004812
<b>Column temperature 6</b>	0.00000000	0.00000000
<b>Column temperature 7</b>	0.00000000	0.00000000
<b>Column Pressure Drop</b>	0.00000000	0.0011503
<b>Column Pressure</b>	0.00000000	0.00000000
<b>Condensate Flowrate</b>	0.00000000	0.00000000
<b>Condensate Temperature</b>	0.00000000	0.00000000
<b>Reboiler Temperature (Inlet Shell)</b>	0.00000000	0.00000000
<b>Reboiler Temperature (Outlet Shell)</b>	0.00000000	0.00000000
<b>Steam Flowrate</b>	0.00000000	0.00000000
<b>Flowrate of Gas from Top</b>	0.00000000	0.0926379
<b>Temperature of Gas from Top</b>	0.00000000	0.00000000
<b>Pressure of Gas from Top</b>	0.00000000	0.00000000

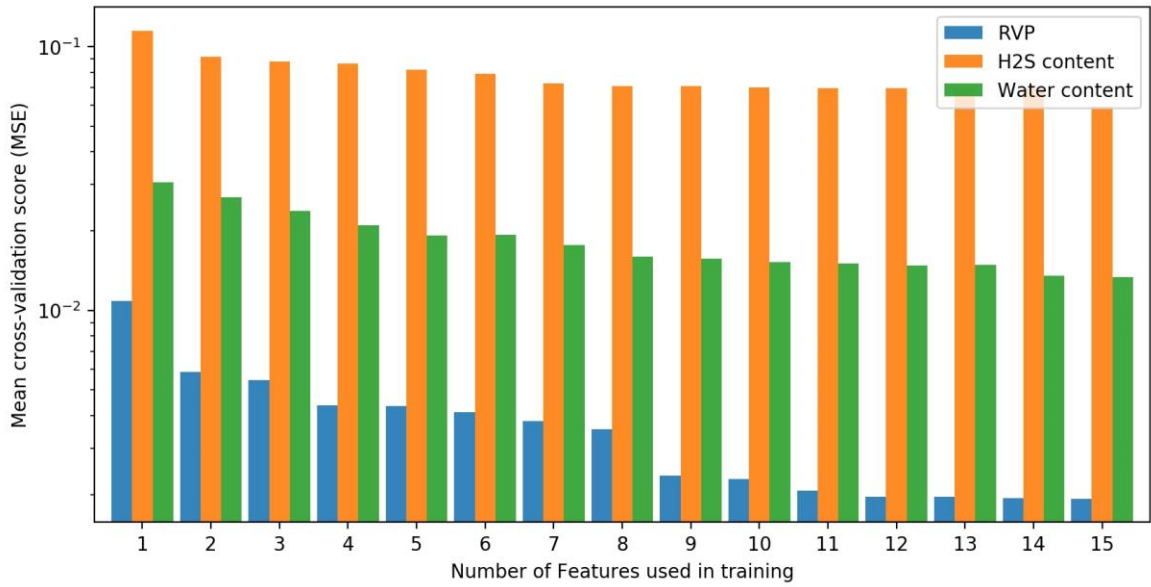


Figure 5.9. Effect of feature selection on plant 1 response variables prediction error using F-test value

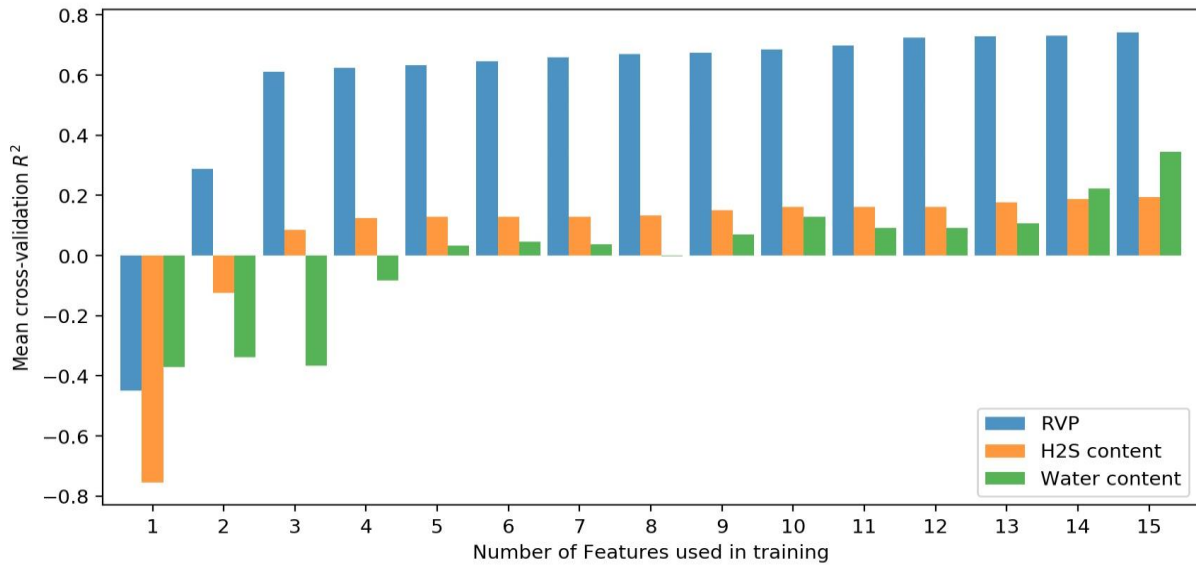


Figure 5.10. Effect of feature selection on plant 1 response variables prediction accuracy ( $R^2$ ) using F-test value

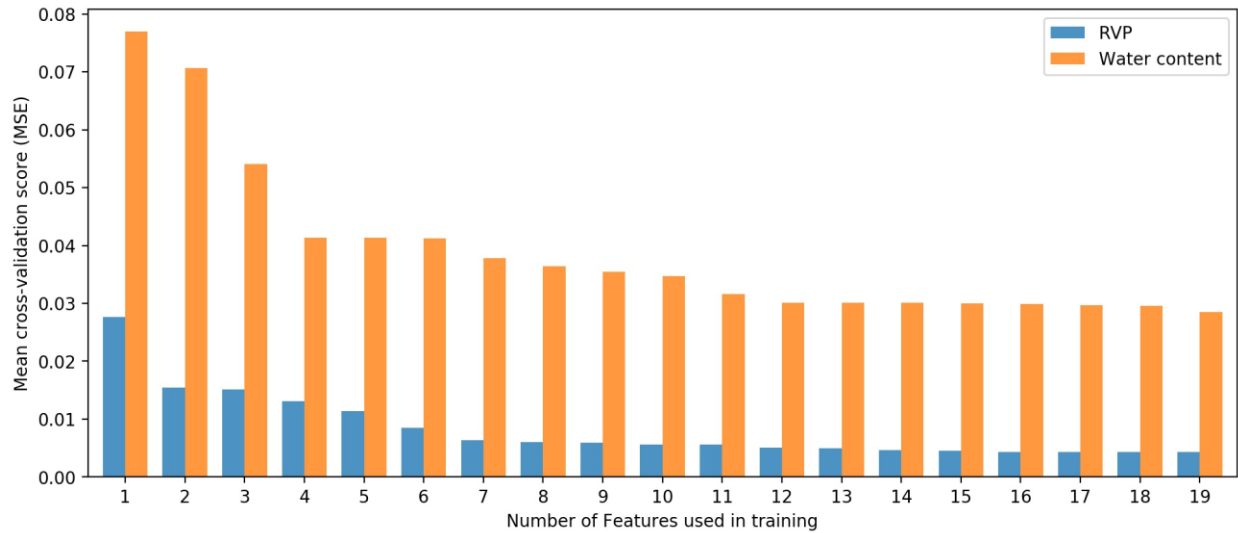


Figure 5.11. Effect of feature selection on plant 2 response variables prediction error using F-test value

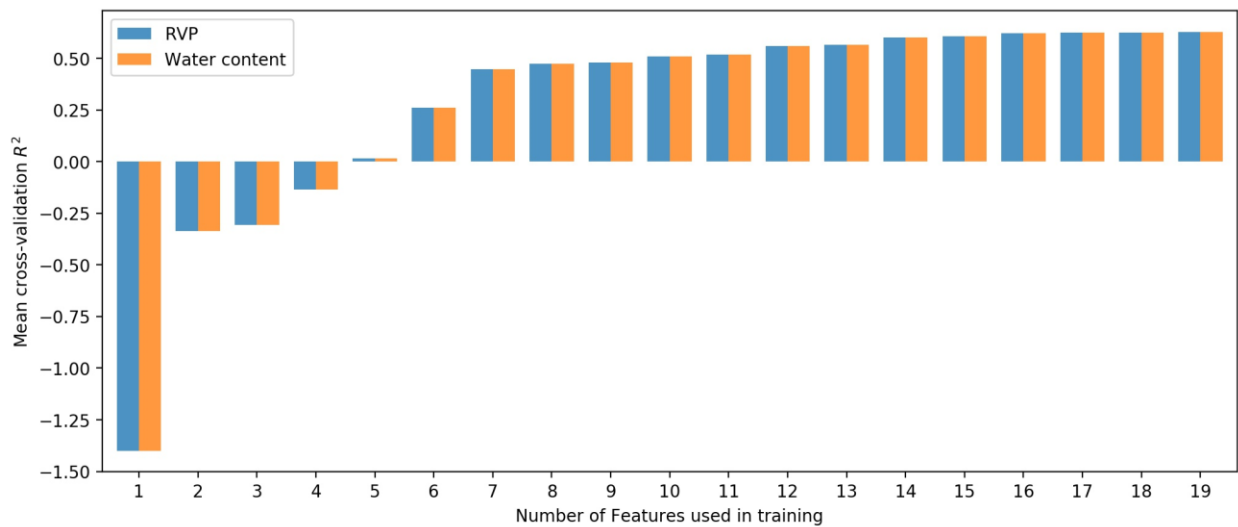


Figure 5.12. Effect of feature selection on plant 2 response variables prediction accuracy (R<sup>2</sup>) using F-test value

The F-test statistic is only applicable to estimate degree of linear correlation between two random variables. In order to explore any kind of statistical dependency (e.g., nonlinear correlation), the Mutual Information (MI) methods were implemented. MI measures the amount of information about one random variable that can be obtained by observing the other random variable. The Mutual information (MI) value is a non-negative value that quantifies the correlation (dependency) between two random variables. It has minimum value of zero when the random variables are totally independent. The larger the value of MI the stronger the dependency between the two variables.

Entropy of random variable is the main concept behind MI [147]. MI is a nonparametric methods, that is computed based on entropy estimation from k-nearest neighbours distances as explained in [148] and [149] and originally proposed by [147]. More technical details on how MI is calculated can be found in [149]. MI score for plant 1 and plant 2 data features with respect all response variables are displayed in Figure 5.13 and Figure 5.14. The figures show that all response (output) variables have a strong dependency on most of the input features. More specifically, all values are greater than zero and most of them were even than 0.2.

To see the effect of feature selection by removing some of feature/ input variables from the training data set that has low MI score, and a similar investigation that was done for F-test. Figure 5.15 - Figure 5.18 illustrate how the mean cross-validation scores are varying with the number of features kept to train the linear models that predict plant 1 and plant 2 target variables. These figures show that the best scores were always achieved when all features were used to train the models. Therefore, it was decided not to remove any feature from data sets during training process of developing machine learning models in the next section. Even though, the prediction of the target variables has weak degree of dependency on only some features, the number of features in this study are relatively not large. As a result of that, removing some of them would not either enhance the training process significantly (increase the speed) or improve the prediction accuracy notably. The feature investigation was performed to check whether selecting subset of features for the training process would enhance the prediction accuracy, additionally, to establish a general framework for developing data-driven surrogate models. It should be also mentioned that when the number of features as well as the training data is large, dimensionality reduction methods can be applied to reduce the data dimension such as Principal Component Analysis (PCA) and Autoencoder (ANN). In the machine learning field, this is referred as feature extraction process which also typical part of data pre-processing step specially when the data set is enormous and feature extraction can significantly speed up the training process.

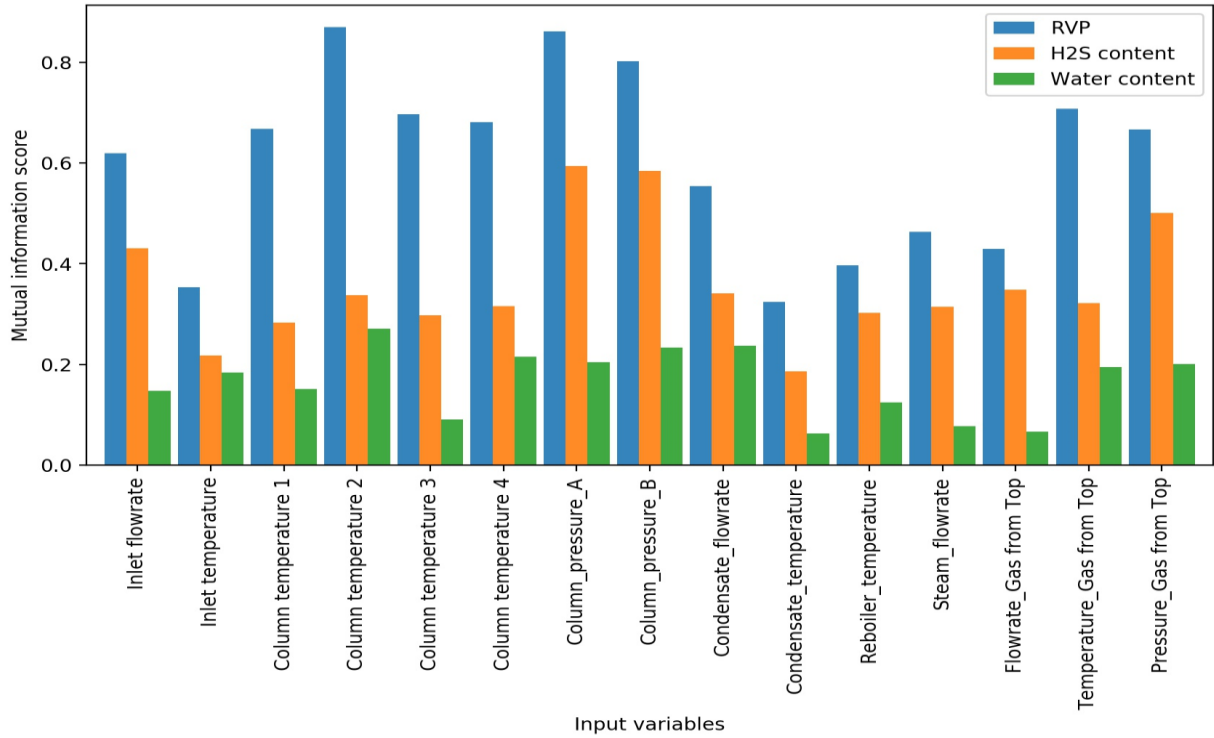


Figure 5.13. Mutual information score values for plant 1 features vs target variables

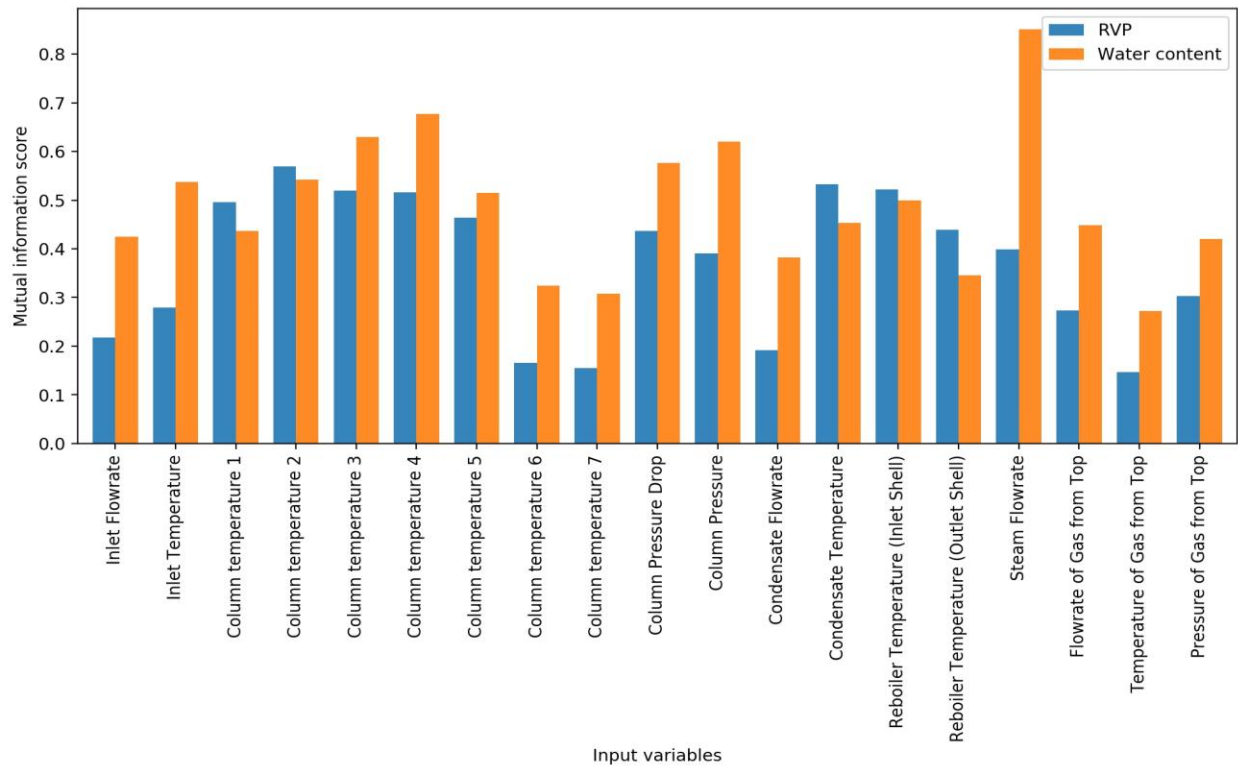


Figure 5.14. Mutual information score values for plant 2 features vs target variables

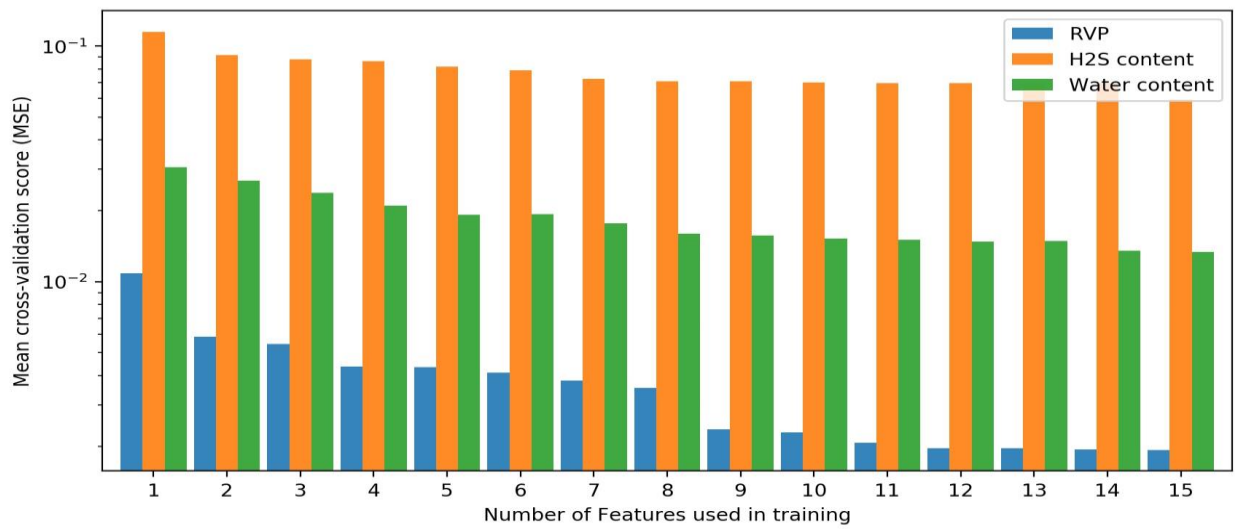


Figure 5.15. Effect of feature selection on plant 1 response variables prediction error using using MI score

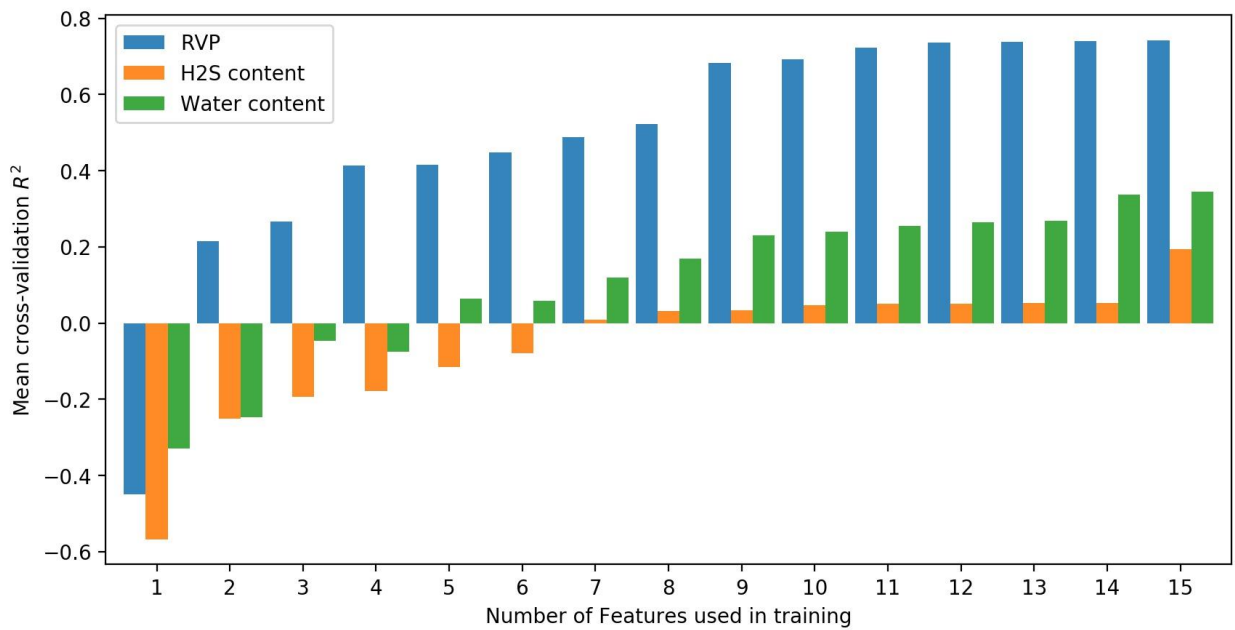


Figure 5.16. Effect of feature selection on plant 1 response variables prediction accuracy ( $R^2$ ) using using MI score

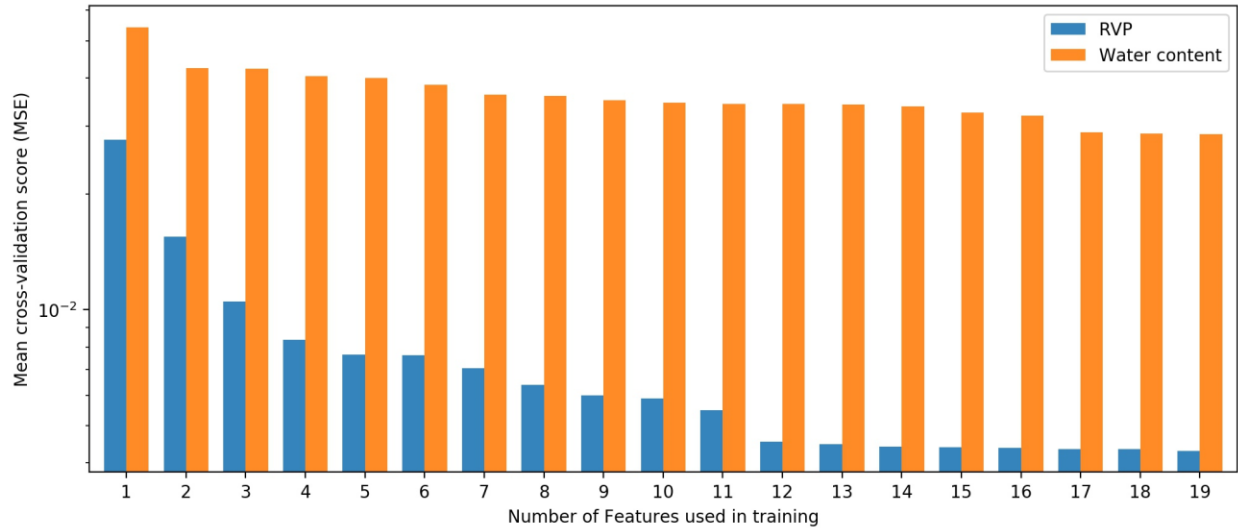


Figure 5.17. Effect of feature selection on plant 2 response variables prediction error using using MI score

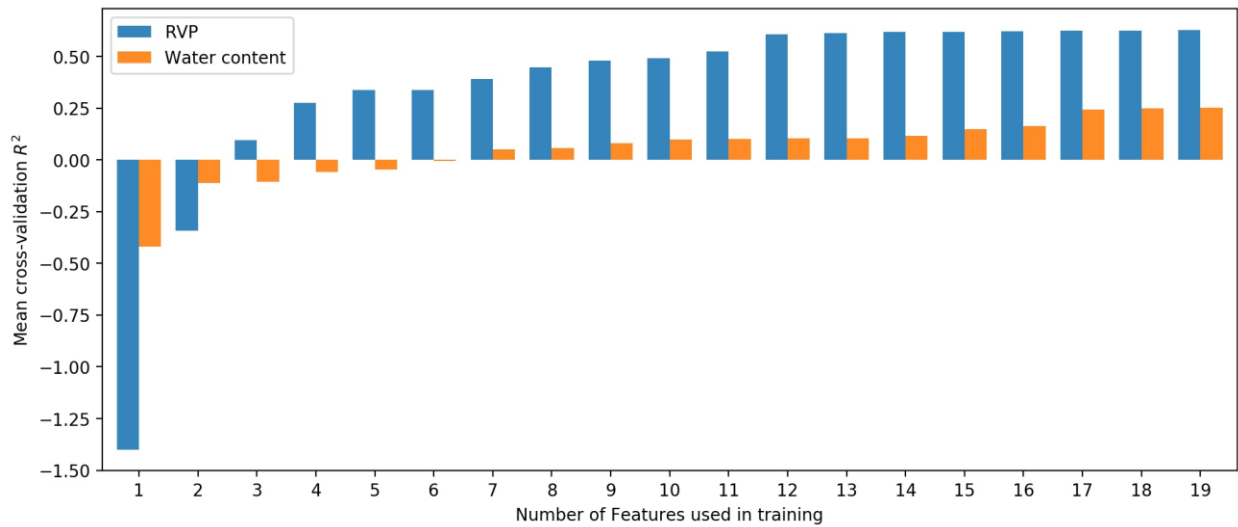


Figure 5.18. Effect of feature selection on plant 2 response variables prediction accuracy ( $R^2$ ) using MI score

### 5.3.3 Data Scaling

After analysing the dependency of response variables on the input variables, all set of data were scaled. All the input (operating variables)/ output data have been normalized to the range of [0 , 1] using equation (5.1). Thus, input and output variables will have the same order of magnitude and same significance, since feature with a higher value range tend to dominate when calculating distances in the machine learning training process. Moreover, neural network optimization algorithm (gradient descent) converge much faster with feature scaling than without it [40], [150].

$$X_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (5.1)$$

Where  $X_i$  denotes the scaled (normalized), value of input/output data.,  $x_i$  is actual value of input/output data  $x_{max}$  and  $x_{min}$  represent minimum observation value of dataset and maximum observation value of dataset, respectively. Target variables were scaled in this study as well because large spread of values of target variables may result in large error gradient values causing weight values to change dramatically, and hence, the learning process would become unstable for gradient based machine learning algorithm (e.g., ANN).

At this point, data sets were processed, and they were ready for the next step which is building the machine learning models. Data-driven machine learning models can easily examine and be used to predict output variables at different operating mode. As it is known that testing new operating mode on the real plant directly is very challenging and risky. Using the proposed models will allow plant operator and decision maker to quickly test new scenarios that may include changing the product quality or estimating how the product will vary under unexpected disturbance. Additionally, data mining models will help in exploring the effect of changing one or two inputs on the performance of outputs, while other inputs variability are negligible on the performance. The data-driven models of chemical process can act as a preliminary simulation tool that can mimic the real plant behaviour. Commercial simulator software is necessary in operating the plant because they can offer a comprehensive and accurate platform for plant simulation. However, due to different hidden factor such as plant age, corrosion, scaling and energy loss to the surrounding in which most of the commercial software packages are using rules of thumbs to model these hidden phenomena; hence, their simulation results might not be very accurate, and modelling these hidden factors accurately need a significant effort. Here comes the advantage of the data-driven models, since they model the process plant under the current situation without the need to understand what is going on inside, at the same time, they can provide accurate prediction results. Moreover, there is always a room to improve the performance of data-driven models by improving the quality along with the quantity of data. Data-driven models can not fully replace the first principle modelling approach, but both can work together to enhance and optimize the plant operations.



## 5.4 Machine Learning Models Developments

This section describes the construction of different machine learning models for the two condensate stabilizer plants. The hyperparameter model parameters and model validation are also included. At the end of this section, the performance these models are compared. All following models were developed in Python.

### 5.4.1 Linear Regression Models

Linear regression models (such as Ridge and Lasso linear regression that described before) were developed to predict plant 1 and plant 2 target variables. As mentioned before, Ridge and Lasso linear regression models are some of the simple methods used to reduce model complexity and prevent over-fitting which may result from simple linear regression models. Three Ridge and Lasso regression models were constructed to predict the target variables of the first plant (RVP, H<sub>2</sub>S content and water content, while for the second plant two Ridge and two Lasso models were developed to estimate the output variables (RVP and Water content). The penalty tuning parameter  $\lambda$  was obtained using cross-validation grid search. The MSE score was used as the scoring measure for the best fit  $\lambda$ . Validation curves for all developed models of each output variables for both plants are shown in Figure 5.19 and Figure 5.21. Five cross-validation folds were used in this study. Table 5.14 shows the optimal shrinkage penalty parameter  $\lambda$ , the mean cross-validation MSE and R<sup>2</sup> for each model output for both plants.

As it can be seen from validation figures that MSE is almost constant for low values of  $\lambda$ , until a certain level of  $\lambda$  where both training and validation errors start to increase. At this level of  $\lambda$ , the greater the value of  $\lambda$ , the more is the bias and variance error (see section 2.4.1 for more details on bias and variance errors). This is because when  $\lambda$  increases more penalty on the linear regression coefficients is enforced leading them to be smaller and hence, failed to predict the response variables accurately (i.e., become bias). On other words, bias error and variance error are high when  $\lambda$  is high. In order to look closely to the variation of MSE cross-validation and training scores at low level of  $\lambda$ , and therefore Figure 5.20 is generated. Although it is obvious when  $\lambda$  is decreasing, the MSE is getting better until a certain level, where after that both training and validation score are appeared to be constant. If we look closely using Figure 5.20, one can see that there is some sort of optimal  $\lambda$  value where after this value, decreasing  $\lambda$  has very slightly opposite

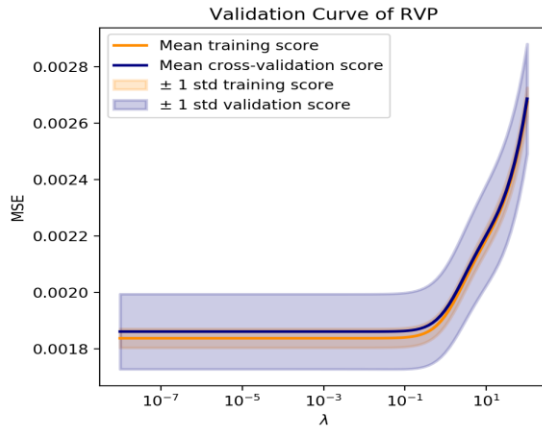
effect on the cross-validation score (increasing the error). This is the point where optimal  $\lambda$  was obtained by the search algorithm.

From Table 5.14, generally the  $R^2$  values are small specially for water content and  $H_2S$  prediction which implies that the relationships between these inputs and output variables are unlikely to be linear. Hence, in the next section more detailed nonlinear machine learning models were developed.

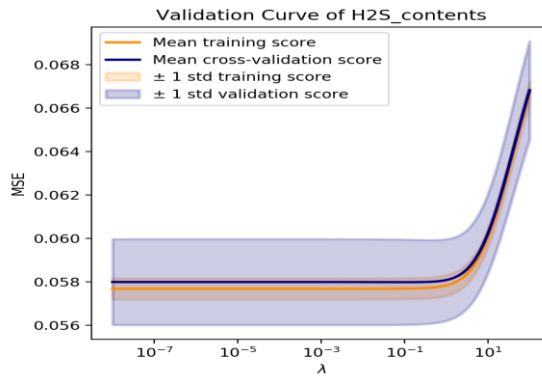
Table 5.14. Linear regression models cross-validation evaluation

	Response variables	Ridge linear regression			Lasso linear regression		
		Mean cross-validation $R^2$	Mean cross-validation MSE	$\lambda$	Mean cross-validation MSE	Mean cross-validation $R^2$	$\lambda$
<b>Plant 1</b>	RVP	0.7496	0.0019	1.150e-2	0.7496	0.0019	5.214e-7
	$H_2S$ content	0.2097	0.0580	9.326e-2	0.2096	0.0580	2.79e-5
	Water content	0.3305	0.0133	2.477e-3	0.3320	0.0128	1.427e-5
<b>Plant 2</b>	RVP	0.6304	0.0043	4.642e-2	0.6304	0.0043	2.565e-6
	Water content	0.2565	0.0284	1.177e-1	0.2565	0.0284	6.734e-6

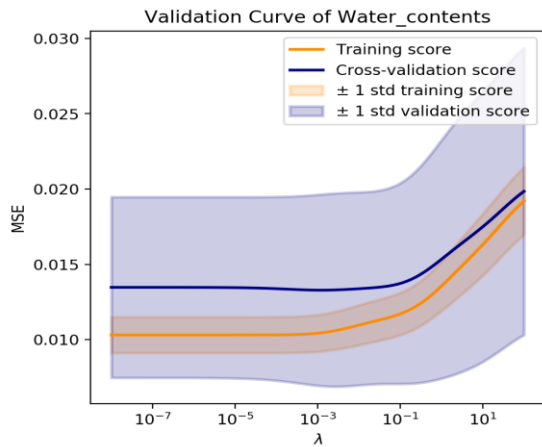
### Ridge regression validation curves



(a)

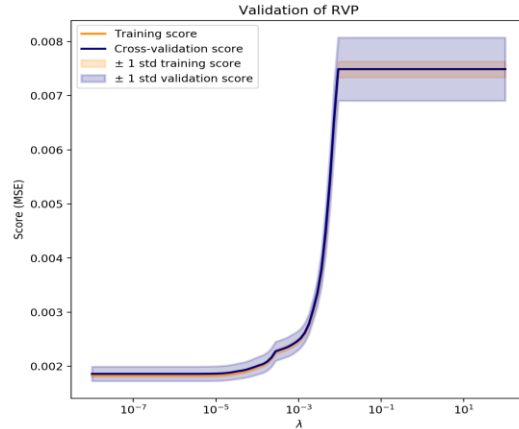


(b)

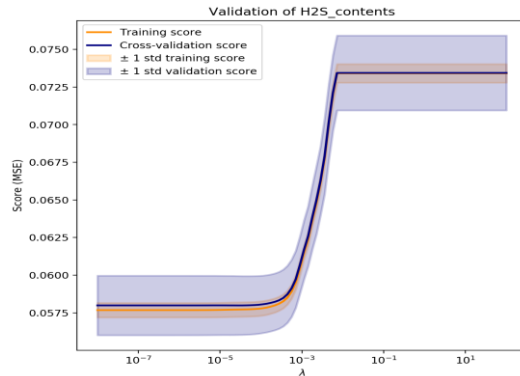


(c)

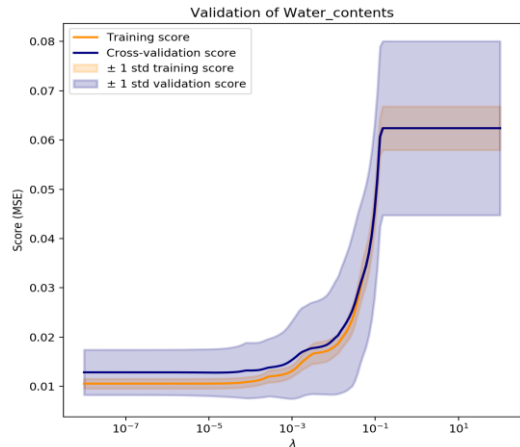
### Lasso regression validation curves



(d)



(e)



(f)

Figure 5.19. Validation curves for linear regression models of ‘plant 1’ where (a), (b) and (c) denote Ridge models for RVP, H<sub>2</sub>S content and water content respectively, while (d), (e) and (f) represent Lasso models for RVP, H<sub>2</sub>S content and wate content respectively

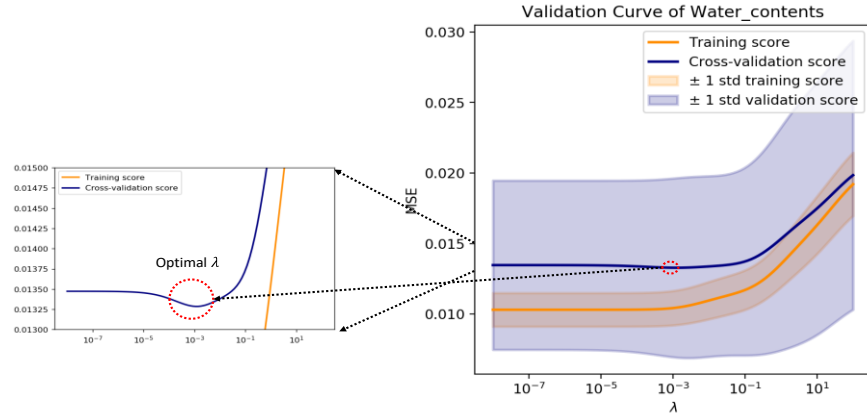


Figure 5.20. Zoomed validation curve of water content for plant 1 condensate

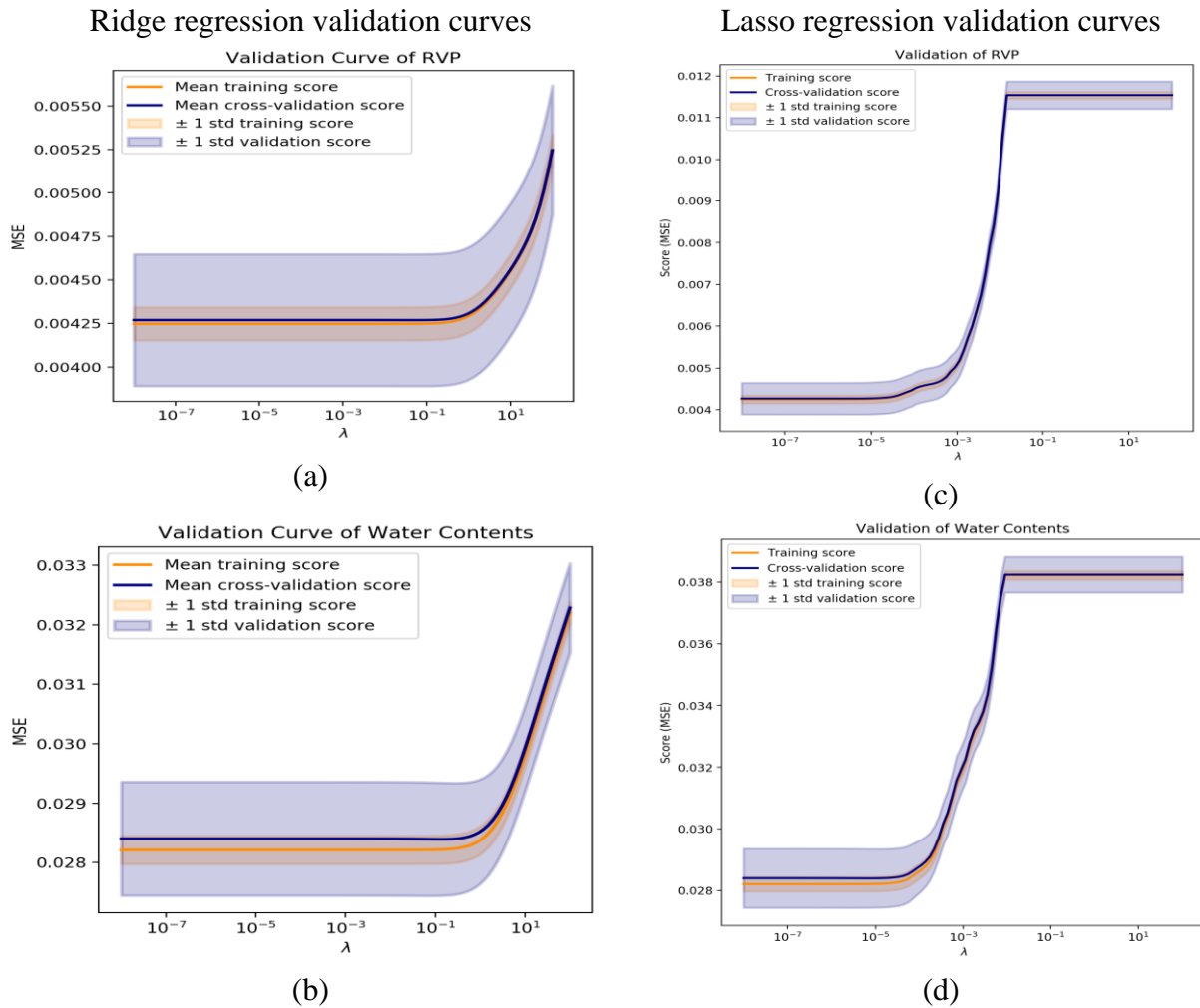


Figure 5.21. Validation curves for linear regression models of 'plant 2' where (a) and (b) denote Ridge models for RVP, water content respectively, while (c), and (d) represent Lasso models for RVP and water content respectively

## 5.4.2 Development of Detailed Models

In this advanced machine learning models such as SVM regression, and artificial neural network (ANN) were developed to predict the performance variables. To develop these models, inlet gas flowrate, inlet temperature, column temperature, column pressure, condensate flowrate, condensate temperature, reboiler temperature and steam flowrate are applied as model input data; whereas RVP, H<sub>2</sub>S and water contents are used as model outputs. The performance of these models was assessed either using cross-validation score or a validation set approach (test set approach) to calculate the mean squared error (MSE) and R<sup>2</sup>.

### 5.4.2.1 SVM regression model development

SVM is one of the most sophisticated and versatile supervised machine learning [12]. SVM can present in many different configurations depending on kernel function used to generate transform function that implicitly mapping inputs into high-dimensional feature spaces. Applying kernel tricks enable SVM to learn nonlinear functions. As it avoids the explicit mapping into high-dimensional space, so necessary computations are made directly in the input space. More specifically, SVM with kernel function can operate in a high-dimensional (implicit feature space) without ever calculating the coordinates of the data in that space, but instead, calculate the inner products between all pairs of data in the feature space (input space) via kernel transformation [151],[152]. Generally, there are several kernels that are used in SVM such as linear, polynomial, Radial Basis Function (RBF) (section 2.4.5). RBF is one of the most popular Kernel that has been widely employed [47]. In this study, RBF was used as the kernel for SVM because it is practical and relatively easy to tune.

In SVM models understudy, two key parameters (hyperparameter) were needed to be tuned (optimized), specifically, regularization parameter ( $C$ ) and kernel function parameter ( $\gamma$ ).  $C$ -value determines the trade-off between minimizing the inaccurate prediction of the estimated function (fitting error minimization) of training instances and simplifying the estimated function (smoothness of the estimated function) [153].  $C$ -value has the opposite effect as  $\lambda$  in Lasso and Ridge linear regression, hence, a higher  $C$ -value means higher prediction training accuracy but higher variance as well. Gamma ( $\gamma$ ) describes how much influence a single training instance has. Therefore, larger gamma means larger estimated function complexity, and smaller gamma implies that model is too constrained and cannot capture the complexity or shape of the data [153].

An evaluation algorithm was used to find the optimal values of SVM regression hyperparameters SVM model. Figure 5.24 shows schematic representation of the implementation of GA with SVM regression to find its optimal hyperparameters. More details on the GA were presented in former section (2.4.5.1). The algorithm starts by generating an initial population that made of individuals (solution). A uniform distribution was used to randomly generate individuals. Each solution is composed of a C-value and gamma parameter. After that, the dual SVM optimization problem is solved for every Individuals under cross-validation approach. Therefore, SVM regression is trained for every individual and every fold. In this study the number of folds of the cross-validation calculation is set to be 5. Then, individual undergoes selection process, where they are evaluated based on the mean cross-validation MSE, and those with the lowest error are selected. In other words, solutions with the best cross-validation score are selected and others are eliminated. These selected individuals are called parents and the number of parents is predetermined value that defined by the user. Parents afterward engage in mating process to produce offspring individuals. Mating processes involve crossover followed by mutation. In crossover, C-value and gamma parameters from the different survival parents are randomly recombined to produce offspring. In mutation, only one parameter of the offspring (since this case we have only two parameters) which randomly selected is randomly modified. The Gaussian mutation function is used to randomly modify the mutated genes (i.e., hyperparameter). By this, a new generation of population is formed, and the process will be repeated unless stopping criteria is met (predetermined number of generation). If stopping criteria is achieved the last new generation population is evaluated using cross-validation process and the solution with the best score is selected.

A comparison example between GA and cross-validation search method to calculate the optimal hyperparameters of SVM regression is illustrated. The two methods were applied to find the tuning parameters for the water content prediction model of plant 1, since the size of data set is small and performing grid search will not be excessively time consuming.

The cross-validation search process was implemented over a grid of parameters that was composed of 100 parameters of gamma, and 100 parameters of C-value, where the limits on C-value were [C: 0.1,15] and gamma were [ $\gamma$ : 0.1, 30]. Therefore, there were 10000 (100X100) candidates to fit in 5 folds resulting in 50000 fittings. For GA search, a population of 50 individuals (i.e., each individual is composed of a single C-value and a single gamma value) with 1000 generations were used. The cross-validation score was calculated for every individual in every generation. The initial

individuals of the population were generated using random uniform distribution function where the upper and lower limits of the distributions were the same as the one used in the cross-validation grid search. Table 5.15 reports the performance of using cross-validation search and GA to calculate the optimal SVM regression hyperparameters. As shown in the table, the total computational time for GA under predefined GA specification is faster than cross-validation search. Moreover, GA obtained a better error score than cross-validation grid search. Both approaches developments were programmed in Python environment and scikit-learn package [37] was implemented for SVM regression model construction.

Figure 5.22 shows the mean cross-validation score surface as a function of gamma and C-values, the figure was generated using the cross-validation approach. As it is clear in this figure a minimum value of cross-validation MSE error exists at certain gamma and C-value. The reduction in the cross-validation MSE of water content as a function of the number generation based on GA approach is illustrated in Figure 5.23. The Figure shows the ability of GA in finding the optimum (or suboptimum) hyperparameters value in very few numbers of generation. This is due to the nature of the GA in surviving the fittest individuals (solution). Therefore, when multiple hyperparameters needed to be optimized in machine learning model construction, GA is a very efficient option to tune those parameters. Also, these results showcase that when the number of hyperparameters in the model increase, finding the optimal set of tuning parameters become challenging.

Accordingly, GA was selected to tune the hyperparameters of the SVM prediction models. The developed models were executed on a personal laptop with Intel i7 hex core 4.00 GHz processor accompanied by 16G RAM that it took on average 1.5 hour to get convergence (training different data size results in different solution time). The optimum values of C-value and gamma for developed SVM models are reported in Table 5.16. The accuracy of developed prediction models in terms of cross-validation MSE and  $R^2$  between the normalized operating plant data and SVM predictions are also reported in Table 5.16. As it can be concluded from Table 5.16, the results from SVM predictions outperform the linear regression methods. Figures (Figure 5.27, Figure 5.28, Figure 5.31 and Figure 5.32) show the variation of gamma and C-value over the GA search procedure (every point represents a candidate solution determined by GA through its search procedure) as a function of cross-validation MSE. Furthermore, the fittest cross-validation MSE is plotted against the number of generation in Figure 5.25, Figure 5.26, Figure 5.29 and Figure

5.30 for different response variables. These figures confirm the ability of GA method to obtain optimum hyperparameters values within a small number of generations.

Table 5.15. Comparison between cross-validation strategy and GA in finding best fit SVM hyperparameters

	GA	Cross-validation strategy
Computation time (second)	168	267
Mean cross-validation MSE	0.010381	0.010387
C-value	1.568	1.537
Gamma	2.07581	2.3581

Table 5.16 SVM regression models cross-validation evaluation

	Response variables	MSE	R2	C-value	Gamma
Plant1	RVP	0.00063	0.91471	1.43111	10.36337
	H <sub>2</sub> S content	0.01224	0.83320	8.68250	29.93681
	Water content	0.01038	0.44604	1.56846	2.07581
Plant2	RVP	0.001799	0.844260662	3.6886	28.8292
	Water	0.006614	0.82678173	3.7615	25.3353

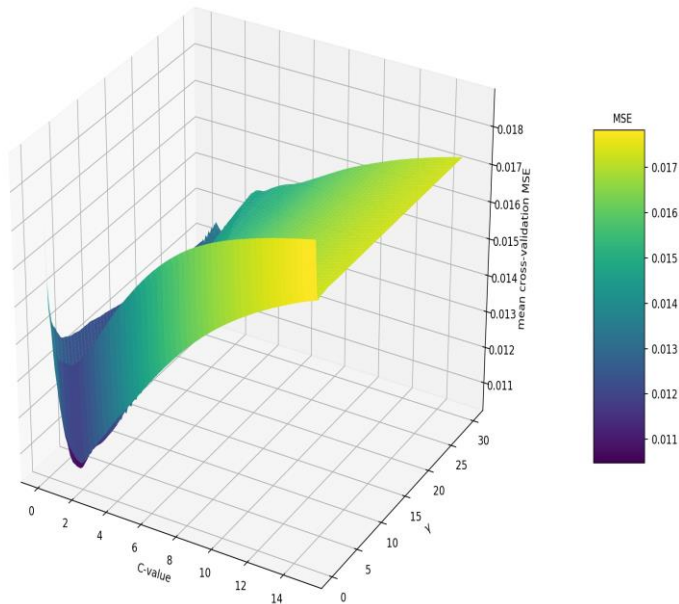


Figure 5.22. Validation surface (cross-validation MSE) as function of SVM regression hyperparameters for plant water content prediction model



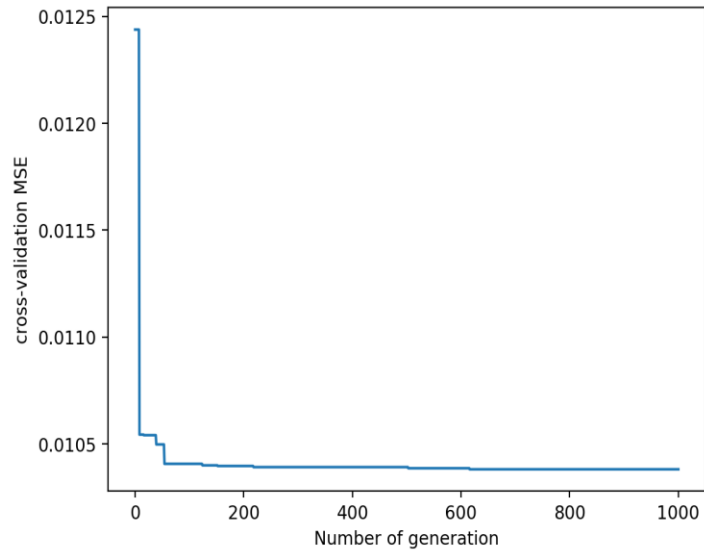


Figure 5.23. Fitness value of 'plant 1' water content prediction model as a function of generation number

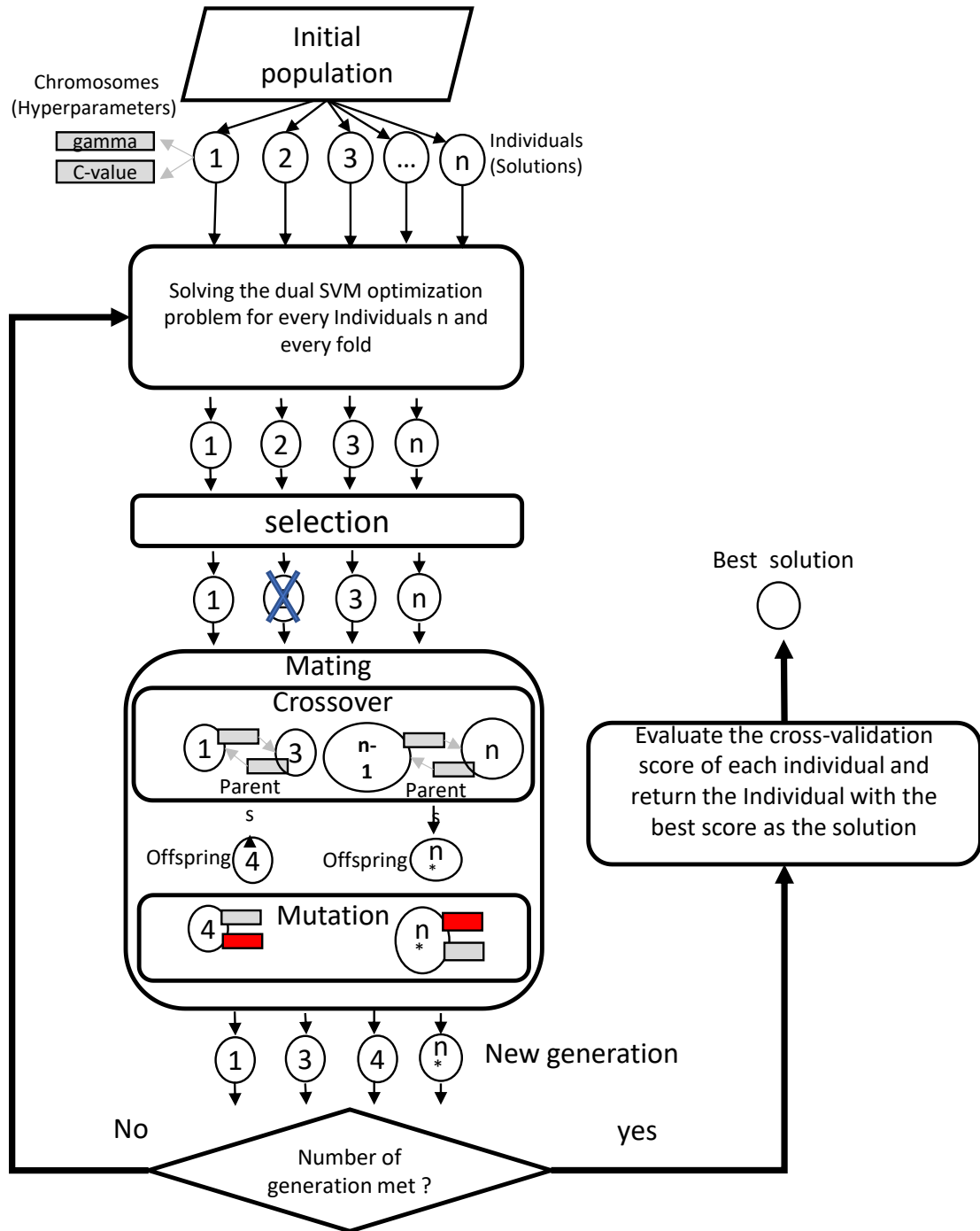


Figure 5.24. Application of GA on SVM regression hypermeters tuning process

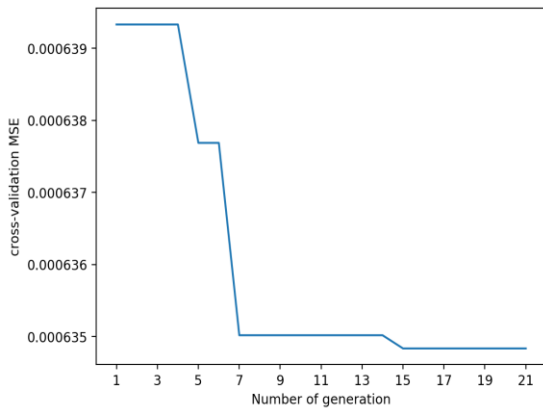


Figure 5.25. Variation of cross-validation MSE as a function of generation numbers for RVP of ‘plant 1’ prediction model

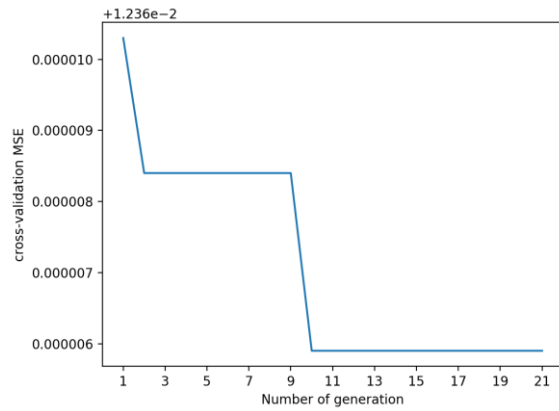


Figure 5.26. Variation of cross-validation MSE as a function of generation numbers for H<sub>2</sub>S content of ‘plant 1’ prediction model

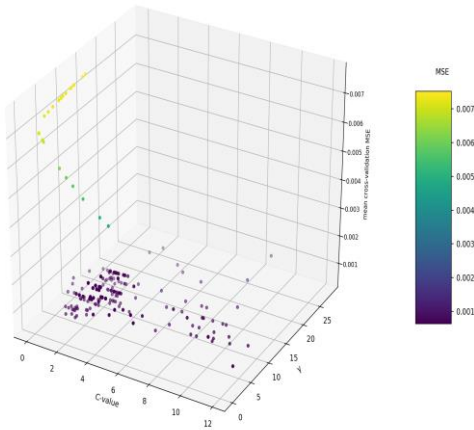


Figure 5.27. GA searching process to find optimal C-value and gamma for RVP of ‘plant 1’ prediction model

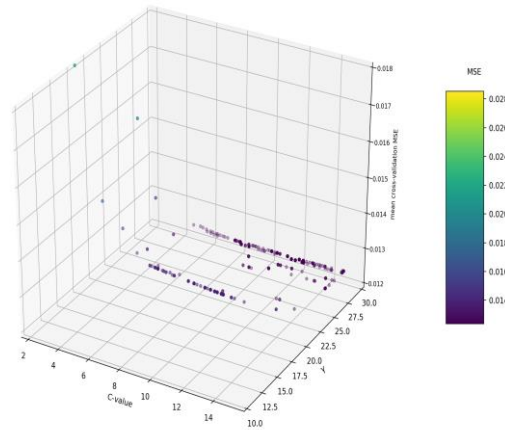


Figure 5.28. GA searching process to find optimal C-value and gamma for H<sub>2</sub>S content of ‘plant 1’ prediction model

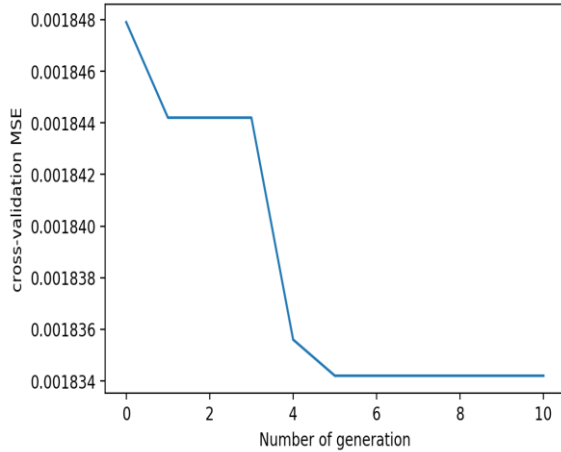


Figure 5.29. Variation of cross-validation MSE as a function of generation numbers for RVP of ‘plant 2’ prediction model

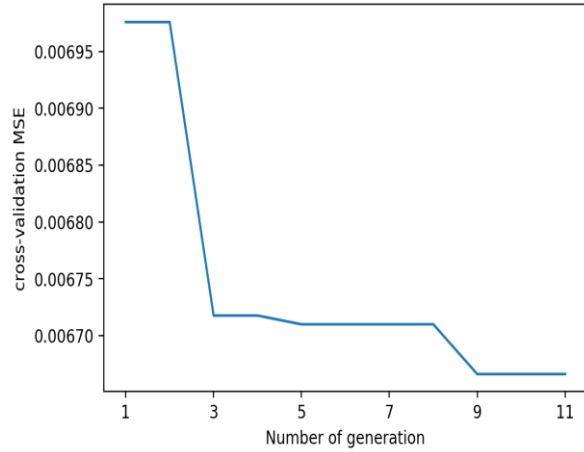


Figure 5.30. Variation of cross-validation MSE as a function of generation numbers for water content of ‘plant 2’ prediction model

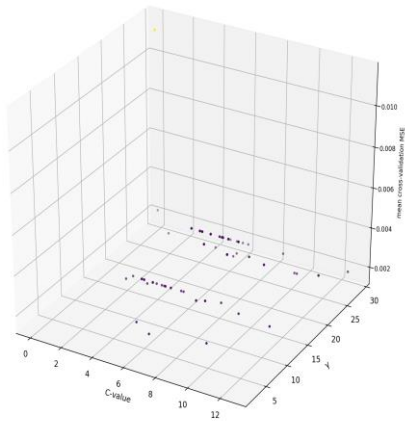


Figure 5.31. GA searching process to find optimal C-value and gamma for RVP of ‘plant 2’ prediction model

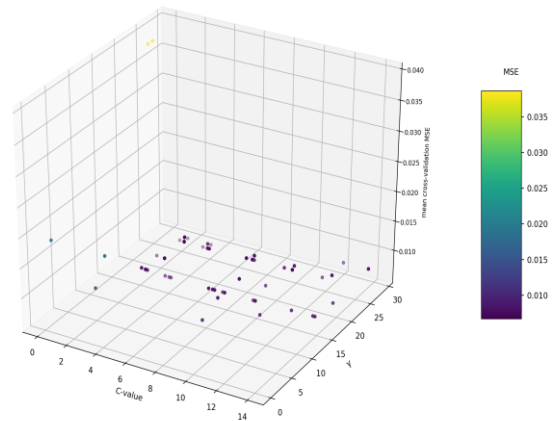


Figure 5.32. GA searching process to find optimal C-value and gamma for water content of ‘plant 2’ prediction model

#### 5.4.2.2 Deep ANN Model Development

Artificial neural network (ANN) is an empirical modelling tool that is usually referred as black-box tool [154]. One effective aspect of ANN models is the ability of ANN to incorporate different nonlinear processing elements (e.g., nonlinear activation function in different nodes) which make them capable of solving complex problems. There is existing wide range of neural network architectures to solve different types of applications. The ANN used in this work is a conventional feed-forward network consisting of an input layer, an output layer, and in between multiple hidden

layers. The ANN with multiple hidden layer architecture is referred as deep neural network. In this study, deep neural networks model are developed in Python using TensorFlow package [150], which is free open source platform for machine learning that focuses on training and inference of deep neural networks [155].

In this study, ANN were developed and trained using validation set strategy. As mentioned before (section 2.4.1), in validation set approach (test set), the data are generally divided into two sets of information, a training set and a testing set. In ANN training process, only training data are used to estimate the network internal parameters while minimizing the loss function (in this case our loss function is MSE), this score can be referred as training score (i.e., error). After the training error is minimized, predictions on the validation set are evaluated at each training epoch (calculating the validation/ generalized error). The performance of any deep neural network can be assessed by monitoring the gap between generalized error and training error. The model will be underfitting when the training error is not reduced to a satisfactory range. While overfitting will occur when the gap between testing error and training error is wide.

Training neural networks involves simultaneous monitoring of both training error and testing error in each epoch. In theory, typically at the early training epochs, both errors will start decreasing. However, after a certain epoch, it is common for the validation error to start increasing (causing overfitting) or not improving while training error keep decreasing [156]. Therefore, in this case study to avoid overfitting, the best version the prediction model is saved during training process at its best performing epoch before it overfits. Moreover, overfitting can also be avoided by adding a penalty term to the cost (loss) function. This is like the  $\lambda$  (shrinking factor) in the case of Ridge and Lasso regression. Typically, two types of regularization penalty can be used to reduce the overfittings effect when training ANN, namely  $L_1$ -norm (similar to the one used in Ridge) that used by ridge) and  $L_2$ -norm (similar to the one used in lasso) penalty term. Mathematically, the regularization term is added to the loss (ANN objective) function of the ANN and loss function  $E$  can be rewritten as follows:

$$E(\theta) = MSE + \tau R(\theta) \tag{5.2}$$

where  $\tau$  is a hyperparameter that penalize the weights norm and  $R(\theta)$  is the norm penalty, and  $\theta$  is the network internal parameters (weights and biases). The norm regularization term can be written as follows:

$$R_{L1}(\theta) = \sum_i |\theta_i| \tag{5.3}$$

$$R_{L2}(\theta) = \sum_i \theta_i^2 \tag{5.4}$$

Both term reduces the contribution of the weights, the  $L_1$ -norm usually results in sparse representation as it forces unrelated weights to vanishes (force them to be zero), whereas  $L_2$ -norm penalty, distributes the changes more equally among all parameters while minimizing the loss function [40].

#### 5.4.2.2.1 Model Design and Training

In this study, the developed deep ANN models are defined as a fully connected ANN. Three deep ANN prediction models were developed, namely: RVP and  $H_2S$  content plant 1 prediction model, water content plant 1 prediction mode, and RVP and water content plant 2 prediction. For simplicity, these models are named model A, B and C respectively. The models were trained using model 10, 2, 20 batches, respectively. While the epochs were set to be 250 for the three models. The configuration of model A network was consisting of four middle layers formed by 100 neurons, activated with ReLU function. Two hidden layers with 30 neurons that, each employed the ReLU activation function were formed to predict the water content of plant 1 (model B predictions). Model C were made using a deep network that made of 5 hidden layers with 300, 300, 300, 300 and 150 neurons respectively. ReLU function was selected as activation function for all the neurons in Model C hidden layers. Since this is a regression problem, a linear activation function was used in the output layer for all prediction models. The output layer of model A, B and C consist of 2 unites, 1 unit and 2 unites respectively

An important step in constructing deep ANN models is to select weight initialization strategy. Weight initialization is a procedure where the weights of a neural network values are randomly initialized based on certain random distribution or specific strategy (i.e., Initialization method) which determine the starting point for the training (learning optimization). Different initial weights can lead to different final set of weights with different performance characteristics [40]. He-normal was the best performing weight initialization method for model A, and C while for model B He-uniform worked better for model B. He-normal, takes samples from a truncated normal

distribution that has 0 mean and standard deviation equal to  $\sqrt{\frac{2}{fan_{in}}}$  where  $fan_{in}$  is the number of input units in the weight tensor. He-uniform draws samples from a uniform distribution within  $[-\sqrt{\frac{6}{fan_{in}}}, +\sqrt{\frac{6}{fan_{in}}}]$ .

These configurations seem to generate satisfactory accurate predictions without significant overfitting. Adam optimizer was used to train all ANN models, with its default learning rate ( $\eta$ ) 0.001, default decay rate for  $\rho_1 = 0.9$  and  $\rho_2 = 0.999$ , and default constant  $\delta = 10^{-8}$  [155]. The model A and C were trained with 70% of the available data and validated with other 30%. Since the number of data points available model B prediction are smaller, data were separated into 80% training and 20% validation.

The problem with deep networks is that they have lots of hyperparameters to tune. Therefore, these configurations selections were mostly based on trial and error, while there is no entire systematic procedure to force neural networks reach its maximum performance. Network designing include number of layers, number of neurons in each layer and activation functions is set to minimize error on the training and validation data sets while avoiding overfitting. Therefore, it is not recommended to construct more complex models, although a more complex model could capture more complex relationships in the data but would also be more susceptible to overfitting. It is important to mention that due to the stochastic nature of the training process, training the network with same data for several time will always result in different estimated function (different model). Models that might perform better under certain configuration is not necessary to happen due the model itself because of this feature (stochastic nature). Thus, fine tuning of model design seems to be even more unpractical [53]. One can say that tuning ANN models are more likely to be an art than engineering task.

#### 5.4.2.2.2 Deep ANN Results

The effect of including regularization term in ANN training process optimization was studied for predicting water content of plant 1. Cross-validation grid search was applied to look for the best penalty parameter that would enhance water content prediction. Regularization penalty terms were added to each layer of the network. The cross-validation grid search was run separately using  $L_1$ -norm and  $L_2$ -norm penalty terms. A set of values for both  $L_1$ -norm and  $L_2$ -norm regularization parameter  $\lambda$  at  $\{0,0.0010.01,0.1,1\}$  (both runs has the same grid) were evaluated separately. a

Figure 5.33 shows the effect L<sub>1</sub>- and L<sub>2</sub>-norm regularization penalty parameters on the cross-validation ANN loss function. As it can be noticed that the best performance (minimum loss function) was obtained when the regularization term was not included. Because of this, hereafter regularization penalty was not considered in developing ANN models.

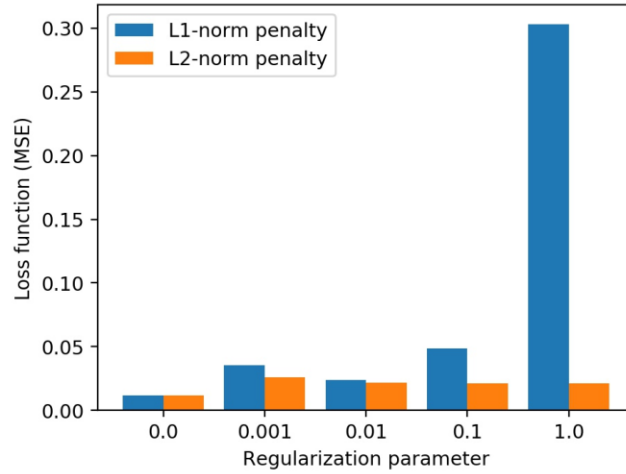


Figure 5.33. Cross-validation evaluation of L<sub>1</sub>-norm and L<sub>2</sub>-norm penalty parameters that implemented for ‘plant 1’ water content ANN prediction model

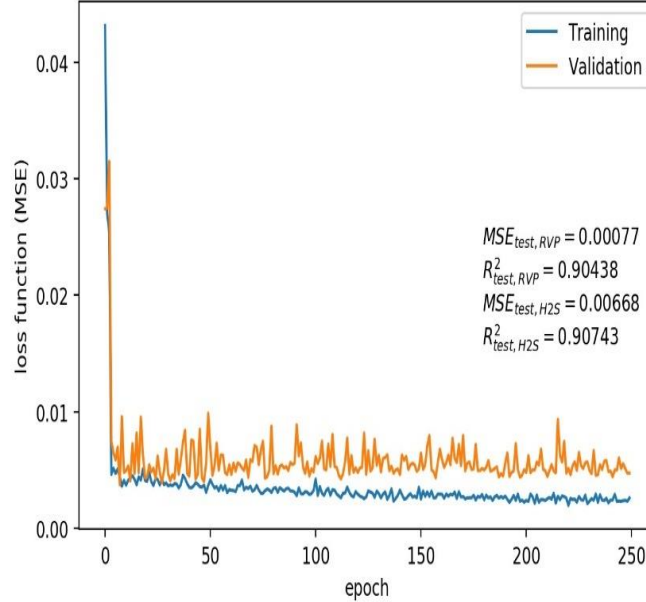


Figure 5.34. Learning curve for model A



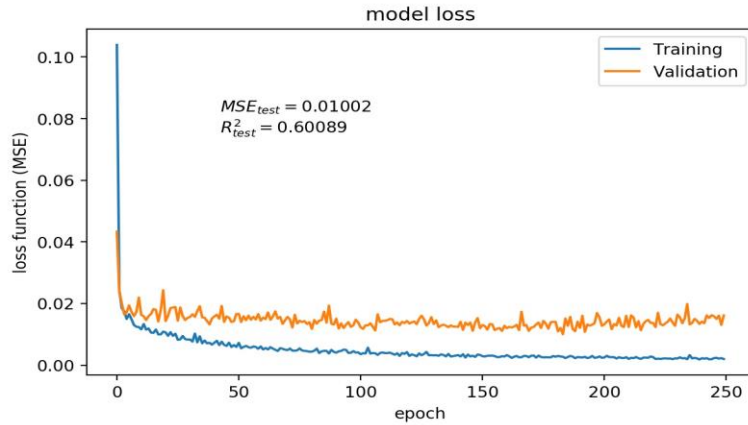


Figure 5.35. Learning curve for model B

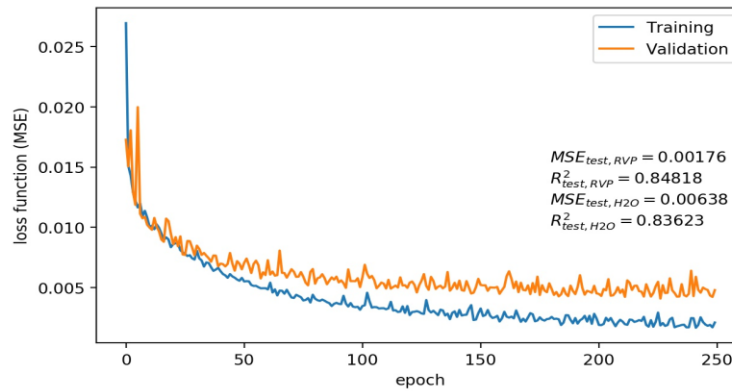


Figure 5.36. Learning curve for model C

Three figures ( Figure 5.34 - Figure 5.36) for the three prediction models that were developed, are generated to show variation of the performance metric (MSE) over the training process for both the training and testing data sets. As it can be seen from these figures, oscillations are occurred in the testing and training error for all three models due to the stochastic nature of the ANN optimization process. Also, it can be noticed that the errors start to decrease dramatically at the beginning then they reach to steady level where there is no improvement in the testing error. It can be said that the errors are converged to low values within few epochs. The gap between testing error and training error are not wide as shown in the three graphs, which means that overfitting is not occurring (negligible). The MSE and  $R^2$  for the validation sets of data are also reported in the figures for each output variables. As it can conclude from testing metrics (MSE and  $R^2$ ), between the normalized operating plant data and ANN prediction results that ANN prediction models

achieve a satisfactory performance. Accordingly, this deduced that ANN models in this study are able to capture the underlying relationship between input and output variables.

A comparison between SVM regression and ANN predictions in terms of performance metrics (MSE and  $R^2$ ) is presented in Table 5.17. Overall, ANN models performed slightly better than SVM regression. Although, the performance of ANN outperforms SVM regression by a small margin, it is more recommended to use ANN in process optimization (next task). This is because a single ANN model predicts more than one response (process output) variable while SVM regression model is limited to predict only single output variable at the time. This makes the implementation of ANN model in plant operation optimization task easier than SVM regression. Also, it should be mentioned that the  $R^2$  value for predicting Water contents is not as high as the  $R^2$  of output variables. This is because the number of data points available for training water content prediction model is much less than the data sets used to train other response variables.

Table 5.17. Comparison between ANN and SVM regression validation prediction performance

	Response variable	MSE		$R^2$	
		SVM regression	ANN	SVM regression	ANN
Plant 1	RVP	0.00063	0.00077	0.91471	0.90438
	H <sub>2</sub> S content	0.01224	0.00668	0.83320	0.90743
	Water content	0.01038	0.01002	0.44604	0.60089
Plant 2	RVP	0.00180	0.00176	0.84426	0.84818
	Water content	0.00661	0.00638	0.82678	0.83623

In order to perform consistent comparison between different developed machine learning models, predictions of the whole set of input data were obtained and validated against the whole actual sets of plants output variables. Figure 5.37, to Figure 5.40 show the prediction results against actual plant where normalized response (output) variable measurement (x-axis) are depicted verses machine learning predictions results (y-axis) for different response variables and machine learning prediction models that corresponds to different target (response) variables. Also regression performance metrics (MSE and  $R^2$ ) between the measured output plant data and machine learning

predictions for the whole set of available data are reported in these figures. As it can be seen from those figures, the nonlinear machine learning algorithms (SVM regression and ANN) outperform linear methods (Ridge and Lasso). This verifies that the relation between input and output variables are complicated and inherent nonlinearities. The predictions of both methods follow the operating plant data closely. However, some data points predictions (SVM regression and ANN) still experience higher deviation from plant actual data than others. This might be due to the inherited noise or faulty measurements of the operating plant data. It should be mentioned that the data under study is real, and typically suffers from noises and anomalies. Also, it is important to point out that there is no ground truth where we can compare prediction results with. So, when we are comparing predictions with actual plant data, we should be mindful of the fact that the plant data are not 100% accurate. This might explain the reason behind not achieving perfect prediction accuracy; however, models that were developed in this study reached an acceptable prediction accuracy. Moreover, it was shown (SVM example 2.4.5) that the machine learning models can learn actual patterns within data while avoiding inherited noise. Accordingly, it can be said that there might be a case where the machine learning models captured the actual patterns of the data although the perfect accuracy or minimum validation error are not achieved.

### **5.4.3 Effect of Some Operating Conditions on the Output Variables**

The SVM and ANN models were used to perform a sensitivity analysis on the effect of some column operating conditions on condensate product properties that include RVP and H<sub>2</sub>S. To study this effect, the feed inlet temperature was varied while all other feature variables kept constant at the plant mean values. Figure 5.41, illustrates the change in the RVP of the condensate as a function of the inlet temperature. As it can be seen in this figure, as the feed temperature increases, the RVP of the condensate decreases. This because when feed temperature increases more fraction of light component will strip off the condensate (evaporate), and this will decrease the RVP, as it is a measure of how much light components the condensate has. It is obvious from the graph that both SVM regression and ANN models are able to capture this trend. According to [157] same trend should occur to the concentration of H<sub>2</sub>S; however, it can be noted that SVM regression could not capture the effect of the feed temperature on the condensate H<sub>2</sub>S concentration. Therefore, for this reason and the better prediction performance of ANN, ANN will be selected to be implemented in plant process optimization. Additionally, Figure 5.42 depicts the effect of the reboiler temperature on the condensate H<sub>2</sub>S content. As it can be seen in this figure that when using ANN prediction

model, the consternation of the H<sub>2</sub>S reduces with higher reboiler temperature. This trend is logical because the higher the temperature the more sulfur components will vaporize off from the condensate fluid. Similar trend was found in Rahmanian *et al.* [157] where a simulation was conducted and verified with actual plant data. As we can see in this figure, the SVM regression was not able to predict this trend. This case validated the capability of proposed data-driven model to capture the real plant behaviour. As a result, it is also recommended to use these data-driven models when a preliminary study is needed to see how different operating conditions can affect the product specification.

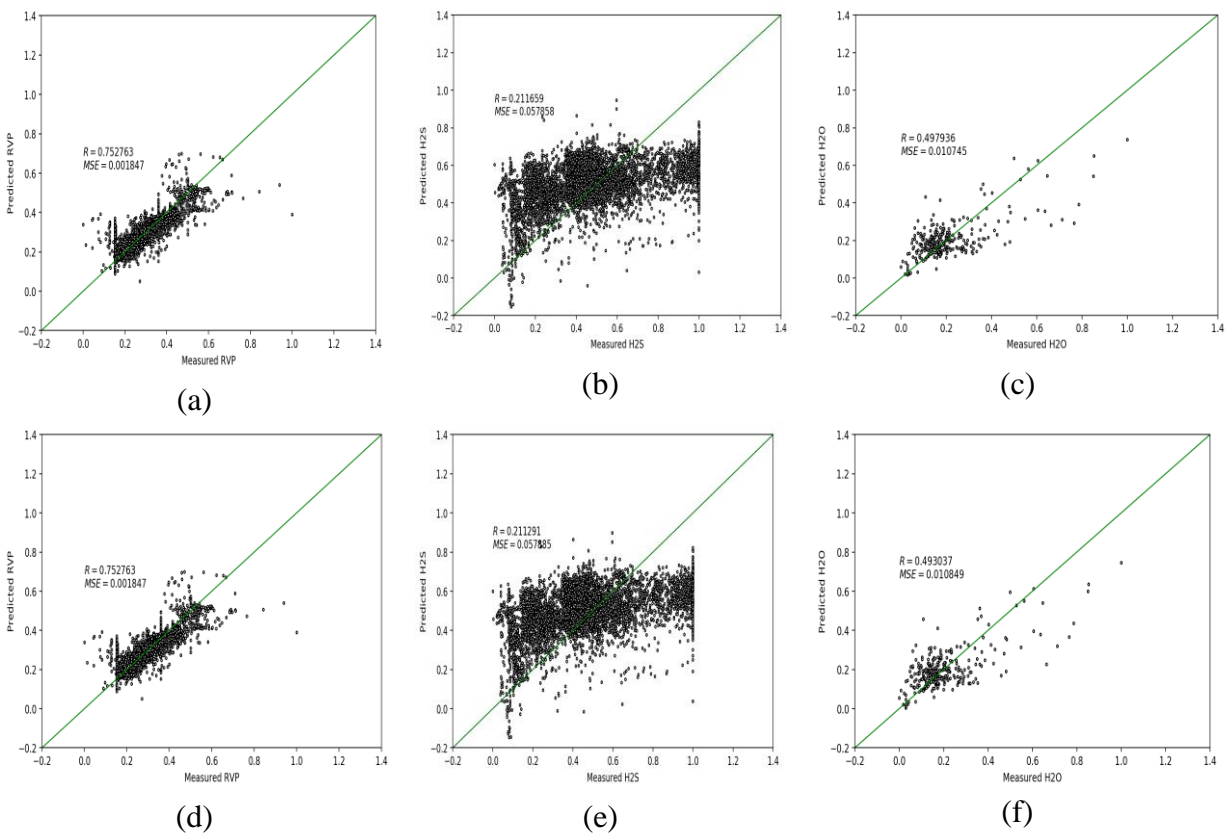


Figure 5.37. Normalized predictions vs. actual outputs of the linear developed models for plant 1 target variables: (a) Ridge model predicting RVP; (b) Ridge model predicting H<sub>2</sub>S content; (c) Ridge model predicting H<sub>2</sub>O content (d) Lasso model predicting RVP; (e) Lasso model predicting H<sub>2</sub>S content; (f) Lasso model predicting H<sub>2</sub>O content

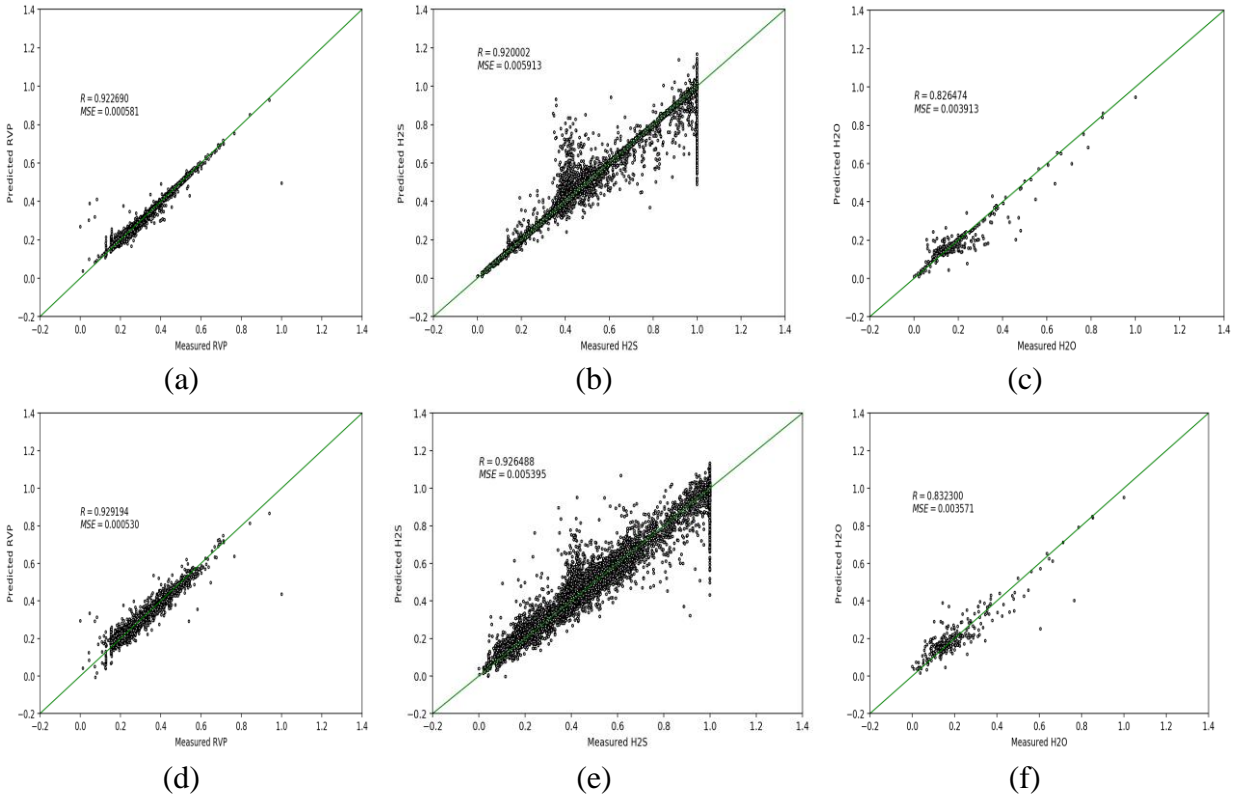
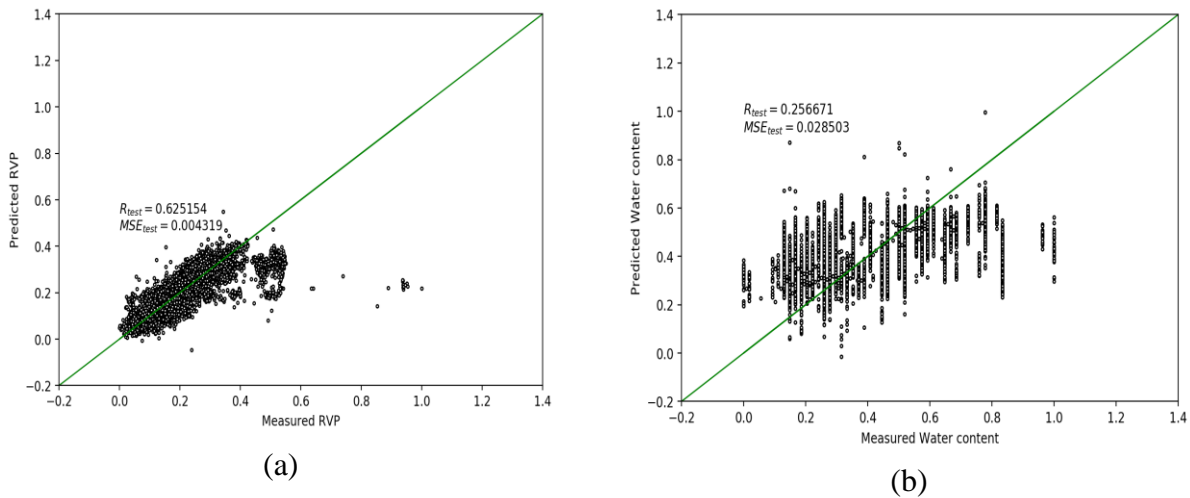
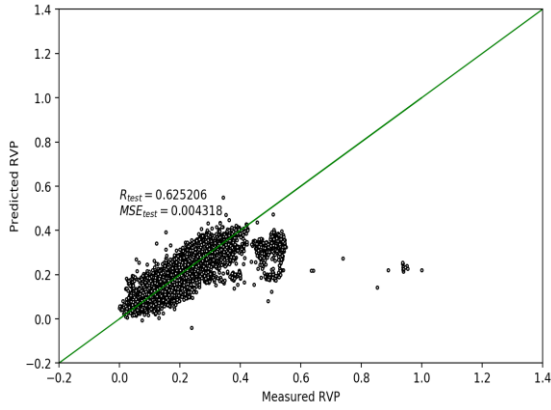
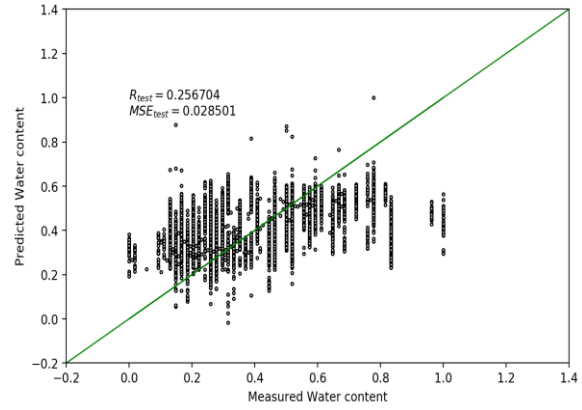


Figure 5.38. Normalized predictions vs. actual outputs of the detailed developed models for plant 1 target variables: (a) SVM model predicting RVP; (b) SVM model predicting H<sub>2</sub>S content; (c) SVM predicting H<sub>2</sub>O content (d) ANN model predicting RVP; (e) ANN model predicting H<sub>2</sub>S content; (f) ANN model predicting H<sub>2</sub>O content



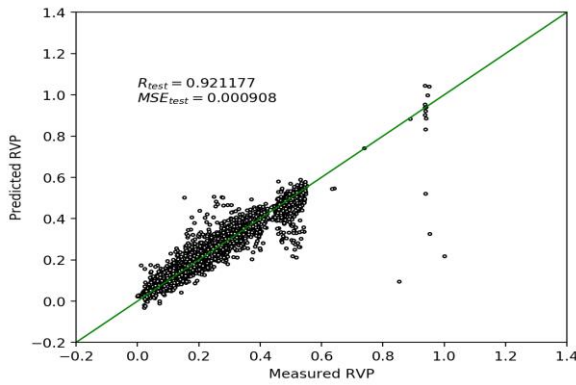


(c)

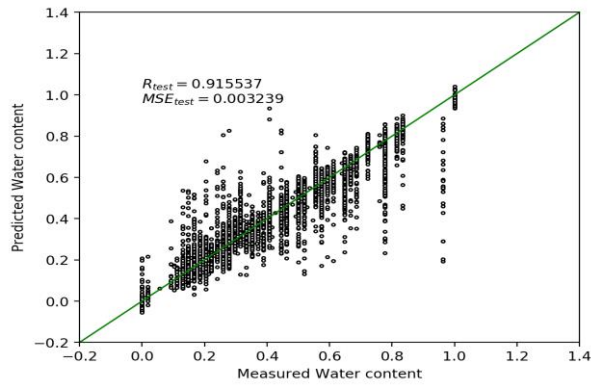


(d)

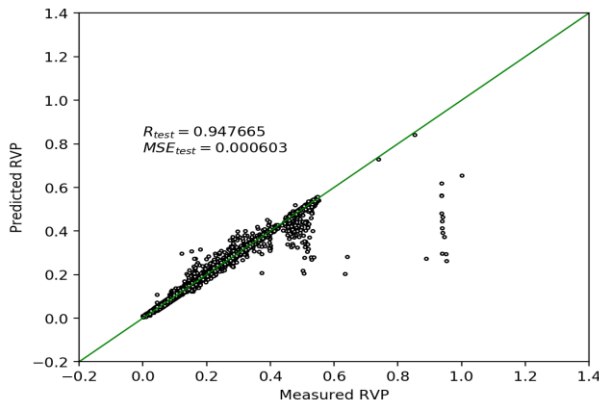
Figure 5.39. Normalized predictions vs. actual outputs of the linear developed models for plant 2 target variables: (a) Ridge model predicting RVP; (b) Ridge predicting H<sub>2</sub>O content (c) Lasso model predicting RVP; Lasso model predicting H<sub>2</sub>O content



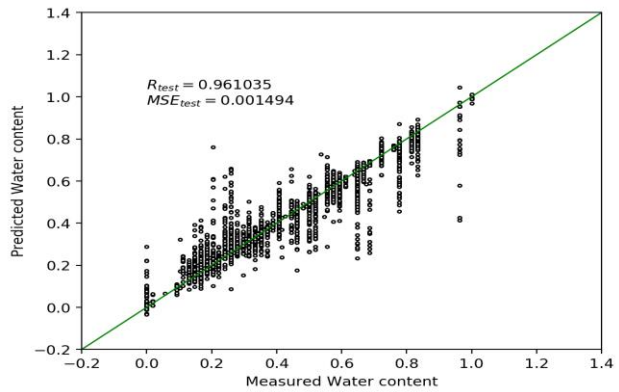
(a)



(b)



(c)



(d)

Figure 5.40. Normalized predictions vs. actual outputs of the detailed developed models for plant 2 target variables: (a) SVM model predicting RVP; (b) SVM model predicting H<sub>2</sub>O content; (c) ANN model predicting RVP content (d) ANN model predicting H<sub>2</sub>O content

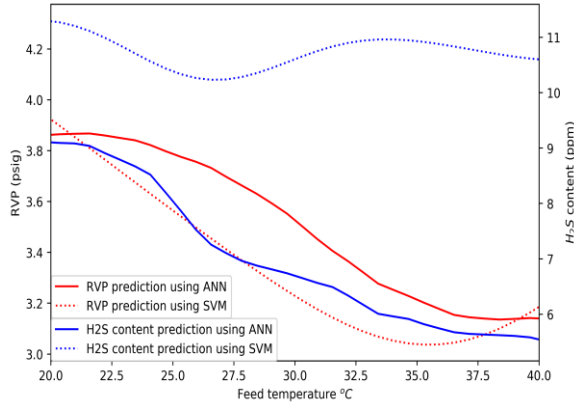


Figure 5.41. Effect of feed temperature on condensate RVP and H<sub>2</sub>S content using ANN and SVM regression prediction models

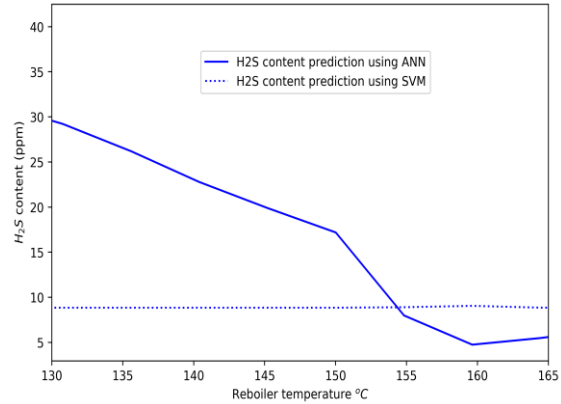


Figure 5.42. Effect of reboiler temperature on condensate H<sub>2</sub>S content using ANN and SVM regression prediction models

## 5.5 Surrogate-Based Optimization Model Developments

In this section, the optimization framework of the condensate stabilizer process operating conditions is discussed. The goal of the optimization from an operational point of view is to reduce and minimize the energy consumed by the reboiler by reducing the amount of steam used. Therefore, the objective function is to minimize the steam flowrate while maintaining RVP and H<sub>2</sub>S content and water content at the desirable levels. The data-driven machine learning models developed in the previous sections will be used as black-box function within the process optimization framework. The machine learning models will be treated as constraints from optimization modelling point of view (phase 1). However, another machine learning model for predicting objective function will be also developed and integrated within optimization framework (phase 2). The section starts by briefly explaining the optimization algorithm used in this section. Then, optimization model formulation and solutions strategies are discussed. Finally, the results of different solution strategies are presented.

### 5.5.1 Trusted-Region Algorithm

Due to the better performance of ANN models as it was shown in the previous section, they will be used as the surrogate models within the plant process optimization. In these optimization models, ANN can approximate the plants behaviour and replace the detailed first principle equations. However, these models are nonlinear and their integration into process optimization will result in nonlinear programming problem. Therefore, the trusted region algorithm was selected to optimize condensate plants operating conditions.

Trust-region method (TRM) is one of the most significant numerical optimization methods in solving nonlinear programming (NLP) problems [158]. It starts by defining a region around the current best solution, in which a quadratic model can somewhat approximate the original objective function. After that, a step forward is taken by the TRM, where the step size is determined before the improving direction. This is different from line search methods where direction is improved before step size is determined [159]. After step forward, if a significant decrease in the objective function (in the case of minimization problem) is achieved, then the approximated model can properly represent the original objective function. On the other hand, if there is slight improvement or even no improvement, then the model will not be considered as good representation of the original objective function within that region. In most of the cases the convergence of the TRM is ensured since the size of the “trust region” (typically specified by the radius in Euclidean norm) in each iteration would depend on the improvement previously made.

Conceptually, the trust-region approach replaces a n-dimensional unconstrained optimization problem by a n-dimensional constrained one. The trust-region is defined as a spherical area of radius  $\Delta_k$  in which the trust-region subproblem lies. Using quadratic method to approximate the objective function, The trust-region sub problem for k-iteration can be formulated as follows [160]:

$$\min_p m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p \quad (5.5)$$

$$s. t. \|p\| \leq \Delta_k \quad (5.6)$$

Where  $f_k$  is the objective function at  $x_k$  point,  $g_k$  is the gradient of the objective function at point k,  $B_k$  is the hessian (or a hessian approximation and  $\| \quad \|$  is the Euclidean norm ( $L_2$ -norm)).

The most critical issue facing the trust-region method is to update the size of the trust-region at every iteration. Empirical threshold values of the ratio  $\rho_k$  can help in determining the size of the trust-region.

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)} \quad (5.7)$$

The use of TRM in unconstrained optimization is more likely to have stronger assurance of local convergence than the line search method [160]. On other words, the trust-region method is more likely to converge and find optimum or suboptimum than the line search. More details on trust-



region algorithm and how to update decision variable  $x_k$  for every k iteration can be found in Gould et al., study [161].

The former description was for non-constrained TRM, however, the understudy optimization problem is NLP constrained problem. Therefore, Trust-Region Constrained Method (TRCM) is used and briefly illustrated in the following paragraphs.

The TRCM deals with constrained minimization problems of the form:

$$\min f(x) \tag{5.8}$$

$$c^u \geq c(x) \geq c^l \tag{5.9}$$

Where  $c^l$  and  $c^u$  is the lower and upper bond of the constraints  $c(x)$ . The inequality constraint can be converted to equality constraint when  $c^l = c^u$ .

As it was discussed, the trust-region method was first proposed to solve unconstraint NLP problems. However, this algorithm was further developed to be used to solve constrained NLP problems. The work done by Lalee et. al., [162] and Byrd et. al., [163] had incorporated equality constraints and inequality constrains, respectively, in TRM. More specifically, the TRCM implemented in this chapter uses Equality Constraint Sequential Quadrating Programming (EQSQP) [162] to deal with equality-constraint problems and Trust-Region Interior Point (TRIP) [163] to deal with inequality constraints. These TRCMs are capable in handling large-scale problems. More details on the implementation TRCM algorithm used in this study can be found in SciPy library [22] which is a collection of mathematical algorithms and convenience functions built on the NumPy [164] extension of Python[165].

The TRCM involves sequential solution of a quadratic programming subproblem with an additional trust region constraint [160]. The Sequential Quadratic Programming (SQP) concept are used to efficiently handle nonlinearities in the constraints and objective function. While Trust-region strategies enable the algorithm to treat convex and nonconvex problems uniformly. Additionally, the algorithm has proved to be efficient in solving a wide range of problems, even ill-conditioning and nonconvexity to some extent [163].

Since sequential programming approximation represents the core of the above-mentioned methods, a brief explanation on the SQP is presented. The fundamentals of SQP are to formulate the a subproblem based on the quadratic approximation of the Lagrangian. The Lagrangian

function  $\mathcal{L}(x, v)$  of an optimization problem with  $f(x)$  objective function and  $N$  number of constraint  $c_n(x)$  can be written as follows [166]:

$$\mathcal{L}(x, v) = f(x) + \sum_{i=1}^N v_i c_i(x) \quad (5.10)$$

Where  $v_i$  is the Lagrange multiplier of constraint  $i$ . Therefore, the SQP subproblem for  $k$  iteration can be formulated as follows:

$$\min_d g_k^T d_k + \frac{1}{2} d_k^T B_k d_k \quad (5.11)$$

S. T.

$$\nabla c_i(x_k)^T d_k + c_i(x_k) = 0, \quad i = 1, \dots, M \quad (5.12)$$

$$\nabla c_i(x_k)^T d_k + c_i(x_k) \leq 0, \quad i = 1 + M, \dots, N \quad (5.13)$$

Where  $g_k$  is the gradient of the objective function at point  $x_k$  ( $g(x_k) = \nabla f(x_k)$ ),  $B_k$  is a positive matrix approximating the Hessian matrix, at iteration  $k$  of the Lagrangian function ( $B_k = \nabla_{xx}^2 \mathcal{L}(x_k, v_k)$ ) and  $d_k$  is the direction of search.  $M$  denotes the total number of equality constraints and  $N - M$  is the total number of inequality constraints. The subproblem is solved to find the vector  $d_k$  which is used to calculate the new iteration  $x_{k+1}$  as follows:

$$x_{k+1} = x_k + a_k d_k \quad (5.14)$$

Where  $a_k$  is the step length determined by an appropriate line search procedure. And the Hessian matrix can be updated by:

$$B_{k+1} = B_k + \frac{q_k q_k^T}{q_k^T (x_{k+1} - x_k)} - \frac{B_k^T B_k}{(x_{k+1} - x_k)^T B_k (x_{k+1} - x_k)} \quad (5.15)$$

Where  $q_k$  is calculated as follows [140]:

$$q_k = g(x_{k+1}) + \sum_{i=1}^N v_i \nabla c_i(x)_{k+1} - \left[ g \left( x_k + \sum_{i=1}^N v_i \nabla c_i(x)_k \right) \right] \quad (5.16)$$

## 5.5.2 Optimization Problem Model Formulation and Solution

The data-driven machine learning models were built to simulate plant behaviour and used as replacement to detailed mechanistic equations or process data in the optimization framework.

ANN models will be used in the optimization framework because of their high accuracy predictions compared to other models. As a result of this integration, the generated surrogate-based optimization models are NLP problems. Therefore, the trusted region algorithm will be used to find the optimal operating condition of the condensate stabilizer. The optimization model is formed to minimize the consumption of the steam flowrate that powers the reboiler. Additionally, the surrogate-based optimization problem development in this study will go through two phases, namely, integration of machine learning model with optimization constraints, and then integration of machine learning model with both constraints and objective function.

### 5.5.3 Integration of Machine Learning Model within Optimization Constraint (phase 1)

As mentioned, in this study machine learning models are firstly integrated with the understudy optimization problem through constraint. Accordingly, the current optimization problem can be represented as follows:

$$\min x_{steam} \quad (5.17)$$

Subject to the following constraints

$$y_{RVP} = r(x) \quad (5.18)$$

$$y_{H_2O} = w(x) \quad (5.19)$$

$$y_{H_2S} = s(x) \quad (5.20)$$

$$RVP^U \geq y_{RVP} \geq RVP^L \quad (5.21)$$

$$H_2O^U \geq y_{H_2O} \geq H_2O^L \quad (5.22)$$

$$H_2S^U \geq y_{H_2S} \geq H_2S^L \quad (5.23)$$

$$x_i^U \geq x_i \geq x_i^L \quad (5.24)$$

Where  $r(x)$ ,  $w(x)$  and  $s(x)$  are the surrogate models that were constructed using machine learning methods, and  $x$  is the vector decision variables (operating conditions), these variables are the same variables that used as features to the machine leaning models.  $y_{RVP}$ ,  $y_{H_2O}$ , and  $y_{H_2S}$  are the computed level of the RVP, water content and H<sub>2</sub>S content (the predictions from the machine learnings models).  $RVP^U$ ,  $RVP^L$ ,  $H_2O^U$ ,  $H_2O^L$ ,  $H_2S^U$  and  $H_2S^L$  denote the desired upper (superscript U) and lower (superscript L) level of the RVP, water content and H<sub>2</sub>S content.

It should be mentioned that the optimization models for plant 2 has no H<sub>2</sub>S content. The current optimization problem in this formulation has the input (x) and output (y) variables as the decision variables, however, a reduced dimension optimization problem can be formulated by not explicitly including the output variables (y). The reduced dimension optimization problem will have only the input variables of the machine learning models as the decision variables. The reduced form of the optimization problem with implicit expression of the output variable can be rewritten as follows:

$$\min x_{steam} \tag{5.25}$$

Subject to the following constraints

$$RVP^U \geq r(x) \geq RVP^L \tag{5.26}$$

$$H_2O^U \geq w(x) \geq H_2O^L \tag{5.27}$$

$$H_2S^U \geq s(x) \geq H_2S^L \tag{5.28}$$

$$x_i^U \geq x_i \geq x_i^L \tag{5.29}$$

This formulation can handle an objective function even if there is an output variable (y) present in it. A schematic of the surrogate-based optimization strategy is shown in Figure 5.43. Firstly, initial guess of the decision variables, where in this case they are the same as the input variables for the machine learning models, are generated. Then, the based on these initial guesses, the TRCM will undergo several iterations to find the set of decision variables that minimize the objective function. While searching for optimal value, function evaluation using the introduced machine leaning model will be executed, and the output values from this function will be compared with the predetermined limits on the response variables (i.e., constraints). Therefore, this strategy will deal with the machine learning models as a black-box, where it will provide the surrogate model with inputs (decision variables that is updated in every TRCM iteration to optimize the objective function), and receive the outputs without knowing or dealing with what inside the black-box. Also, from the input and the output of the machine learning models the constraints gradients can be updated numerically. After several iterations of updating the decision variables, when a stopping criterion is achieved, the TRCM will return a solution. It should be mentioned that the return solution might violate one or two constraints depending on the starting point of the searching procedure (initial guess), however, the method will inform if any constraint violation occurred and if so will report by how much. As can be seen, this method depends heavily on the starting point

of the search (initial guess), therefore, optimization simulation can be executed several times depending on different initial guess approach.

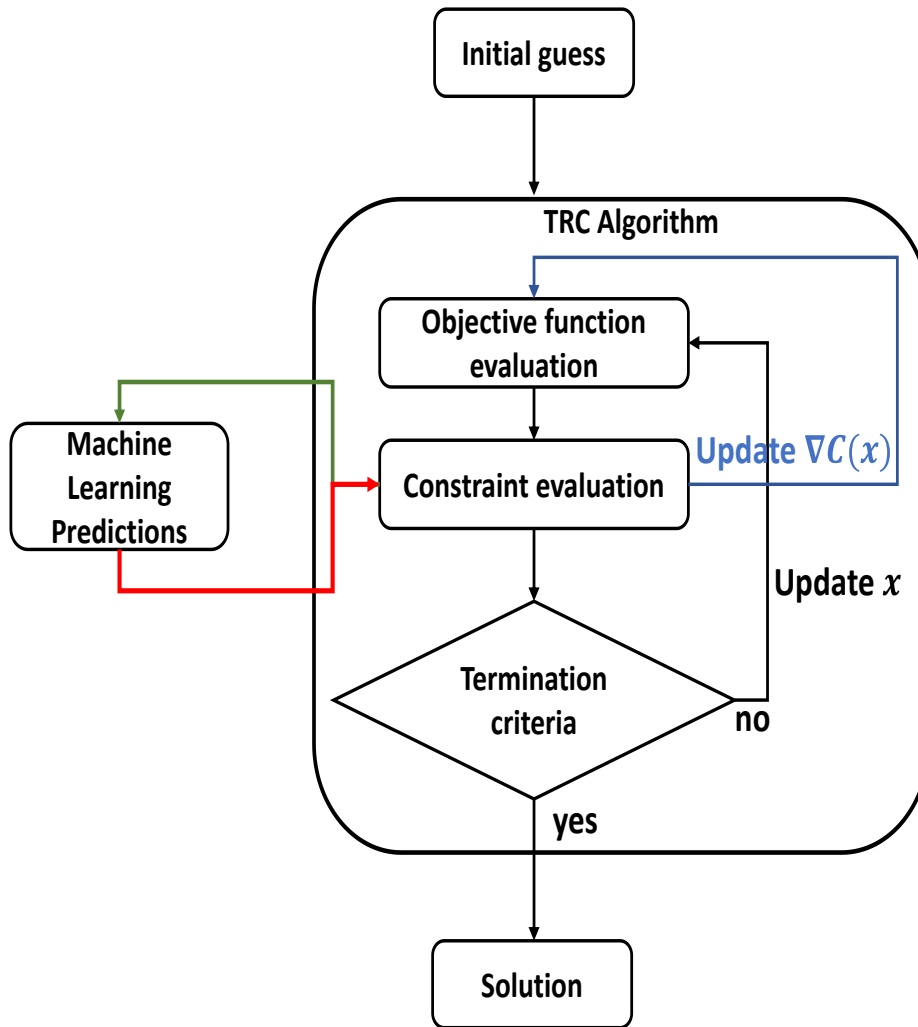


Figure 5.43. Schematic representation of the proposed optimization framework where machine learning models are integrated with optimization problem constraints

### 5.5.3.1 Optimization Results (Phase 1)

Two optimization frameworks were developed based on the illustrated method, the first one was used to optimize the operating conditions of plant 1, while the second one was developed to optimize the operating conditions of plant 2. In both frameworks, ANN models developed in the previous section were leveraged. The objective function was to minimize the energy consumption (minimization of steam flowrate). The initial guesses of the optimization algorithm were set at

mean values of the plant operating data. Decision variables for plant 1 and 2 stabilizer column were inlet temperature, column temperatures, column pressure, condensate flowrate, condensate temperature, reboiler temperature, steam flowrate, and flowrate of gas from top.

The RVP value for plant 1 and plant 2 was set to be between 5 and 8 psi, as this the recommended specification for storing condensate under UAE environment. RVP measures the volatility of the condensate, so the higher the RVP the more volatile component is in the condensate. High levels of RVP are not recommended, because the chance for the condensate to be evaporated is more, which make it difficult to store. On the other side, over stabilization is also not recommended, if the condensate will be used as fuel, as it would be difficult to combust it, especially for winter seasons. The maximum allowable water content value and sulfur content value were set at 0.01 vol% and 10 ppm respectively for plant 1. For plant 2, the water concentration should not exceed 50mh/kg The optimization problems were formulated in Python and solved using TRCM, which is provided by Scipy package [22]. Table 5.18 below indicates the selected TRCM optimization parameter

Table 5.18. Trust region constrained method parameters setting

Parameter	Explanation	value
gtol	Tolerance for termination by the norm of the Lagrangian gradient.	$10^{-6}$
xtol	Tolerance for termination by the change of the decision variable.	$10^{-6}$
barrier_tol	Tolerance for termination on the slack variables barrier parameter of inequality constraints present. The algorithm introduces slack variables that convert the inequality constraint into equality constraint, and minimize this parameter along with objective function	$10^{-6}$
initial_constr_penalty	The penalty parameter is used for balancing the requirements of decreasing the objective function and satisfying the constraints	1 [167]
maxiterint	Maximum number of algorithm iterations	8000

The upper and lower bounds were imposed on the variables according to the range of the given data set. The optimization program was run two times where the difference between these runs were in the initial guesses. The first run was based on the mean values of plant operation. In the second run, the solution of the previous run was set to be initial guess of the second run. The

obtained optimal conditions for plant 1 and the solution time using ANN model as a prediction tool for RVP, H<sub>2</sub>S content and water content are reported in Table 5.19. Table 5.20 reports the optimization results for plant 2 operation by employing the of RVP and water content ANN prediction model. Additionally, the lower and upper limits on decision variables and constraint along with plant mean values are also reported in the same tables. As it can be seen in these tables (Table 5.19 and Table 5.20), the solution time for the first run was much larger than the second run. This is because the optimization is very dependent on the initial guess, in which the initial guess of the second run was at much closer location to the local optimum. The reason for plant 1 to has a longer solution time than plant 2 was the mean of RVP value (2.6 psi) is less than the limit imposed on RVP by the constraint ( $5 \leq r(x) \leq 8$ ). This means that only few data points for plant 1 have high value of RVP, in which the ANN was built on. So, it is difficult to find a solution at that level of RVP. Whereas for plant 2, the first run took shorter time to find the solution than plant 1, because the mean of actual RVP and water content values, in which ANN model was built on, are within the constraint's limits. it can be seen from these two tables that the second run solution was very close to the first run, and the steam flow rate value was at the lower limit. Further optimization model developments where a machine learning model that predict the steam flowrate will be incorporated in the optimization modelling, will be presented in the next section. This model optimization formulation in its form does not account for the effect of input variables on the steam flowrate (objective function), in fact it deals with the steam flowrate as an input variable and keep search for a set of decision variables that satisfy the constraints limitation. So, it can be said that this formulation gives more emphasis on the constraint than objective function. But the truth is that chemical processes are govern by set of complicated mechanistic equations that include mass balance, energy balance, design geometry parameter and thermodynamic laws. Therefore, it would be more accurate and realistic to include the impact of the stabilizer column conditions on the steam flowrate, as the focus here is to minimize steam flowrate. Therefore, in the next section, an ANN network model will be developed to map the process variables without steam flowrate with the steam flowrate. And the prediction model that will be developed along with the previously developed machine learning models will be integrated into one optimization framework.

Table 5.19. Plant 1 optimization result with machine learning prediction models present in constraints only

Constraints	Unit	Solution		Limits and mean value		
		Run 1	Run 2	lower	mean plant data(nominal)	upper
$r(x)$ (RVP)	psi	5.02	5.02	5	2.60	8
$s(x)$ (H <sub>2</sub> S content)	ppm	9.46	9.58	0	10.15	10
$w(x)$ (Water content)	vol%	0.01	0.01	0.00	0.01	0.01
<b>Decision variables</b>						
Inlet flowrate	m <sup>3</sup> /h	16.27	16.32	0.14	7.42	26.52
Inlet temperature	°C	26.58	26.41	13.76	33.31	48.41
Column temperature 1	°C	147.67	147.56	108.63	131.43	147.82
Column temperature 2	°C	117.54	117.23	62.71	121.06	134.77
Column temperature 3	°C	106.78	106.38	50.33	103.60	123.22
Column temperature 4	°C	77.14	77.00	29.00	85.89	115.43
Column pressure 1	barg	2.97	2.97	2.14	2.50	3.06
Column pressure 2	barg	2.16	2.16	2.07	2.40	2.98
Condensate temperature	m <sup>3</sup> /h	104.89	104.48	74.91	103.34	126.05
Condensate flowrate	°C	13.85	13.76	10.54	15.19	33.58
Reboiler temperature	°C	156.09	155.99	152.46	158.53	164.14
Steam flowrate	kg/h	118.54	108.06	108.06	584.20	1954.22
Gas flowrate	m <sup>3</sup> /h	483.94	483.97	0.88	485.64	3228.48
Gas temperature	°C	76.52	76.04	32.23	81.34	113.44
Gas pressure	barg	2.48	2.47	2.05	2.39	2.77
time	second	2591.87	83.07			



Table 5.20. Plant 2 optimization result with machine learning prediction models present in constraints only

Constraints	Unit	Solution		Limits and mean value		
		Run 1	Run 2	lower	mean plant data(nominal)	upper
$r(x)$ (RVP)	psi	6.20	6.22	5.00	6.30	8.00
$w(x)$ (Water content)	mg/kg	37.48	37.22	0.00	56.22	50.00
<b>Decision variables</b>						
Inlet Flowrate	m <sup>3</sup> /h	358.73	358.73	235.59	358.57	497.99
Inlet Temperature	°C	116.10	116.09	94.27	118.26	129.92
Column temperature 1	°C	199.28	199.31	187.22	198.18	208.03
Column temperature 2	°C	165.32	165.31	155.94	167.35	179.37
Column temperature 3	°C	159.75	159.73	149.16	161.95	175.57
Column temperature 4	°C	144.07	144.07	127.99	143.54	157.40
Column temperature 5	°C	128.72	128.73	114.33	127.30	141.69
Column temperature 6	°C	91.23	91.23	78.31	90.63	104.39
Column temperature 7	°C	83.36	83.36	70.80	82.71	100.17
Column pressure drop	bar	0.04	0.04	0.01	0.08	0.13
Column pressure	barg	9.88	9.88	9.30	9.75	11.04
Condensate flowrate	m <sup>3</sup> /h	293.81	293.81	126.24	294.05	391.87
Condensate temperature	°C	197.81	197.82	186.88	197.64	207.23
Reboiler temperature	°C	174.88	174.89	161.05	173.24	183.92
	°C	202.28	202.29	190.74	201.31	211.91
Steam flowrate	kg/h	10671.59	10671.59	10671.59	16992.34	27547.63
Overhead flowrate	m <sup>3</sup> /h	11618.98	11618.98	7615.07	11618.97	16819.40
Overhead temperature	°C	72.40	72.40	59.55	71.55	92.19
Overhead pressure	barg	9.65	9.64	7.64	9.77	10.99
time	second	1389.82	23.29			

#### 5.5.4 Integrating of Machine Learning Model Within Optimization Constraints and Objective Function

In this section, the building of the optimization model that leverage machine learning models with optimization modelling objective function and constraints is presented. Firstly, a machine learning model for predicting the objective function (steam flowrate) is developed. Then the optimization model that can integrate the previously developed model in section (5.4.2.2), and the steam flowrate prediction model is reformulated. Finally, optimization results of the developed model are generated and discussed.

#### 5.5.4.1 Steam Flowrate Prediction Model Development

ANN models were developed to predict the steam flow rate for plant 1 and plant 2 individually. In order to develop these models, inlet gas flowrate, inlet temperature, column temperatures, column pressure, condensate flowrate, condensate temperature, reboiler temperature and steam flowrate were implemented as model input data (see Table 5.9 for plant 1 data, and Table 5.11 for plant 2 data) whereas steam flowrate was used as an output. The ANN configuration for predicting steam flowrate of plant 1 was made of two hidden layers, each has 30 neurons, activated by ReLU. Plant 2 steam flowrate prediction model was constructed using 3 hidden layers, each consist of 50 neurons. Both models have single output layer utilizing linear activation function. Number of batches for plant 1 and plant 2 steam flowrate prediction models were 10 and 20 respectively, and 250 epochs were used in both models. Adam optimization algorithm with its default setting parameters was used to train both models. Before training the model, input and output variables for this model were normalized. TensorFlow [155] library was used to build the models, and the codes were developed in Python.

The results of model validation for predicting the steam flowrate of plant 1 and plant 2 stabilizer columns are depicted in Figure 5.44 and Figure 5.45. As can be seen, the performance of the trained networks is excellent as steam flowrate predictions follow the plants data very well. The statistical results of the testing data set are shown in in the same figures. As can be seen here the  $R^2$  value is very close to 1 and MSE values are close to zero. This means that the developed ANN models are able to capture the relationship between input variables and steam flowrate.

The learning curves (i.e., variation of the performance metric (MSE) over epochs, for both the training and testing data sets) of steam flowrate prediction models are presented in Figure 5.46 for plant 1 and Figure 5.47 for plant 2. The figures show that training process reach to a very small validation error within few epochs. Also, it should be noted that overfitting is not occurring in both figures.

Figures (Figure 5.48-Figure 5.51) show the predictions of RVP and  $H_2S$  content for plant1 condensate and RVP and water content for plant 2 condensate using two approaches :(I), entering the whole set of original plants data s inputs to previously developed model A and model C, and (II) by predicating first the steam flowrate using the current developed ANN models, and then used the predicted value of the stem flowrate along with other plant data as input to model A and model C. As it can be seen from these figures that using intermediate machine learning model to

predict steam flowrate then, use the predicted steam flowrate values along with other plant data to predict the former response variables, does not differ from using original set of features (inputs) to directly predict the former response variables of plant 1 and plant 2. In fact, both approach (I and II) have identical predictions where  $R^2$  values and MSE values are identical for all response variables (plant 1, RVP and  $H_2S$ ) (plant 2, RVP and water content) Therefore, this confirm that the steam flowrate prediction models are very accurate and can be efficiently implemented in developing the process optimization model for objective function estimations.

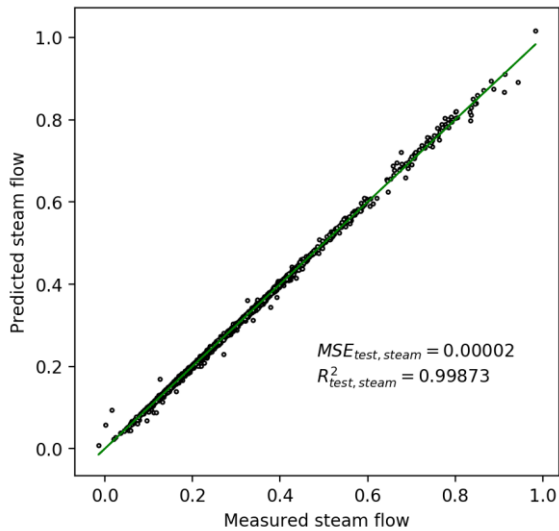


Figure 5.44. Normalized actual vs ANN predictions of steam flowrate for plant 1 condensate stabilizer column

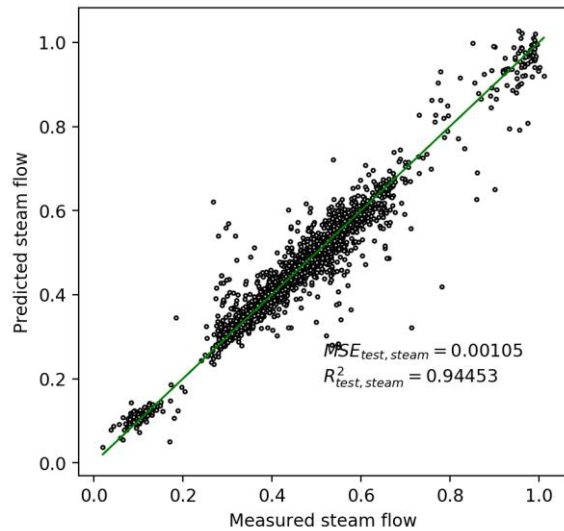


Figure 5.45. Normalized actual vs ANN predictions of steam flowrate for plant 2 condensate stabilizer column

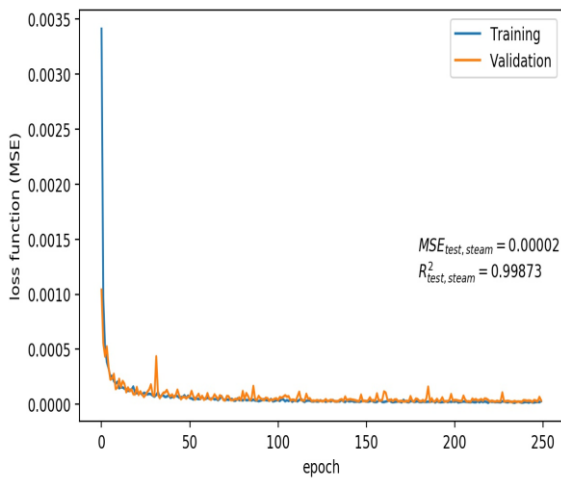


Figure 5.46. Learning curve for ANN steam flowrate of 'plant 1' prediction model

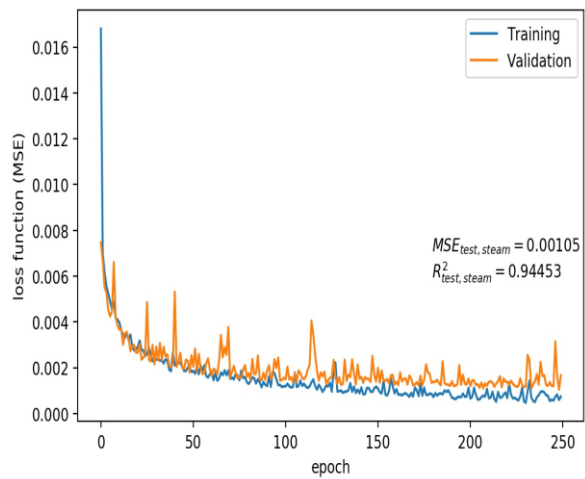


Figure 5.47. Learning curve for ANN steam flowrate of 'plant 2' prediction model

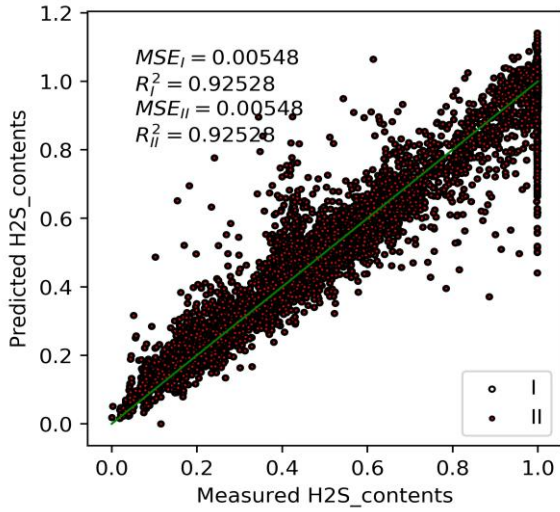


Figure 5.48. Normalized actual vs ANN predictions of RVP for plant 1 using actual steam flowrate data (I) and predicted steam data (II)

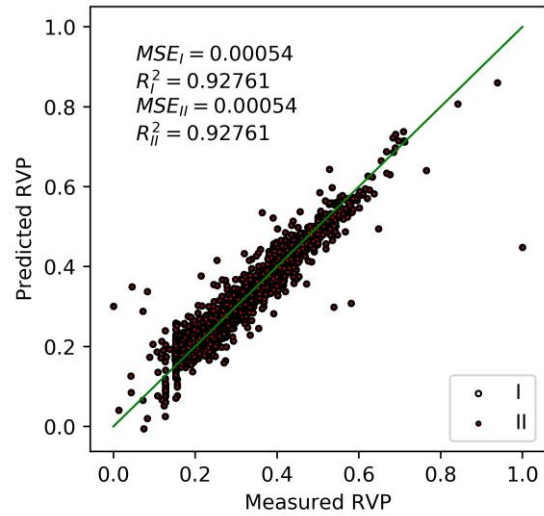


Figure 5.49. Normalized actual vs ANN predictions of H<sub>2</sub>S content for plant 1 using actual steam flowrate data (I) and predicted steam data (II)

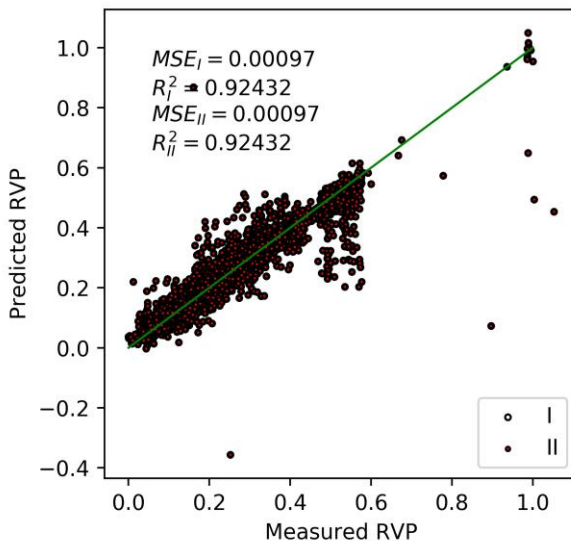


Figure 5.50. Normalized actual vs ANN predictions of RVP for plant 2 using actual steam flowrate data (I) and predicted steam data (II)

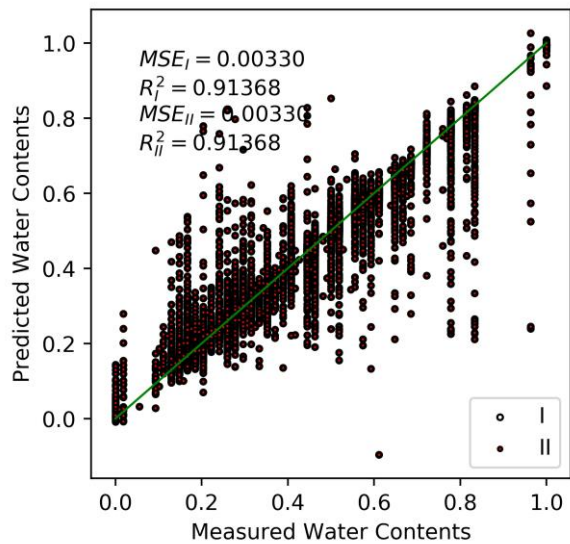


Figure 5.51. Normalized actual vs ANN predictions of water content for plant 2 using actual steam flowrate data (I) and predicted steam data (II)

### 5.5.4.2 Optimization Problem Formulation (phase 2)

The optimization model presented in equations ((5.25)- (5.29)) was modified to include the machine learning model for predicting the steam flowrate, can be rewritten as follows:

$$\min T(x^*) \quad (5.30)$$

Subject to the following constraints

$$RVP^U \geq r(x^*, T(x^*)) \geq RVP^L \quad (5.31)$$

$$H_2O^U \geq w(x^*, T(x^*)) \geq H_2O^L \quad (5.32)$$

$$H_2S^U \geq s(x^*, T(x^*)) \geq H_2S^L \quad (5.33)$$

$$x_i^{*U} \geq x_i^* \geq x_i^{*L} \quad (5.34)$$

Where  $T(x^*)$  the ANN mode; prediction of the steam flowrate,  $x^*$  is the modified set of decision variables which is same as  $x$ , but without steam flowrate.  $r(x^*, T(x^*))$ ,  $w(x^*, T(x^*))$  and  $s(x^*, T(x^*))$  are the prediction models for RVP, water content and sulfur content respectively. In this formulation the steam flowrate will be predicted first using the modified set of input variable  $x^*$ , and then the predicted steam flowrate  $T(x^*)$  along with the  $x^*$  will be used to perform predictions for RVP ( $r(x^*, T(x^*))$ ), water content ( $w(x^*, T(x^*))$ ) and sulfur content ( $s(x^*, T(x^*))$ ).

The surrogate-based optimization strategy that was developed in section (5.5.2.1), was modified to include the new model that predict the steam flow rate A schematic including this modification is illustrated in Figure 5.52. Firstly, initial guess of the decision variables, where in this case they are the same as the input variables for the steam flowrate prediction model, is generated. Then, based on this initial guess, the TRCM will undergo several iterations to find the set of decision variables that minimize the objective function (steam flowrate). At each iteration, function evaluation for the steam flowrate using the developed model ( $T(x^*)$ ) is executed, and the graduate of the objective function, as well as the decision variables are updated accordingly. While searching for optimal value, function evaluation using the RVP, water content and sulfur content (i.e.,  $r(x)$ ,  $w(x)$ ,  $s(x)$ ) prediction models will be performed by entering the decision variables *set* along with the predicted steam flowrate  $T(x^*)$  as an input to those models. Like previously developed surrogate-based model, the output values from these functions are compared with the predetermined constraint limits. Therefore, this strategy will deal with the machine learning models as a black-box model within two optimization problem elements, namely, objective function and constraints. As it was stated before, that the internal of these black-box models will

not be seen by the TRCM, in fact, TRCM will only be dealing with inputs and outputs of these models.

### 5.5.4.2.1 Optimization Results (phase 2)

Based on the proposed optimization framework presented in Figure 5.52, optimization models to minimize the steam flowrate were developed for plant 1 and for plant 2. The decision variables boundaries and constraints limits were the same as the one used in optimization problem of phase 1. Similar to what was done in phase 1, the optimization programs were run two times, the first run used the mean values of the plants data sets as initial guess, and the second time by utilizing the solution of the previous run as initial guess. It should be mentioned that the temperature of the column at different trays are substituted by one average temperature, same thing for the column pressure.

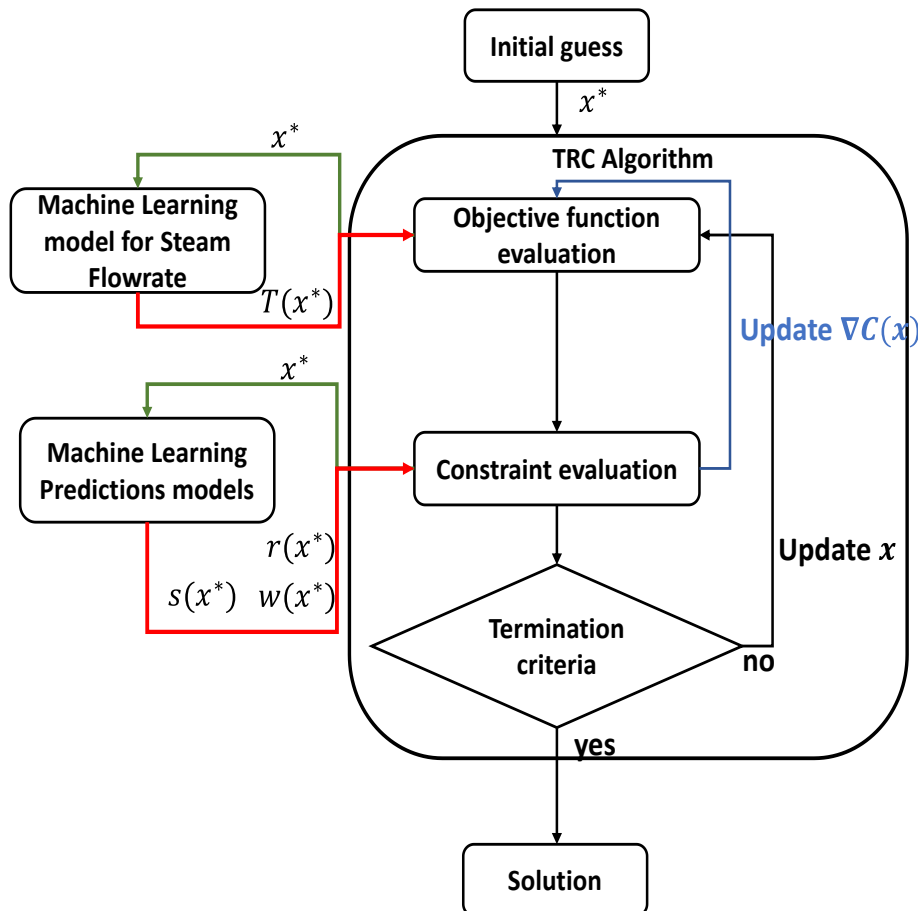


Figure 5.52. Schematic representation of the proposed optimization framework where machine learning models are integrated with optimization problem constraints and objective function

It can be seen from Table 5.21 that the optimization program was not able to satisfy the constraint limits at the first run, however, the solution of the second run is within constraint limits. This points out, the importance of the initial guess in the current optimization problem, where different initial guess can lead to a different result. Solution time for second run was much faster than the first run due to the closer proximity to the optimum. It should be noticed that the optimization achieved a very good reduction in the steam flowrate from its mean value by manipulating other variables. The optimization program decided to reduce the inlet flowrate, and increase the inlet temperature, both actions have reduced the amount of heat that should be provided by the column, and thus decreases the steam consumption. Under these conditions of condensate specification, the optimization program was able to gain reduction in steam flowrate from its mean value by almost 50%. It is worth mentioning that, the results obtained from using the phase 2 approach are different from phase 1 approach, which implies the importance of modelling the steam flowrate behaviour. From Table 5.22 it should be noticed that the optimization was able to find a minimum value for the steam flowrate under the given variables limits and constraints. The optimal solution was not violating any constraints limits and it reduced steam flow rate from its mean value (16992.34 kg/hr to 14772.80 kg/hr) by 13% reduction. The first and second run of this problem have no big difference, and the solution time for second run was faster. Additionally, since the determined optimal operating conditions are not very far from the mean plant operation, the optimization program suggested a minor tuning to the operating conditions to reduce steam flowrate.

From these results, it can be concluded the selection of initial guess is critical. After that, a case study is conducted, in which different initial guesses are randomly sampled from a uniform distribution within upper and lower range of decision variables  $[x_i^U, x_i^L]$ . To make this case study more realistic, the condensate flow rate should not be less than 80% of its maximum value. 100 starting guesses are generated, and the optimization model is solved for every set of initial guesses. After that, every solution is assessed and determined if it is within constraints limits or not. If the solution did not violate the constraint limits, it is saved, else it is excluded. The process of applying different starting guesses is presented in Figure 5.53. Finally, the solution that corresponds to the lowest objective function (steam flowrate) is considered as the best optimum solution.

Table 5.23 reports the optimal operating conditions of plant 1. As it can be observed that steam flowrate obtained from the optimization model is less than the mean steam consumption. However, if we compare these results to Table 5.21, we see that the minimum steam flowrate in the Table

5.21 (previous case) was less than this case. This is because the condensate flowrate at the current case is forced to be high, which requires more energy to separate the light end component from the heavy end component. It can also be noted that, the optimization program decided to reduce the average column temperature and generate condensate at higher level of RVP than the mean value. This clearly will result in reducing the steam flowrate, nevertheless, it will also let lighter component to stay with condensate and increase the value of RVP. To put it another way, the optimization program traded-off between the RVP value and the steam flowrate, it allowed the RVP to increase, by reducing the steam flowrate. This increase in RVP was acceptable because it did not violate the desired specification of the condensate. It can be concluded from this results that, working under mean conditions was not ideal, because the column over stabilized (mean value of RVP was 2.6) the condensate more than it is needed. Over stabilization require extra energy (more steam) to increase the fluid temperature and let lighter components escape from the fluid. Operating the column under the optimal conditions will help reducing the extra cost associated with energy consumption that arise from operating far from optimal or suboptimal conditions.

Table 5.24 shows the optimal results for plant 2. As it can be seen in this table, there was significant reduction in steam consumption by tuning the column operating conditions. The main suggestion by the optimization program is to reduce the column temperature, thus less steam is needed. Reducing column temperature results in less separation of light end hydrocarbon from the condensate. Therefore, more of light hydrocarbon will go with the condensate. However, the reduction in this separation is made while satisfying the condensate RVP desired specification. The reduction in separation can be reflected by the value of RVP, the optimization allows the condensate to have higher value of RVP (i.e., lighter component), but not violating the imposed constraint on RVP. In other words, under mean (nominal conditions) the column separation efficiency was above what is required (over stabilization in which mean RVP = 6). By performing optimization, it was suggested to work in lower temperature allowing less separation of light component, while not violating the required condensate specification. Lower column temperature can be inferred from the lower overhead temperature and lower column average temperature and lower temperature of the stream (circulation column fluid) entering reboiler shell side. Thus, less steam is consumed at lower column temperature, but more of the lighter components are present in the condensate within the acceptable range. Additionally, it should be noticed from the table



that the suggested feed temperature is slightly more than the operating mean value. This might be happened to compensate for the reduction in heat duty that associated with reducing steam flowrate

Table 5.21. Plant 1 optimization results with ANN models used to predict condensate specification (constraints) and steam flowrate (objective function)

Constraints	Unit	Solution		Limits and mean value		
		Run 1	Run 2	lower	mean plant data (nominal)	upper
$r(x^*)$ RVP	psi	4.38	5.00	5.00	2.60	8.00
$s(x^*)$ H <sub>2</sub> S content	ppm	4.52	7.11	0.00	10.15	10.00
$w(x^*)$ Water content	vol%	0.01	0.01	0.00	0.01	0.01
<b>Objective function</b>						
Steam flowrate	kg/hr	668.84	276.33	108.06	584.20	1954.22
<b>Decision variables</b>						
Inlet flowrate	m <sup>3</sup> /h	14.67	4.66	0.14	7.42	26.52
Inlet temperature	°C	43.48	48.40	13.76	33.31	48.41
Column temperature	°C	114.88	118.37	62.67	110.50	130.31
Column pressure	barg	3.02	2.63	2.11	2.45	3.02
Condensate temperature	°C	87.95	95.08	74.91	103.34	126.05
Condensate flowrate	m <sup>3</sup> /h	10.85	10.54	10.54	15.19	33.58
Reboiler temperature	°C	157.40	158.48	152.46	158.53	164.14
Gas flowrate	m <sup>3</sup> /h	484.25	489.48	0.88	485.64	3228.48
Gas temperature	°C	62.25	56.78	32.23	81.34	113.44
Gas pressure	barg	2.37	2.27	2.05	2.39	2.77
time	second	5899.49	144.47			

Table 5.22. Plant 2 optimization results with ANN models used to predict condensate specification (constraints) and steam flowrate (objective function)

Constraints	Unit	Solution		Limits and mean value		
		Run1	Run2	lower	mean	upper
$r(x^*)$ RVP	psi	5.55	5.63	5.00	6.30	8.00
$w(x^*)$ Water content	mg/kg	50.00	50.00	0.00	56.22	50.00
<b>Objective function</b>						
Steam flow	kg/hr	15128.09	14772.80	10671.59	16992.34	27547.63
<b>Decision variables</b>						
Inlet flowrate	m <sup>3</sup> /h	358.72	358.68	235.59	358.57	497.99
Inlet temperature	°C	118.27	117.68	94.27	118.26	129.92
Column temperature	°C	138.84	139.04	126.25	138.81	152.37
Column pressure drop	bar	0.09	0.08	0.01	0.08	0.13
Column pressure	barg	10.06	10.02	9.30	9.75	11.04
Condensate flowrate	m <sup>3</sup> /h	293.95	293.61	126.24	294.05	391.87
Condensate temperature	°C	197.71	199.12	186.88	197.64	207.23
Reboiler Temperature (inlet and output)	°C	173.32	173.92	161.05	173.24	183.92
	°C	201.20	199.15	190.74	201.31	211.91
Overhead flowrate	m <sup>3</sup> /h	11618.94	11618.83	7615.07	11618.97	16819.40
Overhead temperature	°C	71.60	71.77	59.55	71.55	92.19
Overhead pressure	barg	9.55	9.45	7.64	9.77	10.99
time	second	256.65	71.00			

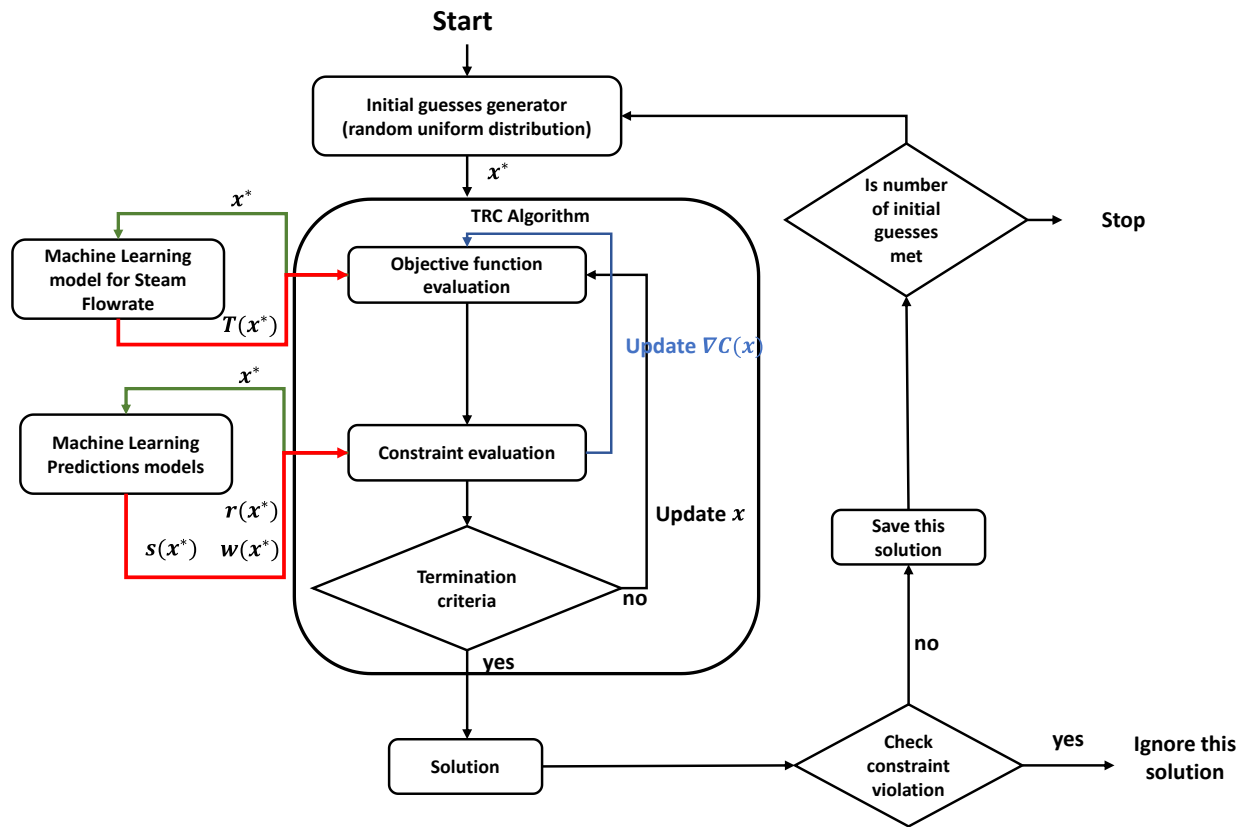


Figure 5.53. Schematic representation of the proposed optimization framework where machine learning models are integrated with optimization problem constraints and objective function under different initial guess

Table 5.23. The optimal operating conditions of plant 1 over all initial guess

Constraints	Unit	Solution	Limits and mean value		
		optimal	lower	mean plant data(nominal)	upper
$r(x^*)$ RVP	psi	6.00	5	2.6	8
$s(x^*)$ H <sub>2</sub> S content	ppm	2.21	0	10.15	10
$w(x^*)$ Water content	vol%	0.00	0	0.01	0.01
<b>Objective function</b>					
Steam flow	kg/hr	456.02	108.06	584.2	1954.22
<b>Decision variables</b>					
Inlet flowrate	m <sup>3</sup> /h	15.69	0.14	7.42	26.52
Inlet temperature	°C	26.37	13.76	33.31	48.41
Column temperature	°C	97.09	62.67	110.5	130.31
Column pressure	barg	2.91	2.11	2.45	3.02
Condensate temperature	°C	97.51	74.91	103.34	126.05
Condensate flowrate	m <sup>3</sup> /h	26.86	10.54	15.19	33.58
Reboiler temperature	°C	156.95	152.46	158.53	164.14
Gas flowrate	m <sup>3</sup> /h	46.32	0.88	485.64	3228.48
Gas Temperature	°C	32.32	32.23	81.34	113.44
Gas pressure	barg	2.77	2.05	2.39	2.77

Table 5.24. The optimal operating conditions of plant 2 over all initial guess

Constraints	Unit	Solution	Limits and mean value		
		optimal	lower	mean	upper
$r(x^*)$ RVP	psi	7.61	5	6.3	8
$w(x^*)$ Water content	mg/kg	50.00	0	56.22	50
<b>Objective function</b>					
Steam flow	kg/hr	13561.89	10671.59	16992.34	27547.63
<b>Decision variables</b>					
Inlet flowrate	m <sup>3</sup> /h	414.73	235.59	358.57	497.99
Inlet temperature	°C	120.95	94.27	118.26	129.92
Column temperature	°C	137.32	126.25	138.81	152.37
Column pressure drop	bar	0.07	0.01	0.08	0.13
Column pressure	barg	10.45	9.3	9.75	11.04
Condensate flowrate	m <sup>3</sup> /h	352.82	126.24	294.05	391.87
Condensate temperature	°C	197.07	186.88	197.64	207.23
Reboiler temperature	°C	166.05	161.05	173.24	183.92
	°C	201.75	190.74	201.31	211.91
Overhead flowrate	m <sup>3</sup> /h	13331.64	7615.07	11618.97	16819.4
Overhead temperature	°C	62.44	59.55	71.55	92.19
Overhead pressure	barg	9.38	7.64	9.77	10.99

## 5.6 Conclusion and Future Work

In this chapter, a comprehensive development of machine learning models and optimization framework for 2 condensate stabilizer units was performed successfully. Data-driven model development that include outliers' removal and hyperparameters tuning was implemented. A generalized genetic algorithm approach was developed to optimally tune the SVM regression hyperparameters. The developed genetic algorithm can be applied to any supervised machine learning algorithm to compute its optimal set of hyperparameters. Different machine learning models were developed and compared with each other. The results showed that nonlinear models outperform linear models, implying that the relationships between inputs and outputs are nonlinear. The performance of both SVM regression and ANN were very good in predicting plant outputs. However, ANN was preferable because single ANN model can predict more than 1 output variable simultaneously. Another reason is that ANN models were able to capture actual plant trends (behaviour) as ANN simulation results were in good agreement with plant simulation results from literature. Machine learning models have shown potential to capture plant behaviour based on data only and offer reliable and accurate predictions. And therefore, they can be used as an effective replacement of the complicated mechanistic model within process optimization.

An NLP model was developed and solved by implementing trust-region method (i.e., numerical optimization algorithm) that integrates the developed machine learning model as surrogate model to determine the plant optimal operating conditions. The developed surrogate models were integrated into the optimization model through the objective function and associated constraints. Since the numerical optimization method is sensitive to initial guess, a multiple initial guess approach was proposed. The proposed data-driven surrogate-based optimization framework, that include outlier detection, machine learning model development, machine learning model optimization and data-driven surrogate-based process optimization can act as computer-aided software that can be applied to a wide range of applications.

Optimization results show that the proposed surrogate-based optimization framework is effective and can lead to a reduction in cost and energy consumption. A significant reduction in steam flowrate while maintaining high product flowrate was obtained for both plants by manipulating the operating conditions using the proposed optimization model (around 20% and for both plants). Future work should continue exploring different optimization algorithm approaches that can start from a good location in feasible space (has built-in feature to select good initial guess). Moreover,

a multi-objective optimization model can be developed by modifying the current optimization model. Future work can also include the development of a dynamic data-driven model based on time series observations, which can predict dynamic nonlinear relationships between the input variables and output variables. For example, dense plant time data with a high sampling frequency can be used to develop recurrent neural networks, where forecasting can be performed. Improving training data in term of quality and quantity results in more reliable and accurate predictions, as the performance of machine learning models is entirely determined by data quality and quantity. For example, use more data that covers as much ground within the design space as possible. It is worth noting that the optimization algorithm used in this case study is a numerical optimization method which is based on objective function and constraints approximation near solution candidate. As a result of that, reaching and defining the global optimum might be challenging and require extensive effort and investigation. It was also demonstrated that the numerical optimization used is very sensitive to the initial guess. Therefore, there is still room for improvement in the selection of the initial guess for the proposed surrogate-based optimization framework. Further research can be conducted to explore different numerical methods that are more effective to work with machine learning models and can find global or near global optimum.

## Chapter 6 Conclusions and Future Work

All research goals of the proposed work were successfully accomplished. This dissertation presented different data-driven solution frameworks that were successfully implemented in optimally planning energy infrastructure and operating chemical process which can deal with uncertainties, multiscale modelling, and unit equation complexity.

As discussed in Chapter 3, a reduced size stochastic scenarios generation framework that was efficient in representing a wide spectrum of the most probable scenarios and leading to an inexpensive computational problem, was developed. Furthermore, deterministic and stochastic data-driven power generation planning models were developed. The deterministic approach was solved based on a single population parameter (i.e., mean) from the data, while the stochastic data-driven approach was based on more detailed information from the data without explicitly knowing its distribution. In the stochastic data-driven approach, a clustering algorithm (k-means) was applied to generate reduced size scenarios (i.e., clusters) from the historically available data and thus, lead to an inexpensive computational problem. It was demonstrated that power generation design and operation under stochastic approach is more practical than designing under both: extreme case (i.e., worst-case scenario), and deterministic case. From the assessment of applying the proposed data-driven approach on different types of power generation planning model, it was concluded that reducing the uncertain data size by implementing k-mean clustering method, is an effective tool to tackle the computational tractability of considering many stochastic scenarios. In general, the proposed method does not require a full understanding of the data's behaviour. At the same time, it presents a simple framework that can offer acceptable results. Therefore, the data-driven stochastic method is a trade-off between computational effort and data accuracy.

In chapter 4, a clustering approach to reduce multiple attribute demand data size by representing the yearly days by "typical" days representatives was developed. Heuristic size reduction approach derived from the general formulation clustering approach was utilized to cluster the given demand data in less computational time. Results revealed that the heuristic approach can reduce the clustering running time by 2 orders of magnitude than the general clustering approach without compromising clustering accuracy. The clustering approach was applied to stochastic energy hub system to reduce the complexity of the model with reasonable accuracy. For applications that do not need sequencing, it is advantageous to apply normal clustering to minimize computational

effort and deal with large scale models. Although, there was no clear relationship between attributes weight factors and the model objective function, it was found that giving more weight to the attribute with higher degree of fluctuation can enhance the objective function and multiscale decision variables (become closer to the full-size energy hub model solution). In all clustered cases, results showed that average relative errors of the reduced size energy hub objective functions with respect to the full-size mode, were not exceeding 11%. It was proven from this study that the solutions (design and operational decisions) from solving energy hub model with reduced size demand, are very close to the solution of full-size energy hub model, at much less computational time (less by 2 to 3 order of magnitude). Assessment on the stochastic formulation showed that considering wind uncertainties in design and operation of energy hub model had positive effects. A comprehensive development of machine learning models and surrogate-based optimization framework for 2 condensate stabilizer units was presented in Chapter 5. Data-driven model development that include outlier detection and model hyperparameters tuning was performed. This case study showcases that developed models can perform reliable and accurate predictions. And consequently, they can be used as a convenient alternative to the process unit operation models within process optimization. Optimization approach for the condensate stabilizers were developed based on TRCM, in which machine learning prediction models were integrated. Results obtained from solving the optimization problem showcase that the proposed method is a useful data-driven tool that can help the gas industry to simultaneously achieve process efficiency, profitability, and safety. The proposed approach is general and can be applied to a wide range of chemical processes based on plant or simulation data.

Further research and investigation can be conducted based on the case studies presented in Chapter 3, 4 and 5 as follows:

The power generation planning model developed in Chapter 3 can be further expanded to include different types of generation units, powered by different types of fuel, and investigate the effect on design and operational decisions. Additionally, the integration of solar energy as a source of energy be explored, where the proposed approach can also be used to model the uncertainty behaviour of solar energy. An energy storage system can be combined, and the behaviour of the system under different realization of intermittent renewable energy and demand can be investigated. The stochastic power generation planning model can be extended to include a carbon



capturing unit which can be modelled using either mass and energy balances or supervised machine learning models.

Future works related to Chapter 4 can include the application of the proposed clustering approach to different multiscale planning problem. The stochastic energy hub planning model can be extended to include capacity expansion planning decisions, while satisfying multiple attributes demand. In detail, forecasting techniques can be employed to forecast the future demands, clustering approach will be applied to reduce the size of these multiple attributes demand, in which they can be used as an input to the energy hub capacity expansion planning model. In another example of future work, the multiscale clustering approach can be applied to superstructure modelling approach for designing new chemical or power plants. Furthermore, the clustering approach that was proposed in Chapter 3 to generate stochastic scenarios, can be used in this case study to generate reduced size wind speed scenarios.

For Chapter 5, a case study exploring different optimization algorithm approaches that can start from a good location in feasible region (good starting point) can be investigated. Additionally, a multi-objective optimization model can be developed by adjusting the current optimization model. Future work can also include the development of dynamic data-driven models based on time series observations which predict dynamic plant behaviour. Improving training data in term of quality, results in more reliable and accurate predictions, because the performance of machine learning models is entirely determined by data quality and quantity. There is still room for improving the selection of initial guess approach for the proposed surrogate-based optimization framework. Further research can be conducted to explore different numerical methods that are more effective to work with machine learning models and can find global or near global optimum.

## References

- [1] L. G. Papageorgiou, "Supply chain optimisation for the process industries: Advances and opportunities," *Comput. Chem. Eng.*, vol. 33, no. 12, pp. 1931–1938, 2009, doi: <https://doi.org/10.1016/j.compchemeng.2009.06.014>.
- [2] F. Alhameli, "Multiscale modeling in mathematical programming: Application of Clustering," University of Waterloo, Waterloo, Ontario, Canada, 2017.
- [3] D. Bertsimas and A. Thiele, "Robust and Data-Driven Optimization: Modern Decision Making Under Uncertainty," in *Models, Methods, and Applications for Innovative Decision Making*, INFORMS, 2006, pp. 95–122.
- [4] E. Weinan, "Principles of Multiscale Modeling," Princeton, USA, 2011.
- [5] C. A. Henao and C. T. Maravelias, "Surrogate-based superstructure optimization framework," *AIChE J.*, vol. 57, no. 5, pp. 1216–1232, 2011, doi: [doi:10.1002/aic.12341](https://doi.org/10.1002/aic.12341).
- [6] R. Gani, I. Cameron, and M. Georgiadis, "Process Systems Engineering, 2. Modeling and Simulation," in *Ullmann's Encyclopedia of Industrial Chemistry*, .
- [7] "Aspen Plus 11.1." Aspen Technology, Inc. - USA, 2001, doi: [citeulike-article-id:1474197](https://doi.org/10.1002/aic.12341).
- [8] "ProMax®." Bryan Research & Engineering, Texas, USA.
- [9] "gProms." Process System Enterprise (PSE).
- [10] B. Mehdizadeh and K. Movagharnejad, "A comparative study between LS-SVM method and semi empirical equations for modeling the solubility of different solutes in supercritical carbon dioxide," *Chem. Eng. Res. Des.*, vol. 89, no. 11, pp. 2420–2427, 2011, doi: <https://doi.org/10.1016/j.cherd.2011.03.017>.
- [11] R. Haghbakhsh, H. Adib, P. Keshavarz, M. Koolivand, and S. Keshtkari, "Development of an artificial neural network model for the prediction of hydrocarbon density at high-pressure, high-temperature conditions," *Thermochim. Acta*, vol. 551, pp. 124–130, 2013, doi: <https://doi.org/10.1016/j.tca.2012.10.022>.
- [12] H. Adib, A. Sabet, A. Naderifar, M. Adib, and M. Ebrahimzadeh, "Evolving a prediction model based on machine learning approach for hydrogen sulfide removal from sour condensate of south pars natural gas processing plant," *J. Nat. Gas Sci. Eng.*, vol. 27, pp. 74–81, 2015, doi: <https://doi.org/10.1016/j.jngse.2015.08.012>.
- [13] S. J. Qin, "Process data analytics in the era of big data," *AIChE J.*, vol. 60, no. 9, pp.

- 3092–3100, 2014, doi: <https://doi.org/10.1002/aic.14523>.
- [14] J. Li *et al.*, “Data-driven mathematical modeling and global optimization framework for entire petrochemical planning operations,” *AIChE J.*, vol. 62, no. 9, pp. 3020–3040, 2016, doi: <https://doi.org/10.1002/aic.15220>.
- [15] “General Algebraic Modeling System (GAMS 24.5).” GAMS Development Corporation, Washington, DC, USA, 2015.
- [16] Y. Chen, T. A. Adams, and P. I. Barton, “Optimal Design and Operation of Static Energy Polygeneration Systems,” *Ind. Eng. Chem. Res.*, vol. 50, no. 9, pp. 5099–5113, 2011, doi: [10.1021/ie101568v](https://doi.org/10.1021/ie101568v).
- [17] B. A. McCarl and J. Apland, “Validation Of Linear Programming Models,” *South. J. Agric. Econ.*, vol. 18, no. 2, pp. 1–10, Dec. 1986, doi: [10.22004/ag.econ.29773](https://doi.org/10.22004/ag.econ.29773).
- [18] P. Kall and S. W. Wallace, *Stochastic programming*. Wiley, 1994.
- [19] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory, Second Edition*. Society for Industrial and Applied Mathematics, 2014.
- [20] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, 2nd ed. Springer Publishing Company, Incorporated, 2011.
- [21] X. Li, E. Armagan, A. Tomsgard, and P. I. Barton, “Stochastic pooling problem for natural gas production network design and operation under uncertainty,” *AIChE J.*, vol. 57, no. 8, pp. 2120–2135, 2011, doi: [doi:10.1002/aic.12419](https://doi.org/10.1002/aic.12419).
- [22] P. Virtanen *et al.*, “SciPy 1.0: fundamental algorithms for scientific computing in Python,” *Nat. Methods*, vol. 17, no. 3, pp. 261–272, 2020, doi: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).
- [23] V. Dhar, “Data science and prediction,” *Commun. ACM*, vol. 56, no. 12, pp. 64–73, 2013, doi: [10.1145/2500499](https://doi.org/10.1145/2500499).
- [24] C. Hayashi, “What is Data Science ? Fundamental Concepts and a Heuristic Example,” in *Data Science, Classification, and Related Methods*, 1998, pp. 40–51.
- [25] U. C. B. S. of Information, “What is Data Science ?” <https://datascience.berkeley.edu/about/what-is-data-science/>.
- [26] P. B. and S. T. Patil, “A Comparative Study of Data Analysis Techniques,” *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 3, no. 2, p. 7, 2014.
- [27] P. Gong and B. Selene Xia, “Review of business intelligence through data analysis,”

- Benchmarking An Int. J.*, vol. 21, no. 2, pp. 300–311, 2014, doi: 10.1108/BIJ-08-2012-0050.
- [28] D. on the Analysis of Massive, T. on Mathematical Sciences, T. on Engineering, P. S. Council, and R. National, *Frontiers in Massive Data Analysis*. The National Academies Press, 2013.
- [29] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011, doi: 10.1561/22000000016.
- [30] K. P. Murphy, *Machine learning: a probabilistic perspective*. Cambridge, MA, 2012.
- [31] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.
- [32] J. Davis, T. Edgar, J. Porter, J. Bernaden, and M. Sarli, “Smart manufacturing, manufacturing intelligence and demand-dynamic performance,” *Comput. Chem. Eng.*, vol. 47, pp. 145–156, 2012, doi: <https://doi.org/10.1016/j.compchemeng.2012.06.037>.
- [33] J. H. Lee, J. Shin, and M. J. Realf, “Machine learning: Overview of the recent progresses and implications for the process systems engineering field,” *Comput. Chem. Eng.*, vol. 114, pp. 111–121, Jun. 2018, doi: 10.1016/j.compchemeng.2017.10.008.
- [34] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science (80-. )*, vol. 349, no. 6245, pp. 255–260, 2015, doi: 10.1126/science.aaa8415.
- [35] G. Shmueli, P. C. Bruce, I. Yahav, N. R. Patel, and K. C. Lichtendahl, *Data mining for business analytics : concepts, techniques, and applications in R*. 2018.
- [36] D. Arthur and S. Vassilvitskii, “k-means++: the advantages of careful seeding,” *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, New Orleans, Louisiana, pp. 1027–1035, 2007.
- [37] F. Pedregosa *et al.*, “Scikit-Learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. null, pp. 2825–2830, Nov. 2011.
- [38] L. Buitinck *et al.*, “{API} design for machine learning software: experiences from the scikit-learn project,” in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.
- [39] D. W. Gareth James Trevor Hastie, Robert Tibshirani, *An introduction to statistical*

- learning : with applications in R*. New York : Springer, [2013] ©2013.
- [40] I. Goodfellow, B. Yoshua, and C. Aaron, “Deep Learning,” *Deep Learn.*, 2016, doi: 10.1016/B978-0-12-391420-0.09987-X.
- [41] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. USA: Cambridge University Press, 2007.
- [42] R. Rifkin, “Regularized Least Squares,” no. February, 2006.
- [43] V. N. Vapnik, “Constructing Learning Algorithms BT - The Nature of Statistical Learning Theory,” V. N. Vapnik, Ed. New York, NY: Springer New York, 1995, pp. 119–166.
- [44] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, “Support vector machines,” *IEEE Intell. Syst. their Appl.*, vol. 13, no. 4, pp. 18–28, 1998, doi: 10.1109/5254.708428.
- [45] J. Yu, “A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses,” *Comput. Chem. Eng.*, vol. 41, pp. 134–144, 2012.
- [46] T. Hofmann, B. Schölkopf, and A. J. Smola, “Kernel methods in machine learning,” *Annals of Statistics*. 2008, doi: 10.1214/009053607000000677.
- [47] B. Zhao, “Modeling pressure drop coefficient for cyclone separators: A support vector machine approach,” *Chem. Eng. Sci.*, vol. 64, no. 19, pp. 4131–4136, 2009, doi: <https://doi.org/10.1016/j.ces.2009.06.017>.
- [48] J. H. Holland, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Oxford, England: U Michigan Press, 1975.
- [49] J. R. Koza, “Survey of genetic algorithms and genetic programming,” in *Proceedings of WESCON'95*, 1995, pp. 589-, doi: 10.1109/WESCON.1995.485447.
- [50] D. E. Goldberg and J. H. Holland, “Genetic Algorithms and Machine Learning,” *Mach. Learn.*, vol. 3, no. 2, pp. 95–99, 1988, doi: 10.1023/A:1022602019183.
- [51] N. M. Kazerooni, H. Adib, A. Sabet, M. A. Adhami, and M. Adib, “Toward an intelligent approach for H2S content and vapor pressure of sour condensate of south pars natural gas processing plant,” *J. Nat. Gas Sci. Eng.*, vol. 28, no. C, pp. 365–371, 2016, doi:

- 10.1016/j.jngse.2015.12.006.
- [52] J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Eds., *Multiobjective Optimization: Interactive and evolutionary approaches*, vol. 5252. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
  - [53] P. Izquierdo, “Condition monitoring of the cooling of variable speed drives using artificial intelligence,” Aalborg University, 2020.
  - [54] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010, pp. 807–814.
  - [55] X. Glorot, A. Bordes, and Y. Bengio, “Deep Sparse Rectifier Neural Networks,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, vol. 15, pp. 315–323, [Online]. Available: <https://proceedings.mlr.press/v15/glorot11a.html>.
  - [56] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.
  - [57] L. Bottou, “On-line learning and stochastic approximations,” 1999.
  - [58] L. Bottou, “Stochastic Gradient Learning in Neural Networks,” 1991.
  - [59] B. T. Polyak, “Some methods of speeding up the convergence of iteration methods,” *USSR Comput. Math. Math. Phys.*, vol. 4, no. 5, pp. 1–17, 1964, doi: 10.1016/0041-5553(64)90137-5.
  - [60] J. Duchi, E. Hazan, and Y. Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011, doi: 10.1109/CDC.2012.6426698.
  - [61] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” pp. 1–18, 2012, doi: arXiv:1207.0580.
  - [62] D. P. Kingma and J. Ba, “Adam: {A} Method for Stochastic Optimization,” in *3rd International Conference on Learning Representations, {ICLR} 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015, [Online]. Available: <http://arxiv.org/abs/1412.6980>.

- [63] S. Ruder, “An overview of gradient descent optimization algorithms,” pp. 1–14, 2016, doi: 10.1111/j.0006-341X.1999.00591.x.
- [64] T. Tulabandhula and C. Rudin, “Robust optimization using machine learning for uncertainty sets,” *Int. Symp. Artif. Intell. Math. ISAIM 2014*, pp. 1–28, 2014.
- [65] C. Ning and F. You, “Data-driven adaptive nested robust optimization: General modeling framework and efficient computational algorithm for decision making under uncertainty,” *AIChE J.*, vol. 63, no. 9, pp. 3790–3817, 2017, doi: 10.1002/aic.15717.
- [66] C. Ning and F. You, “Data-driven stochastic robust optimization: General computational framework and algorithm leveraging machine learning for optimization under uncertainty in the big data era,” *Comput. Chem. Eng.*, vol. 111, pp. 115–133, 2018, doi: <https://doi.org/10.1016/j.compchemeng.2017.12.015>.
- [67] C. Ning and F. You, “Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods,” *Comput. Chem. Eng.*, vol. 112, pp. 190–210, 2018, doi: <https://doi.org/10.1016/j.compchemeng.2018.02.007>.
- [68] C. Ning and F. You, “Adaptive robust optimization with minimax regret criterion: Multiobjective optimization framework and computational algorithm for planning and scheduling under uncertainty,” *Comput. Chem. Eng.*, vol. 108, pp. 425–447, 2018, doi: <https://doi.org/10.1016/j.compchemeng.2017.09.026>.
- [69] C. Shang and F. You, “Distributionally robust optimization for planning and scheduling under uncertainty,” *Comput. Chem. Eng.*, vol. 110, pp. 53–68, 2018, doi: <https://doi.org/10.1016/j.compchemeng.2017.12.002>.
- [70] L. Sun and F. You, “Machine Learning and Data-Driven Techniques for the Control of Smart Power Generation Systems: An Uncertainty Handling Perspective,” *Engineering*, 2021, doi: <https://doi.org/10.1016/j.eng.2021.04.020>.
- [71] I. G. Moghaddam, M. Saniei, and E. Mashhour, “A comprehensive model for self-scheduling an energy hub to supply cooling, heating and electrical demands of a building,” *Energy*, vol. 94, pp. 157–170, Jan. 2016, doi: 10.1016/j.energy.2015.10.137.
- [72] M. Majidi, S. Nojavan, and K. Zare, “A cost-emission framework for hub energy system under demand response program,” *Energy*, vol. 134, pp. 157–166, Sep. 2017, doi: 10.1016/j.energy.2017.06.003.

- [73] S. Nojavan, M. Majidi, and K. Zare, “Optimal scheduling of heating and power hubs under economic and environment issues in the presence of peak load management,” *Energy Convers. Manag.*, vol. 156, pp. 34–44, Jan. 2018, doi: 10.1016/j.enconman.2017.11.007.
- [74] T. Ma, J. Wu, and L. Hao, “Energy flow modeling and optimal operation analysis of the micro energy grid based on energy hub,” *Energy Convers. Manag.*, vol. 133, pp. 292–306, Feb. 2017, doi: 10.1016/j.enconman.2016.12.011.
- [75] Q. Lu, S. Lü, Y. Leng, and Z. Zhang, “Optimal household energy management based on smart residential energy hub considering uncertain behaviors,” *Energy*, vol. 195, p. 117052, Mar. 2020, doi: 10.1016/j.energy.2020.117052.
- [76] M. J. Vahid-Pakdel, S. Nojavan, B. Mohammadi-ivatloo, and K. Zare, “Stochastic optimization of energy hub operation with consideration of thermal energy market and demand response,” *Energy Convers. Manag.*, vol. 145, pp. 117–128, Aug. 2017, doi: 10.1016/j.enconman.2017.04.074.
- [77] K. Palmer and M. Realff, “Metamodeling Approach to Optimization of Steady-State Flowsheet Simulations: Model Generation,” *Chem. Eng. Res. Des.*, vol. 80, no. 7, pp. 760–772, 2002, doi: <https://doi.org/10.1205/026387602320776830>.
- [78] R. Jin, W. Chen, and T. Simpson, “Comparative studies of metamodeling techniques under multiple modeling criteria,” in *8th Symposium on Multidisciplinary Analysis and Optimization*, American Institute of Aeronautics and Astronautics, 2000.
- [79] B. S. Matthew Joseph Hetzel and D. J. B. Riggs, “Refinery-wide optimization using neural network surrogate models,” Texas Tech University, 2002.
- [80] H. R. Sant Anna, A. G. Barreto, F. W. Tavares, and M. B. de Souza, “Machine learning model and optimization of a PSA unit for methane-nitrogen separation,” *Comput. Chem. Eng.*, vol. 104, pp. 377–391, 2017, doi: <https://doi.org/10.1016/j.compchemeng.2017.05.006>.
- [81] M. Alkatheri, M. Rizwan, F. Alhameli, A. Elkamel, A. Almansoori, and P. Douglas, “Data-driven power generation design and operation under demand uncertainty,” 2019.
- [82] N. V Sahinidis, “Optimization under uncertainty: state-of-the-art and opportunities,” *Comput. Chem. Eng.*, vol. 28, no. 6, pp. 971–983, 2004, doi: <https://doi.org/10.1016/j.compchemeng.2003.09.017>.



- [83] I. E. Grossmann, R. M. Apap, B. A. Calfa, P. García-Herreros, and Q. Zhang, “Recent advances in mathematical programming techniques for the optimization of process systems under uncertainty,” *Comput. Chem. Eng.*, vol. 91, pp. 3–14, 2016, doi: <https://doi.org/10.1016/j.compchemeng.2016.03.002>.
- [84] M. G. Ierapetritou, E. N. Pistikopoulos, and C. A. Floudas, “Operational planning under uncertainty,” *Comput. Chem. Eng.*, vol. 18, pp. S553–S557, 1994, doi: [https://doi.org/10.1016/0098-1354\(94\)80090-1](https://doi.org/10.1016/0098-1354(94)80090-1).
- [85] Y. Pochet and F. Warichet, “A tighter continuous time formulation for the cyclic scheduling of a mixed plant,” *Comput. Chem. Eng.*, vol. 32, no. 11, pp. 2723–2744, 2008, doi: <https://doi.org/10.1016/j.compchemeng.2007.09.001>.
- [86] D. Wu and M. Ierapetritou, “Cyclic short-term scheduling of multiproduct batch plants using continuous-time representation,” *Comput. Chem. Eng.*, vol. 28, no. 11, pp. 2271–2286, 2004, doi: <https://doi.org/10.1016/j.compchemeng.2004.04.002>.
- [87] V. K. Tumuluru, Z. Huang, and D. H. K. Tsang, “Unit commitment problem: A new formulation and solution method,” *Int. J. Electr. Power Energy Syst.*, vol. 57, pp. 222–231, 2014, doi: <https://doi.org/10.1016/j.ijepes.2013.11.043>.
- [88] M. G. Marcovecchio, A. Q. Novais, and I. E. Grossmann, “Deterministic optimization of the thermal Unit Commitment problem: A Branch and Cut search,” *Comput. Chem. Eng.*, vol. 67, pp. 53–68, 2014, doi: <https://doi.org/10.1016/j.compchemeng.2014.03.009>.
- [89] “Hourly Ontario and Market Demands 2018.” <http://www.ieso.ca/Pages/Power-Data/Data-Directory.aspx>.
- [90] V. N. Dieu and W. Ongsakul, “Augmented Lagrange Hopfield network based Lagrangian relaxation for unit commitment,” *Int. J. Electr. Power Energy Syst.*, vol. 33, no. 3, pp. 522–530, 2011, doi: <https://doi.org/10.1016/j.ijepes.2010.12.004>.
- [91] D. N. Simopoulos, S. D. Kavatza, and C. D. Vournas, “Unit Commitment by an Enhanced Simulated Annealing Algorithm,” in *2006 IEEE PES Power Systems Conference and Exposition*, 2006, pp. 193–201, doi: 10.1109/PSCE.2006.296296.
- [92] A. da Rosa, “Chapter 15 - Wind Energy,” in *Fundamentals of Renewable Energy Processes (Third Edition)*, A. da Rosa, Ed. Boston: Academic Press, 2013, pp. 685–763.
- [93] L. B. & S. Matysik, “Vestas V90-1.8.” <https://en.wind-turbine-models.com/turbines/971-vestas-v90-1.8>.

- [94] Kriemann *et al.*, “Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change,” IPCC, Cambridge, United Kingdom and New York, NY, USA., 2014.
- [95] EIA and U. S. D. of Energy, “Capital Cost Estimates for Utility Scale Electricity Generating Plants,” Washington, DC, USA, 2016.
- [96] M. Elf, C. Gutwenger, J. Michael, #252, nger, and G. Rinaldi, “Branch-and-Cut Algorithms for Combinatorial Optimization and Their Implementation in ABACUS,” *Computational Combinatorial Optimization, Optimal or Provably Near-Optimal Solutions [based on a Spring School]*. Springer-Verlag, pp. 157–222, 2001.
- [97] F. Alhameli, A. Elkamel, A. Betancourt-Torcat, and A. Almansoori, “A mixed-integer programming approach for clustering demand data for multiscale mathematical programming applications,” *AIChE J.*, 2019, doi: 10.1002/aic.16578.
- [98] A. Schwele, J. Kazempour, and P. Pinson, “Do unit commitment constraints affect generation expansion planning? A scalable stochastic model,” *Energy Syst.*, vol. 11, pp. 247–282, May 2020, doi: 10.1007/s12667-018-00321-z.
- [99] U. Mukherjee, A. Maroufmashat, A. Narayan, A. Elkamel, and M. Fowler, “A Stochastic Programming Approach for the Planning and Operation of a Power to Gas Energy Hub with Multiple Energy Recovery Pathways,” *Energies*, vol. 10, no. 7, 2017, doi: 10.3390/en10070868.
- [100] N. H. Jabarullah, M. S. Shabbir, M. Abbas, A. F. Siddiqi, and S. Berti, “Using random inquiry optimization method for provision of heat and cooling demand in hub systems for smart buildings,” *Sustain. Cities Soc.*, vol. 47, p. 101475, May 2019, doi: 10.1016/j.scs.2019.101475.
- [101] A. Maroufmashat, S. Sattari, R. Roshandel, M. Fowler, and A. Elkamel, “Multi-objective Optimization for Design and Operation of Distributed Energy Systems through the Multi-energy Hub Network Approach,” *Ind. Eng. Chem. Res.*, vol. 55, no. 33, pp. 8950–8966, Aug. 2016, doi: 10.1021/acs.iecr.6b01264.
- [102] M. Geidl and G. Andersson, “Optimal Power Flow of Multiple Energy Carriers,” *IEEE Trans. Power Syst.*, vol. 22, no. 1, pp. 145–155, Feb. 2007, doi: 10.1109/TPWRS.2006.888988.
- [103] A. Turk, Q. Wu, M. Zhang, and J. Østergaard, “Day-ahead stochastic scheduling of

- integrated multi-energy system for flexibility synergy and uncertainty balancing,” *Energy*, vol. 196, p. 117130, Apr. 2020, doi: 10.1016/j.energy.2020.117130.
- [104] T. Liu, D. Zhang, S. Wang, and T. Wu, “Standardized modelling and economic optimization of multi-carrier energy systems considering energy storage and demand response,” *Energy Convers. Manag.*, vol. 182, pp. 126–142, Feb. 2019, doi: 10.1016/j.enconman.2018.12.073.
- [105] L. Kotzur, P. Markewitz, M. Robinius, and D. Stolten, “Impact of different time series aggregation methods on optimal energy system design,” *Renew. Energy*, vol. 117, pp. 474–487, Mar. 2018, doi: 10.1016/j.renene.2017.10.017.
- [106] A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010, doi: 10.1016/j.patrec.2009.09.011.
- [107] M. Rao, “Cluster analysis and mathematical programming,” *J. Am. Stat. Assoc.*, vol. 66, no. 335, pp. 622–626, 1971, doi: 10.1080/01621459.1971.10482319.
- [108] B. Sağlam, F. S. Salman, S. Sayın, and M. Türkay, “A mixed-integer programming approach to the clustering problem with an application in customer segmentation,” *Eur. J. Oper. Res.*, vol. 173, no. 3, pp. 866–879, Sep. 2006, doi: 10.1016/j.ejor.2005.04.048.
- [109] P. Balachandra and V. Chandru, “Modelling electricity demand with representative load curves,” *Energy*, vol. 24, no. 3, pp. 219–230, 1999, doi: 10.1016/S0360-5442(98)00096-6.
- [110] S. Fazlollahi, G. Becker, and F. Maréchal, “Multi-objectives, multi-period optimization of district energy systems: III. Distribution networks,” *Comput. Chem. Eng.*, vol. 66, pp. 82–97, Jun. 2014, doi: 10.1016/j.compchemeng.2014.02.018.
- [111] P. Balachandra and V. Chandru, “Supply demand matching in resource constrained electricity systems,” *Energy Convers. Manag.*, vol. 44, no. 3, pp. 411–437, 2003, doi: 10.1016/S0196-8904(02)00058-4.
- [112] N. E. Koltsaklis, G. M. Kopanos, and M. C. Georgiadis, “Design and Operational Planning of Energy Networks Based on Combined Heat and Power Units,” *Ind. Eng. Chem. Res.*, vol. 53, no. 44, pp. 16905–16923, Nov. 2014, doi: 10.1021/ie404165c.
- [113] A. Maroufmashat, M. Fowler, S. Sattari Khavas, A. Elkamel, R. Roshandel, and A. Hajimiragha, “Mixed integer linear programming based approach for optimal planning and operation of a smart urban energy network to support the hydrogen economy,” *Int. J. Hydrogen Energy*, vol. 41, no. 19, pp. 7700–7716, May 2016, doi:

- 10.1016/j.ijhydene.2015.08.038.
- [114] H. Wang, H. Zhang, C. Gu, and F. Li, “Optimal design and operation of CHPs and energy hub with multi objectives for a local energy system,” in *Energy Procedia*, Dec. 2017, vol. 142, pp. 1615–1621, doi: 10.1016/j.egypro.2017.12.539.
- [115] T. Zhang, M. Wang, P. Wang, J. Gu, W. Zheng, and Y. Dong, “Bi-stage stochastic model for optimal capacity and electric cooling ratio of CCHPs—a case study for a hotel,” *Energy Build.*, vol. 194, pp. 113–122, Jul. 2019, doi: 10.1016/j.enbuild.2019.04.004.
- [116] J. Faraji, H. Hashemi-Dezaki, and A. Ketabi, “Stochastic operation and scheduling of energy hub considering renewable energy sources’ uncertainty and N-1 contingency,” *Sustain. Cities Soc.*, vol. 65, p. 102578, Feb. 2021, doi: 10.1016/j.scs.2020.102578.
- [117] F. Alhameli, M. Alkatheri, A. Betancourt-Torcat, A. Elkamel, and A. Almansoori, *Using Big Data Analytics to Reduce the size of High Dimensional Attributes for Multiscale Decision Making: Applications to Energy Hub Demand Data*. 2020.
- [118] F. Alhameli, A. Ahmadian, and A. Elkamel, “Multiscale Decision-Making for Enterprise-Wide Operations Incorporating Clustering of High-Dimensional Attributes and Big Data Analytics: Applications to Energy Hub,” *Energies*, vol. 14, no. 20, 2021, doi: 10.3390/en14206682.
- [119] S. Bektaş and Y. Şişman, “The comparison of L11 and L22-norm minimization methods,” *Int. J. Phys.*, vol. 5, no. 11, pp. 1721–1727, 2010.
- [120] K. Sabo, “Center-based l1?clustering method,” *Int. J. Appl. Math. Comput. Sci.*, vol. 24, no. 1, pp. 151–163, Jan. 2014, doi: 10.2478/amcs-2014-0012.
- [121] Q. Lyu, Z. Lin, Y. She, and C. Zhang, “A comparison of typical ?p minimization algorithms,” *Neurocomputing*, vol. 119, pp. 413–424, Nov. 2013, doi: 10.1016/j.neucom.2013.03.017.
- [122] R. Green, I. Staffell, and N. Vasilakos, “Divide and Conquer? k-Means Clustering of Demand Data Allows Rapid and Accurate Simulations of the British Electricity System,” *IEEE Trans. Eng. Manag.*, vol. 61, no. 2, pp. 251–260, May 2014, doi: 10.1109/TEM.2013.2284386.
- [123] C. Chelmiss, J. Kolte, and V. K. Prasanna, “Big data analytics for demand response: Clustering over space and time,” in *2015 IEEE International Conference on Big Data (Big Data)*, Oct. 2015, pp. 2223–2232, doi: 10.1109/BigData.2015.7364011.

- [124] H. D. Vinod, "Integer Programming and the Theory of Grouping," *J. Am. Stat. Assoc.*, vol. 64, no. 326, pp. 506–519, 1969, doi: 10.2307/2283635.
- [125] O. L. Mangasarian, "Absolute value equation solution via dual complementarity," *Optim. Lett.*, vol. 7, no. 4, pp. 625–630, May 2013, doi: 10.1007/s11590-012-0469-5.
- [126] H. Mirzaesmaeeli, A. Elkamel, P. L. Douglas, E. Croiset, and M. Gupta, "A multi-period optimization model for energy planning with CO(2) emission consideration.," *J. Environ. Manage.*, vol. 91, no. 5, pp. 1063–70, May 2010, doi: 10.1016/j.jenvman.2009.11.009.
- [127] R. Xu and D. C. Wunsch, *Clustering*, vol. 1. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2008.
- [128] F. Üney and M. Türkay, "A mixed-integer programming approach to multi-class data classification problem," *Eur. J. Oper. Res.*, vol. 173, no. 3, pp. 910–920, Sep. 2006, doi: 10.1016/j.ejor.2005.04.049.
- [129] A. Maroufmashat *et al.*, "Modeling and optimization of a network of energy hubs to improve economic and emission considerations," *Energy*, vol. 93, pp. 2546–2558, 2015, doi: 10.1016/j.energy.2015.10.079.
- [130] M. Bakker, H. van Duist, K. van Schagen, J. Vreeburg, and L. Rietveld, "Improving the Performance of Water Demand Forecasting Models by Using Weather Input," *Procedia Eng.*, vol. 70, pp. 93–102, 2014, doi: 10.1016/j.proeng.2014.02.012.
- [131] A. da Rosa, "Wind Energy," in *Fundamentals of Renewable Energy Processes*, Elsevier, 2013, pp. 685–763.
- [132] M. Sengupta, Y. Xie, A. Lopez, A. Habte, G. Maclaurin, and J. Shelby, "The National Solar Radiation Data Base (NSRDB)," *Renewable and Sustainable Energy Reviews*, vol. 89. Elsevier Ltd, pp. 51–60, Jun. 01, 2018, doi: 10.1016/j.rser.2018.03.003.
- [133] J. Stander, "The specification of a small commercial wind energy conversion system for the South African Antarctic Research Base SANAE IV," 2008.
- [134] Battelle Memorial Institute, "Manufacturing Cost Analysis: 100kW and 250 kW Fuel Cell Systems for Primary Power and Combined Heat and Power Applications," *U.S. Dep. Energy, Fuel Cell Technol. Off.*, vol. 98, no. January, p. 293, 2017, Accessed: Mar. 22, 2021. [Online]. Available: <https://www.google.com/search?q=Manufacturing+Cost+Analysis+of+100+and+250+kW+Fuel+Cell+Systems+for+Primary+Power+and+Combined+Heat+and+Power+Applicatio>

ns&rlz=1C1CHBF\_enCA809CA809&sxsrf=ALeKk00L6xdO1sXtYTjfr5uEXqEKTq-  
loQ%3A1616464054196&ei=tkhZYJe3C7uu5No.

- [135] S. Lee and I. E. Grossmann, “New algorithms for nonlinear generalized disjunctive programming,” *Comput. Chem. Eng.*, vol. 24, no. 9–10, pp. 2125–2141, Oct. 2000, doi: 10.1016/S0098-1354(00)00581-0.
- [136] M. Scott and J. Stake, “Growing Condensates Require Optimized Designs for Gathering, Processing,” 2013. [Online]. Available: <https://www.aogr.com/magazine/editors-choice/growing-condensates-require-optimized-designs-for-gathering-processing>.
- [137] M. Rizwan, M. Alkatheri, F. Alhameli, A. Elkamel, and A. Almansoori, “Big Data and Machine Learning Based Approach to Gas Processing: A Case of Condensate Stabilization,” in *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2019, pp. 343–344, [Online]. Available: <http://ieomsociety.org/toronto2019/papers/83.pdf>.
- [138] M. Koolivand Salooki, R. Abedini, H. Adib, and H. Koolivand, “Design of neural network for manipulating gas refinery sweetening regenerator column outputs,” *Sep. Purif. Technol.*, vol. 82, pp. 1–9, 2011, doi: <https://doi.org/10.1016/j.seppur.2011.07.015>.
- [139] F. Alhameli, M. Alkatheri, A. Elkamel, A. Almansoori, and P. Douglas, “Surrogate-based process optimization: A case study on simple natural gas processing plant,” in *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2020, vol. 0, no. March, pp. 1485–1494.
- [140] A. Shalaby, A. Elkamel, P. L. Douglas, Q. Zhu, and Q. P. Zheng, “A machine learning approach for modeling and optimization of a CO<sub>2</sub> post-combustion capture unit,” *Energy*, vol. 215, p. 119113, 2021, doi: <https://doi.org/10.1016/j.energy.2020.119113>.
- [141] F. E. Grubbs, “Procedures for Detecting Outlying Observations in Samples,” *Technometrics*, vol. 11, no. 1, pp. 1–21, Feb. 1969, doi: 10.1080/00401706.1969.10490657.
- [142] L. Rade and B. Westergren, “A review of: ‘BETA: Mathematics Handbook: Concepts, Theorems, Methods, Algorithms, Formulas, Graphs, Tables’ Second Edition , 1993 Lund (Sweden), Studentlitteratur SEK: 320,” *Eur. J. Eng. Educ.*, vol. 19, no. 2, p. 237, Jan. 1994, doi: 10.1080/03043799408923289.
- [143] A. Fernández, S. García, M. Galar, R. Prati, B. Krawczyk, and F. Herrera, *Learning from*

*Imbalanced Data Sets*. 2018.

- [144] M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander, *LOF: Identifying Density-Based Local Outliers.*, vol. 29. 2000.
- [145] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.
- [146] M. Amer, M. Goldstein, and S. Abdennadher, *Enhancing one-class Support Vector Machines for unsupervised anomaly detection*. 2013.
- [147] L. . Kozachenko and N. . Leonenko, “Sample estimate of the entropy of a random vector.,” *Probl. Peredachi Inf*, vol. 23, no. 2, pp. 9–16, 1987.
- [148] A. Kraskov, H. Stögbauer, and P. Grassberger, “Estimating mutual information,” *Phys. Rev. E*, vol. 69, no. 6, p. 66138, Jun. 2004, doi: 10.1103/PhysRevE.69.066138.
- [149] B. C. Ross, “Mutual Information between Discrete and Continuous Data Sets,” *PLoS One*, vol. 9, no. 2, pp. 1–5, 2014, doi: 10.1371/journal.pone.0087357.
- [150] M. Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.” [Online]. Available: [www.tensorflow.org](http://www.tensorflow.org).
- [151] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.
- [152] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.
- [153] V. Cherkassky and Y. Ma, “Practical selection of SVM parameters and noise estimation for SVM regression,” *Neural Networks*, vol. 17, no. 1, pp. 113–126, 2004, doi: [https://doi.org/10.1016/S0893-6080\(03\)00169-2](https://doi.org/10.1016/S0893-6080(03)00169-2).
- [154] K. A. Al-shayji, S. Al-wadyei, and A. Elkamel, “Modelling and optimization of a multistage flash desalination process,” *Eng. Optim.*, vol. 37, no. 6, pp. 591–607, Sep. 2005, doi: 10.1080/03052150412331335801.
- [155] M. Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.” [Online]. Available: [www.tensorflow.org](http://www.tensorflow.org).
- [156] L. Prechelt, “Automatic early stopping using cross validation: Quantifying the criteria,” *Neural Networks*, vol. 11, no. 4, pp. 761–767, 1998, doi: 10.1016/S0893-6080(98)00010-0.
- [157] N. Rahmanian, L. S. Bt Jusoh, M. Homayoonfard, K. Nasrifar, and M. Moshfeghian,

- “Simulation and optimization of a condensate stabilisation process,” *J. Nat. Gas Sci. Eng.*, vol. 32, pp. 453–464, 2016, doi: <https://doi.org/10.1016/j.jngse.2016.04.028>.
- [158] W. Sun and Y.-X. Yuan, *Optimization Theory and Methods*, 1st ed., vol. 1. Boston: Springer US, 2006.
- [159] L. Hei, “Practical techniques for nonlinear optimization,” Northwestern University, 2007.
- [160] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, “A Trust Region Algorithm for Nonlinearly Constrained Optimization,” *SIAM J. Numer. Anal.*, vol. 24, no. 5, pp. 1152–1170, Oct. 1987, doi: 10.1137/0724076.
- [161] N. I. M. Gould, S. Lucidi, M. Roma, and P. L. Toint, “Solving the Trust-Region Subproblem using the Lanczos Method,” *SIAM J. Optim.*, vol. 9, no. 2, pp. 504–525, Jan. 1999, doi: 10.1137/S1052623497322735.
- [162] M. Lalee, J. Nocedal, and T. Plantenga, “On the Implementation of an Algorithm for Large-Scale Equality Constrained Optimization,” *SIAM J. Optim.*, vol. 8, no. 3, pp. 682–706, Aug. 1998, doi: 10.1137/S1052623493262993.
- [163] R. H. Byrd, M. E. Hribar, and J. Nocedal, “An Interior Point Algorithm for Large-Scale Nonlinear Programming,” *SIAM J. Optim.*, vol. 9, no. 4, pp. 877–900, Jan. 1999, doi: 10.1137/S1052623497325107.
- [164] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020, doi: 10.1038/s41586-020-2649-2.
- [165] F. Perez and B. E. Granger, “IPython: A System for Interactive Scientific Computing,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 21–29, 2007, doi: 10.1109/MCSE.2007.53.
- [166] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, “Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints,” *ACM Trans. Math. Softw.*, vol. 10, no. 3, pp. 282–298, Aug. 1984, doi: 10.1145/1271.1276.
- [167] A. R. Conn, N. I. M. Gould, and P. L. Toint, *TRUST-REGION METHODS*. Philadelphia: SIAM, 2000.



## Appendices

### Appendix A

Following are the mathematical equation of the UC power generation planning problem presented in Section 3.4. In this case only 1 demand load ( $l$ ) and 1 wind farm ( $k$ ) are considered.

**Objective Function:**

$$\begin{aligned} \min \quad & CRF \left( x^{capg} C^{capg} + \sum_k (x_k^{capwind} C_k^{capwind}) \right) \\ & + \sum_{i,t} (c_{i,t}^{SU} + c_{i,t}^{SD} + C_i^{op} p_{g,h}^{DA}) + \sum_s prob_s \left( \sum_{i,t} C_i^{op} r_{g,h,s} + \sum_{l,t} V_l^{SH} p_{l,t,s}^{SH} \right) \end{aligned}$$

**Day-ahead Constraints:**

Maximum and minimum power generation capacity in day-ahead:

$$p_i^L * u_{i,t} \leq p_{i,t}^{DA} \leq p_i^U * u_{i,t} \quad \forall i, \forall t$$

Maximum and minimum ramp up and ramp down limits in day-ahead:

$$\begin{aligned} -RD_i &\leq (p_{i,t}^{DA} - p_i^{ini}) \leq RU_i & \forall i, t = 1 \\ -RD_i &\leq (p_{i,t}^{DA} - p_{i,t-1}^{DA}) \leq RU_i & \forall g, \forall t > 1 \end{aligned}$$

Transmission capacity in day-ahead:

$$B_{(n,m)} (\theta_{n,t}^{DA} - \theta_{m,t}^{DA}) \leq F_{n,m} \quad \forall n, \forall m, \forall t$$

Start-up cost for generators in day-ahead:

$$\begin{aligned} c_{i,t}^{su} &\geq C_i^{su} (u_{i,t} - U_i^{ini}) & \forall g, t = 1 \\ c_{i,t}^{su} &\geq C_i^{su} (u_{i,t} - u_{i,t-1}) & \forall g, \forall t > 1 \\ c_{i,t}^{su} &\geq 0 & \forall g, \forall t \end{aligned}$$

Shut-down cost for generators in day-ahead:

$$\begin{aligned} c_{i,t}^{sd} &\geq C_i^{sd} (U_i^{ini} - u_{i,t}) & \forall g, t = 1 \\ c_{i,t}^{sd} &\geq C_i^{sd} (u_{i,t-1} - u_{i,t}) & \forall g, \forall t > 1 \\ c_{i,t}^{sd} &\geq 0 & \forall g, \forall h \end{aligned}$$

Power balance of system in day-ahead:

$$\sum_i p_{i,t}^{DA} + \sum_k w_{k,t}^{DA} - \sum_l D_{l,h} = \sum_m B_{(n,m)}(\theta_{n,h}^{DA} - \theta_{m,h}^{DA}) \quad \forall n, \forall h$$

**Real time constraints:**

Power balance in real time:

$$\sum_i (p_{i,t}^{DA} + r_{i,t,s}) + \sum_k (w_{k,t}^{DA} - w_{k,t,s}^{SP}) + \sum_l (D_{l,t} - p_{l,t,s}^{SH}) = \sum_m (B_{(n,m)}(\theta_{n,h,s}^{RT} - \theta_{m,h,s}^{RT})) \quad \forall n, \forall h, \forall s$$

Maximum and minimum power generation capacity in real time:

$$p_i^L u_{i,t} \leq (p_{i,t}^{DA} + r_{i,t,s}) \leq p_i^U u_{i,t} \quad \forall i, t = 1, \forall s$$

$$p_i^L u_{i,t} \leq (p_{i,t}^{DA} + r_{i,t,s}) \leq p_i^U u_{i,t} \quad \forall i, \forall t, \forall s$$

Maximum and minimum ramp-up and ramp-down limits in real time:

$$-RD_i \leq [p_{i,t}^{DA} + r_{i,t,s} - p_i^{ini}] \leq RU_i \quad \forall i, \forall s, t = 1$$

$$-RD_i \leq [(p_{i,t}^{DA} + r_{i,t,s}) - (p_{i,(t-1)}^{DA} + r_{i,(t-1),s})] \leq RU_i \quad \forall i, \forall t > 1, \forall s$$

Installed capacity of wind farm and conventional power generating units:

$$w_{k,t}^{DA} \leq x_k^{cap,wind} W_{k,t}^{DAmax} \quad \forall t$$

Where  $W_{k,t}^{DAmax}$  is the expected value of wind power realization factor (in per unit) of farm k at time t in day and is calculated as follows:

$$W_{k,t}^{DAmax} = \sum_s prob_s(W_{k,t,s}) \quad \forall k, \forall t$$

$$u_{i,t} \leq x g_i \quad \forall i, \forall t$$

$$x g_i \leq \sum_t (u_{i,t}) \quad \forall i$$

$$x^{capg} = \sum_i x g_i * P_i^U$$

Power transmission capacity limit in real time:

$$B_{(n,m)}(\theta_{n,t,s}^{RT} - \theta_{m,t,s}^{RT}) \leq F_{n,m} \quad \forall n, \forall m, \forall t, \forall s$$

Real-time power shading and wind power spillage:

$$0 \leq w_{spk,t,s} \leq W_{k,t,s} x_{wind}^{cap}$$

$$p_{i,t,s}^{SH} \geq 0$$

Renewable energy portfolio constraint:

$$\sum_k (x_k^{capwind}) \geq REP * \left( \sum_k (x_k^{capwind} C_k^{capwind}) + x^{capg} \right)$$

Table A.1 Model parameters

Generator	$C_i^{su}$ (\$)	$C_i^{sd}$ (\$)	$P_i^U$ (MW)	$P_i^L$ (MW)	$C_i^{op}$ (\$/MWh)	$RU_i$ (MW/h)	$RD_i$ (MW/h)
1	175	10	450	90	11	90	90
2	132	10	100	10	17	20	20
3	175	10	76	2	11	20	20
4	107	10	400	0	23	200	200

## Appendix A List of Symbols

### Indices

$t$	time period
$i$	power generating units
$s$	stochastic scenarios
$n$	indices for system nodes
$k$	index for wind farms
$l$	index for loads

### Discrete variables

$u_{i,t}$	binary operational/scheduling decision variable representing the on/off status of unit $i$ at period $t$
$xg_i$	binary design decision variable representing whether unit $i$ should be installed or not

### Continuous variables

$x_k^{cap_{wind}}$	installed capacity of wind farm $k$ (MW)
$x^{cap_g}$	the total Installed capacity of all conventional units (MW)
$p_{i,t}^{DA}$	power output variable of unit $i$ at period $t$ in day-ahead stage
$r_{i,t,s}$	power adjustment of conventional unit $i$ at time $t$ under scenario $s$ (MW)
$p_{l,t,s}^{SH}$	involuntary active load shedding (load loss) of load $l$ at time $t$ under scenario $s$ (MW)
$c_{i,t}^{su}$	start-up cost variable of unit $i$ at period $t$ ;
$c_{i,t}^{sd}$	shut-down cost variable of unit $i$ at period $t$
$w_{k,t}^{DA}$	power scheduled for wind farm $k$ at time $t$ in day-ahead stage (MW)
$wsp_{k,t,s}$	wind power spillage of farm $k$ at time $t$ under scenario $s$ (MW)
$\theta_{n,t}^{DA}$	voltage angle at node $n$ at time $t$ in day-ahead stage (radian)
$\theta_{n,t,s}^{RT}$	voltage angle at node $n$ at time $t$ under scenario $s$ in the real-time stage (radian)

### Parameters

$C^{cap_g}$	capital cost conventional generating units (\$/MW)
$C_k^{cap_{wind}}$	capital cost of wind farm $k$ (\$/MW)
$C_i^{op}$	operational marginal cost of conventional unit $g$ (\$/MWh)
$N_d$	lifetime of power generation plant (years)
$Prob_s$	probability of each stochastic realization scenario $s$
$P_i^U$	upper power generating limit of unit $i$
$P_i^L$	lower power generating limit of unit $i$
$V_l^{SH}$	value of lost load for load $l$ (\$/MW)

$U_i^{ini}$	initial commitment status of conventional unit i (0/1)
$RU_i$	ramp-up rate of unit i (MW/h)
$RD_i$	ramp-down rate of unit i (MW/h)
$C_i^{su}$	start-up cost for unit i (\$)
$C_i^{sd}$	shutdown cost for unit i (\$)
$CRF$	capital recovery factor
$B_{(n,m)}$	imaginary part of the admittance of lines between two connected node n and m
$F_{n,m}$	capacity of transmission line n to m in MW and it was set to 1000MW
$W_{k,t,s}$	wind power realization factor (in per unit) of farm k at time t under scenario s

## Appendix B

This section is related to Chapter 4

Figure.B.1 and B.2 show the annual hourly heat and electricity demands of the energy hub system [129], respectively.

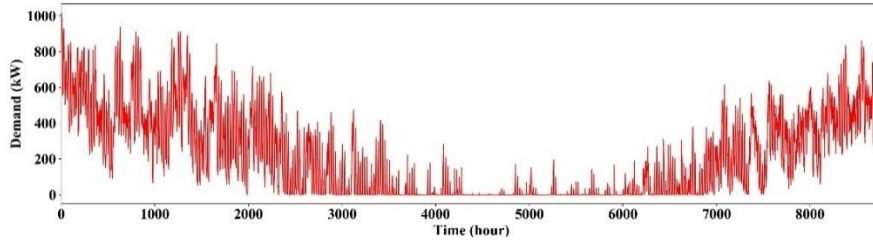


Figure B.1. Annual hourly heat demand [129]

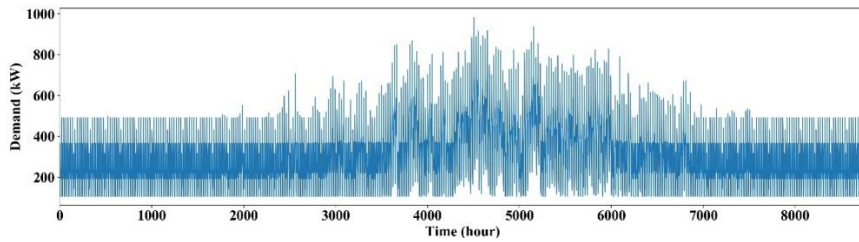


Figure B.2. Annual hourly electricity demand [129]

Raw data time-series (i.e., electricity demand and heat demand) are arranged into the candidate periods similar to what was done in Section 3.3.2 (see Figure. B.3.). Therefore, raw data of electricity demand and heat demand are reshaped into new matrix where the number of rows represent the number of days in one year (i.e., 365 days) and the number of columns represent the number of hours in one day (i.e., 24 hours). The reshaped electricity demand and heat demand profile are displayed in Figure B.4.

$$\boxed{
 \begin{array}{c}
 \text{parameter}_{8764} = \begin{pmatrix} \text{parameter}_1 \\ \text{parameter}_2 \\ \vdots \\ \text{parameter}_{8764} \end{pmatrix} \xrightarrow{\text{rearrange}} \begin{pmatrix} \text{parameter}_{1,1} & \cdots & \text{parameter}_{1,24} \\ \vdots & \ddots & \vdots \\ \text{parameter}_{366,1} & \cdots & \text{parameter}_{366,24} \end{pmatrix}
 \end{array}
 }$$

Figure B.3. Process of rearranging the dimension of wind speed and electric demand

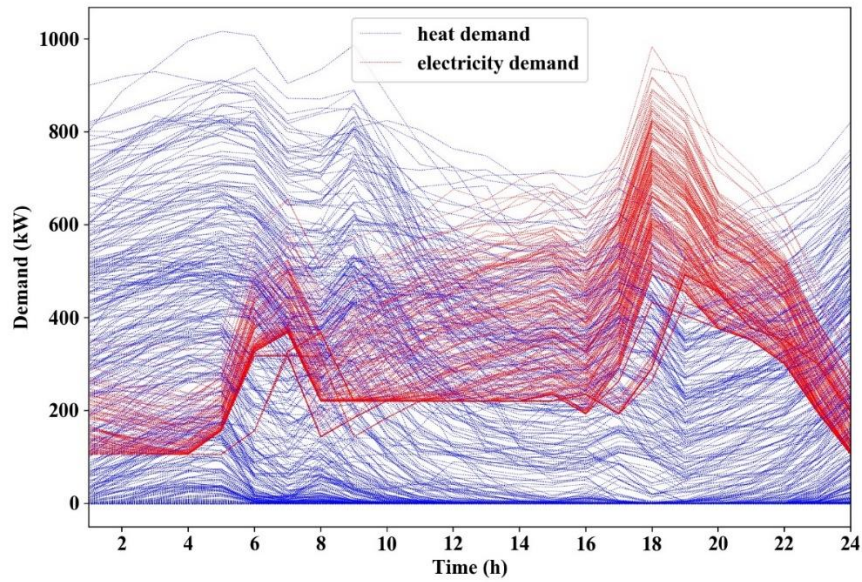


Figure B.4. Processed annual electricity (blue lines) heat demand (red lines) data

Figures B.5 to B.10 show the clusters and day assignment results of normal and sequence clustering for weight factors 1 and 8 with 4, 5, and 6 clusters.

Normal Clustering using 4 clusters

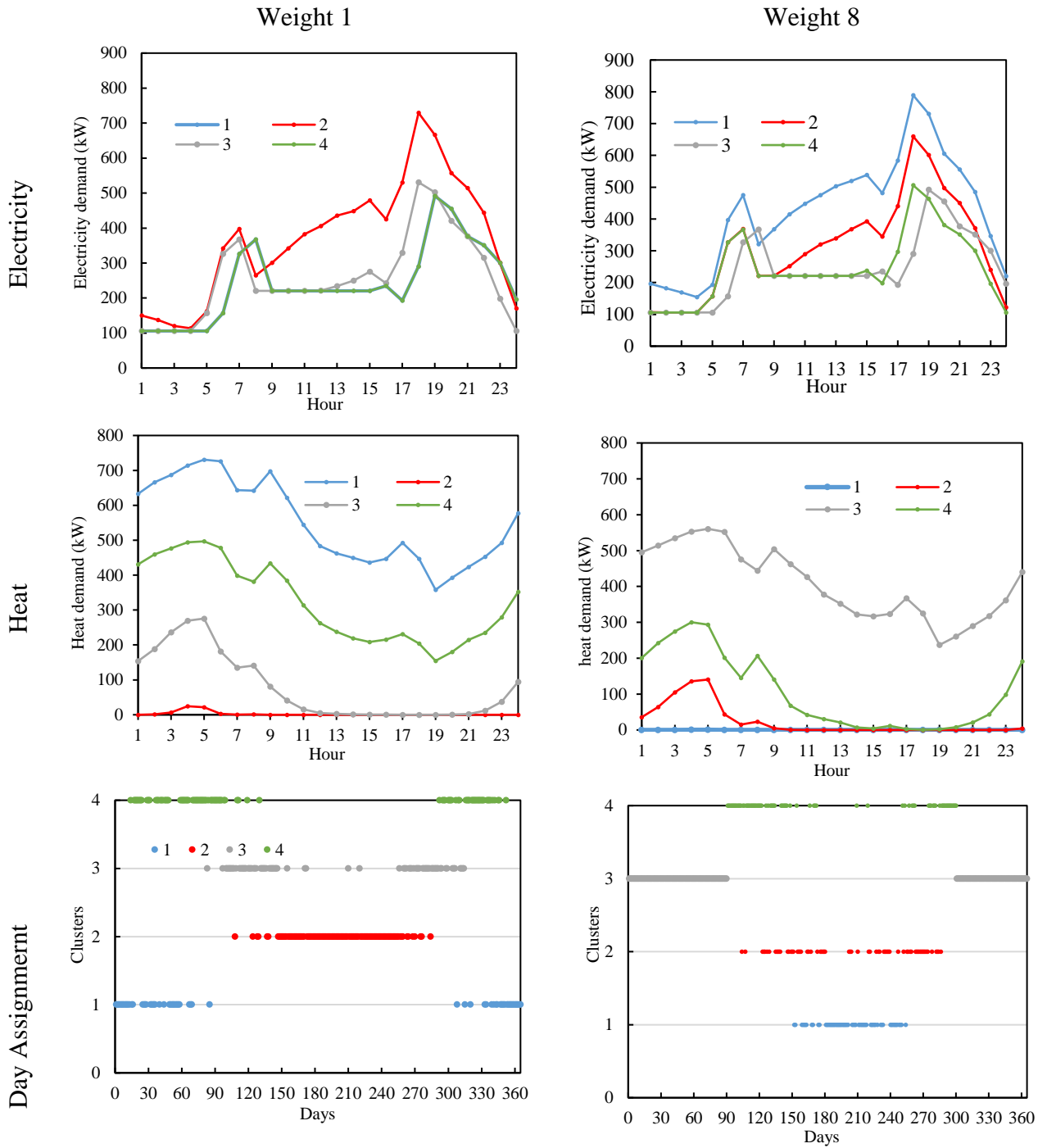


Figure B.5. heat and electricity demand cluster curves with day assignment for weight factors 1 and 8 using 4 normal clustering



Normal Clustering using 5 clusters

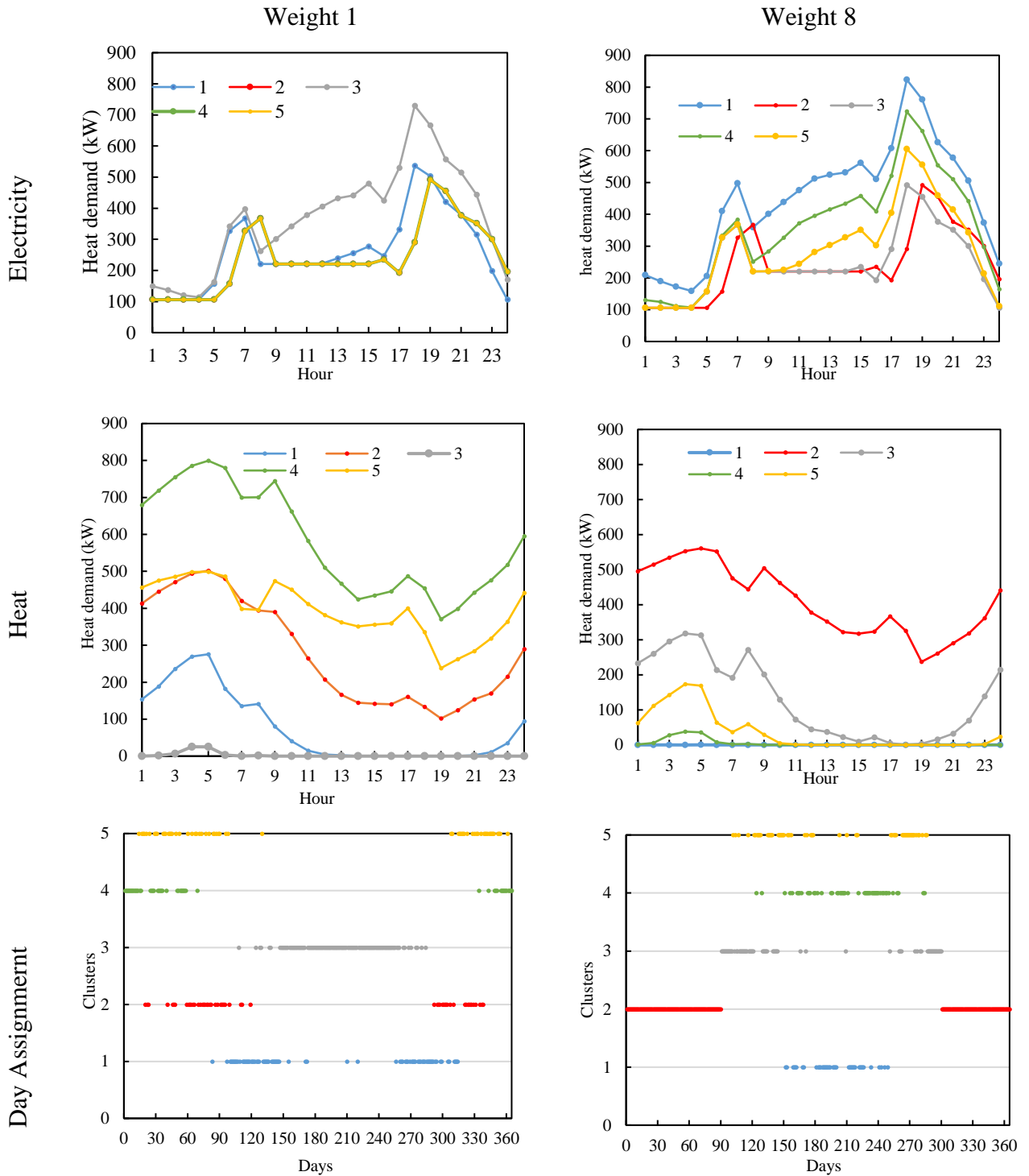


Figure. B.6. heat and electricity demand cluster curves with day assignment for weight factors 1 and 8 using 5 normal clustering

Normal Clustering using 6 clusters

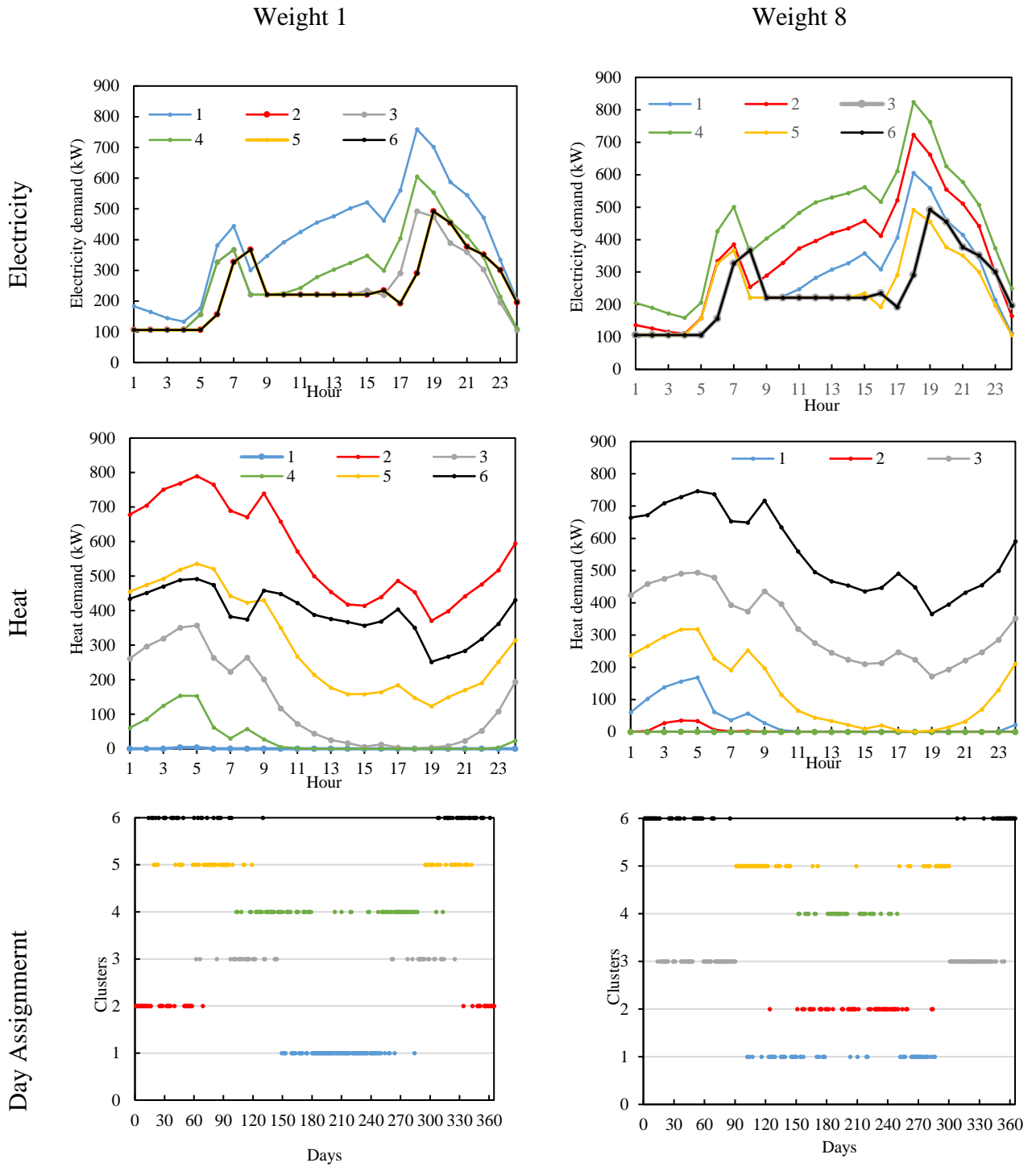


Figure B.7. heat and electricity demand cluster curves with day assignment for weight factors 1 and 8 using 6 normal clustering

Sequence Clustering using 4 clusters

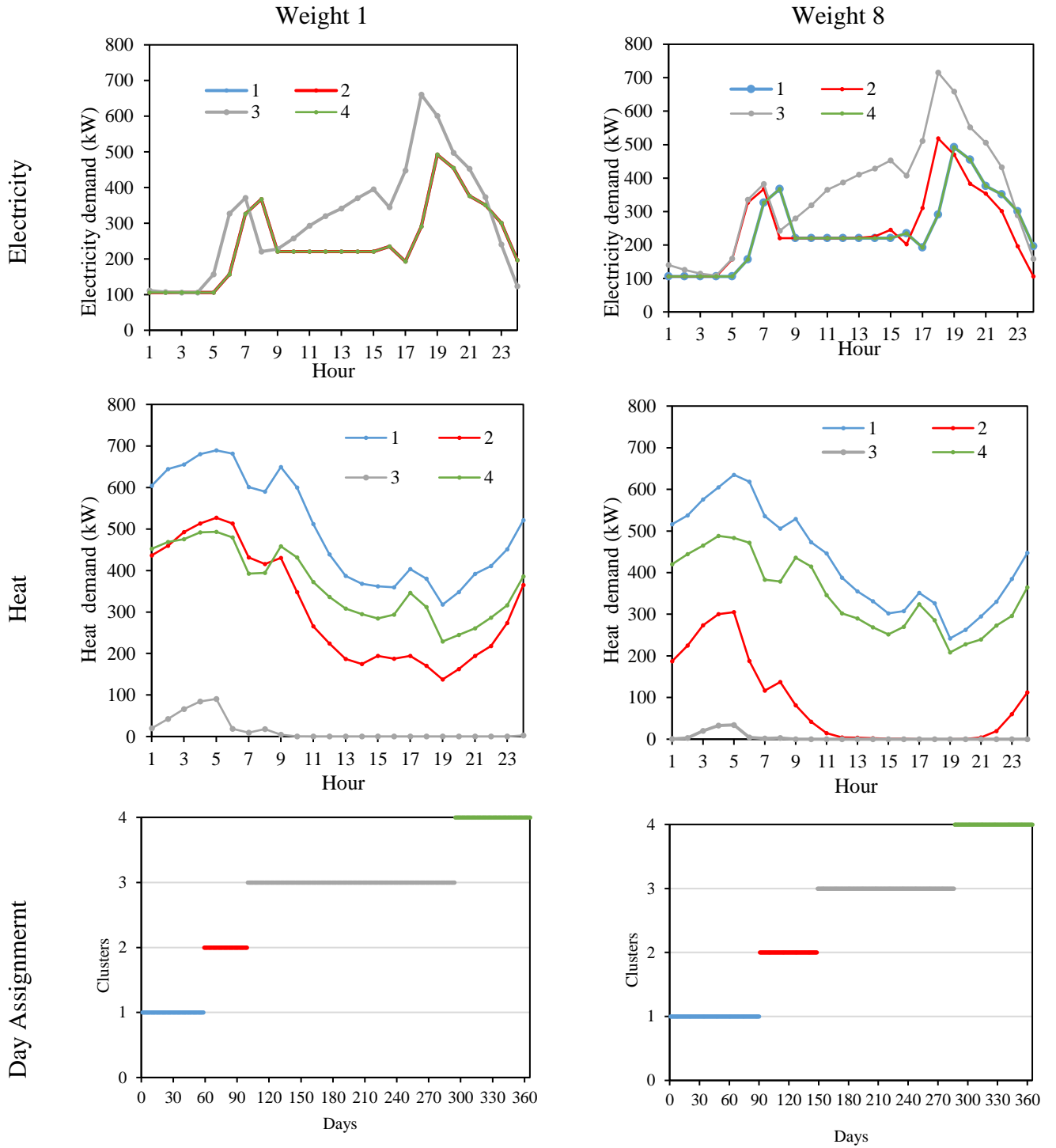


Figure B.8. heat and electricity demand cluster curves with day assignment for weight factors 1 and 8 using 4 sequence clustering

### Sequence Clustering using 5 clusters

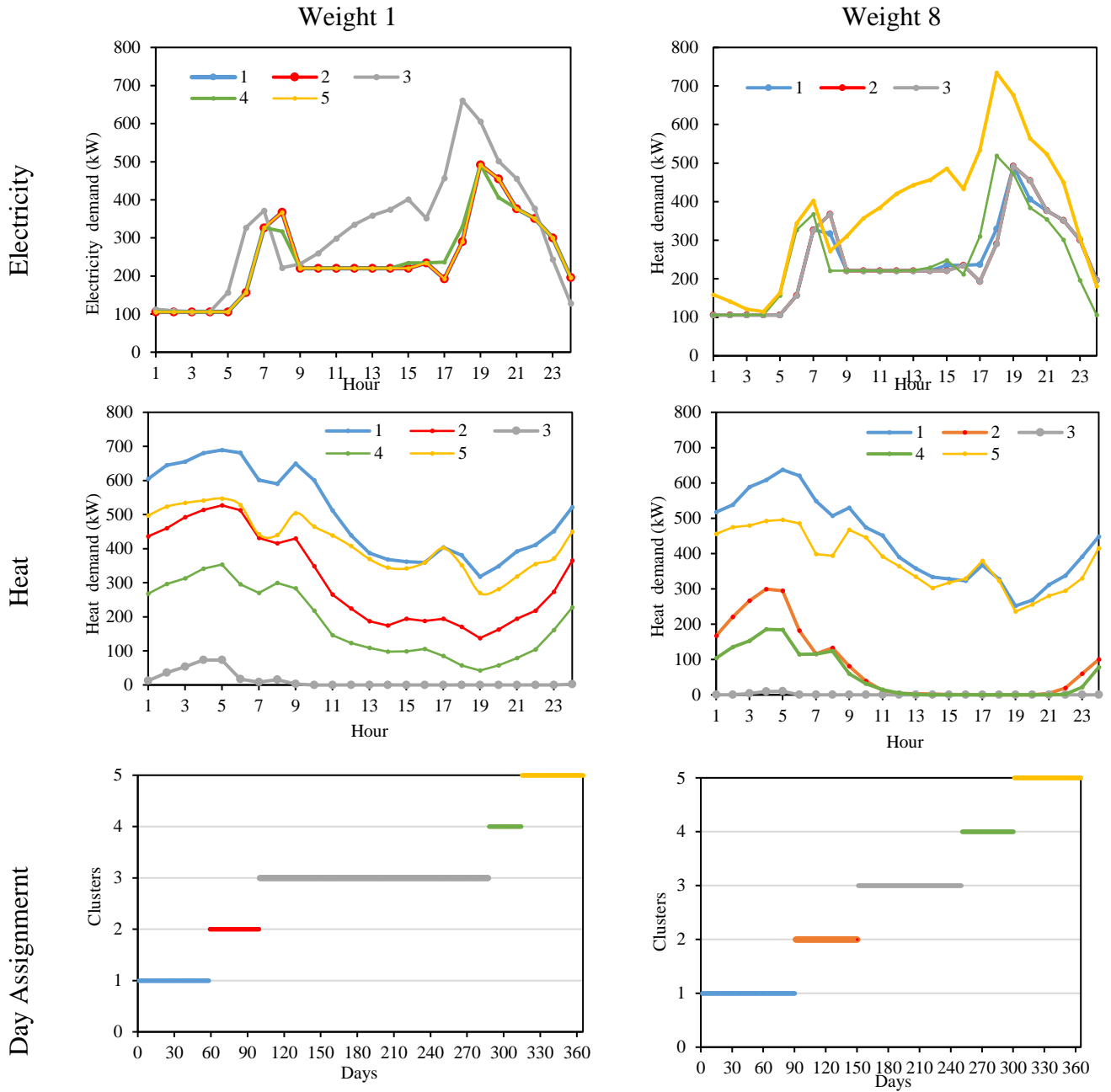


Figure B.9. heat and electricity demand cluster curves with day assignment for weight factors 1 and 8 using 5 sequence clustering

### Sequence Clustering using 6 clusters

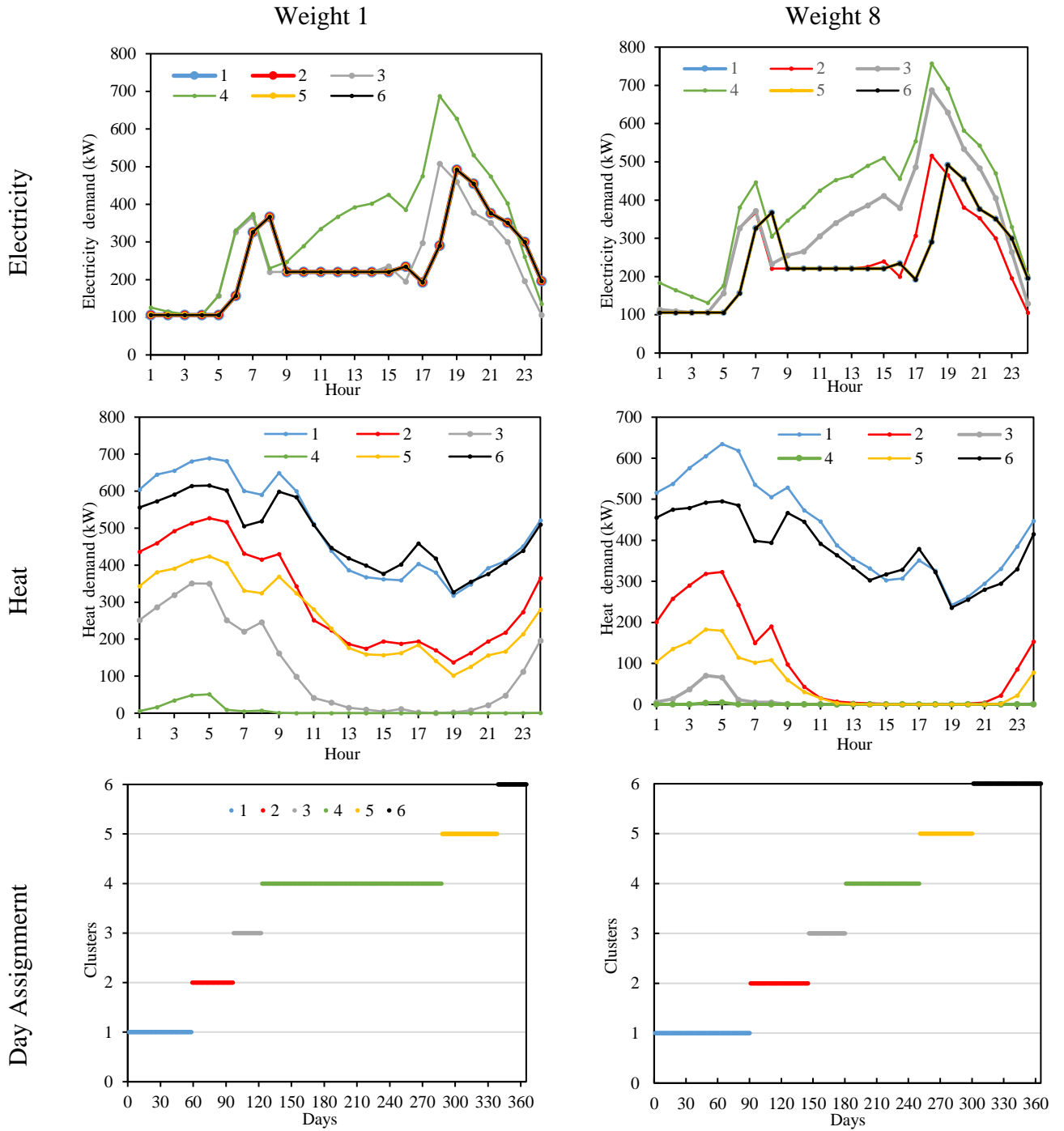


Figure B.10. heat and electricity demand cluster curves with day assignment for weight factors 1 and 8 using 6 sequence clustering

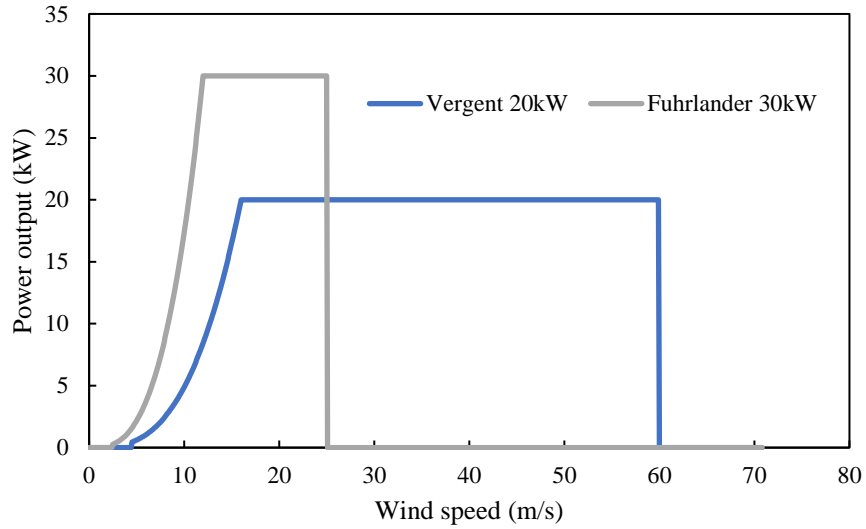


Figure B.11. Power output of the two wind turbines in this study as a function of wind speed

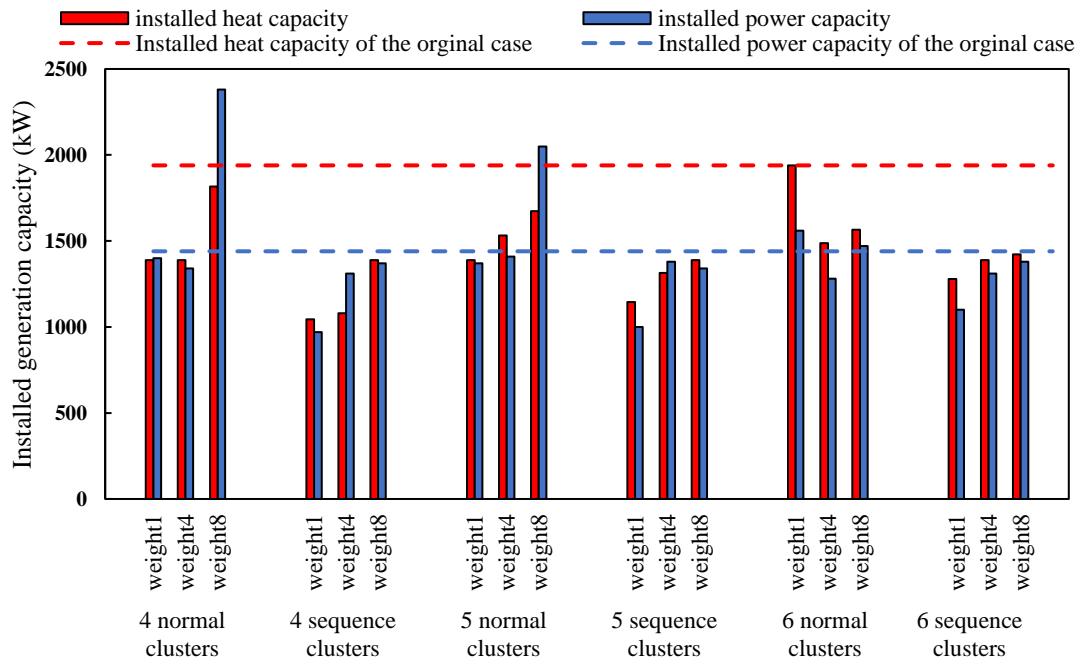


Figure B.12. Installed heat and power generation capacity for the energy hub system under CO<sub>2</sub> emission regulations

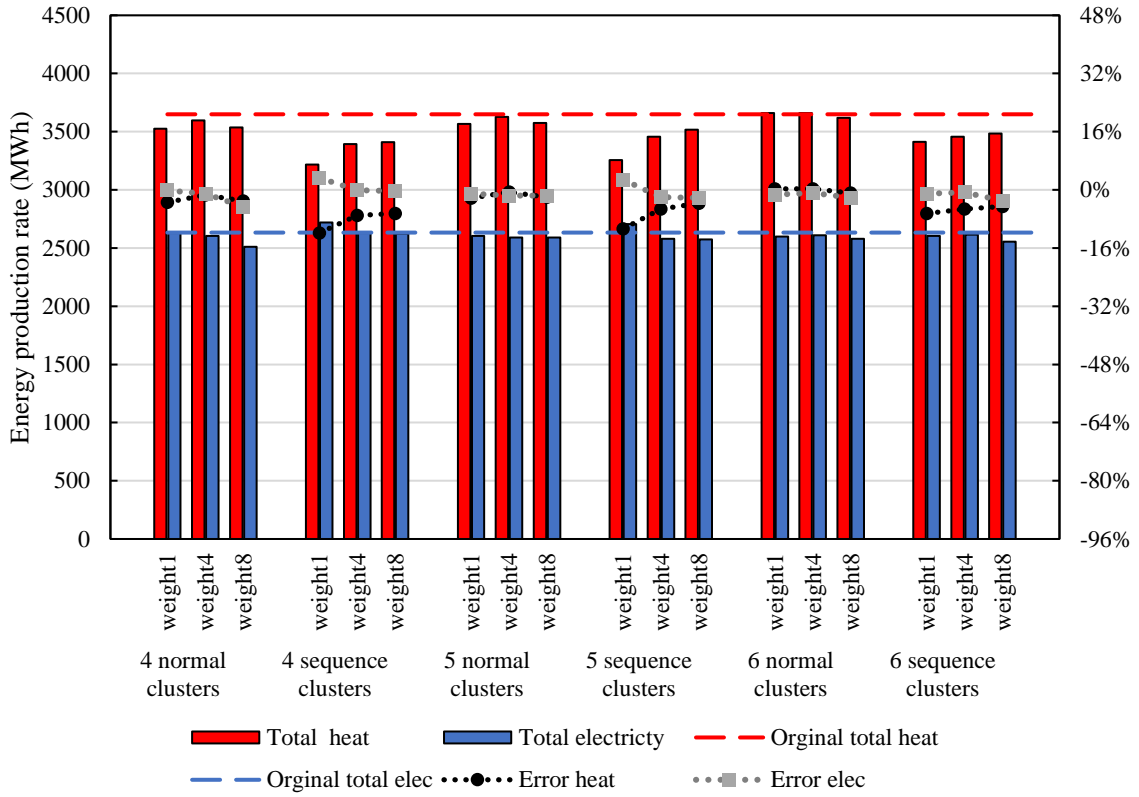


Figure B.13. Energy hub's total utility production rates comparison between original and clustered cases under CO<sub>2</sub> emissions regulation