uOttawa

L'Université canadienne
Canada's university

Machine learning and multicriteria optimization for the optimal design of chemical processes

Farough Agin

Thesis submitted to the University of Ottawa in partial fulfillment of the requirements for the degree of Ph.D. in Chemical Engineering

Department of Chemical and Biological Engineering

Faculty of Engineering

University of Ottawa, Ottawa, Ontario, Canada

Abstract

Chemical engineering systems are increasingly characterized by nonlinear behaviour, strong interactions among variables, and growing volumes of experimental, simulated, and literature-derived data. At the same time, many engineering problems involve competing technical objectives and require not only accurate prediction, but also systematic support for analysis, design, and decision-making. In this context, this thesis investigates data-driven frameworks for a range of chemical engineering applications, with emphasis on building predictive models that can be integrated into broader workflows for process analysis, optimization, and engineering decision support.

The thesis is organized around two main groups of studies. The first group focuses on process-level applications, using the styrene reactor as a representative case study. In this group, multiple supervised machine learning (ML) models are developed to predict ethylbenzene conversion and styrene yield, and the trained models are integrated with multi-objective optimization (MOO) and multi-criteria decision-making (MCDM) to identify favourable reactor operating conditions under competing objectives. This work demonstrates how model choice, data quality, and uncertainty influence the reliability of Pareto-optimal recommendations and introduces a consensus-based strategy to identify more robust operating conditions across different surrogate models and data scenarios. A related study addresses missing-data imputation in a styrene production dataset containing process variables using an autoassociative neural network (AANN). The results show that missing process variables can be reconstructed with acceptable accuracy, even when relatively small or noisy datasets are used, although the accuracy depends strongly on the correlation between the missing variable and the remaining process variables. Together, these studies demonstrate how data-driven methods can support both reactor optimization and data reconstruction, strengthening process understanding while improving the data foundation required for reliable downstream modelling and decision support.

The second group focuses on catalytic and reaction-oriented systems. One set of studies examines the catalytic conversion of sulfur dioxide to sulfur trioxide, where predictive models are developed to support catalyst and process design using literature-derived experimental data. In this work, different preprocessing strategies and feature representations are evaluated to determine how

data preparation choices influence model accuracy, interpretability, and the reliability of engineering conclusions. The resulting models are integrated with optimization to jointly consider conversion, productivity, and cost, and the study demonstrates that combining information obtained from different preprocessing strategies after optimization can reduce dependence on a single data representation and lead to less biased engineering recommendations. Another study addresses chemoselectivity in hydrogenation reactions involving competing organic substrates. In this work, ML models are developed to predict conversion and yield across different substrate–catalyst pairs and operating conditions, and the trained models are integrated with optimization to recommend catalysts and conditions that favour the desired transformation while limiting undesired conversion. This framework provides directional catalyst-recommendation maps and identifies favourable operating windows for selective hydrogenation. Across these studies, the emphasis is placed not only on predictive accuracy, but also on translating model outputs into practically useful guidance for catalyst selection, operating-condition screening, and reaction planning.

Overall, this thesis demonstrates that data-driven approaches can contribute far beyond prediction alone. Across process modelling, data reconstruction, catalytic process design, and reaction planning, the work shows how predictive frameworks can be used to support more informed, efficient, and transparent engineering decisions. The thesis, therefore, presents a coherent perspective on the role of data-driven methods in chemical engineering, showing their value as tools for transforming diverse data sources into actionable knowledge for analysis, design, and decision support.

Résumé

Les systèmes de génie chimique sont de plus en plus caractérisés par des comportements non linéaires, de fortes interactions entre les variables et des volumes croissants de données expérimentales, simulées et issues de la littérature. Parallèlement, de nombreux problèmes d'ingénierie impliquent des objectifs techniques concurrents et nécessitent non seulement des prédictions précises, mais aussi un soutien systématique à l'analyse, à la conception et à la prise de décision. Dans ce contexte, cette thèse étudie des cadres méthodologiques fondés sur les données pour diverses applications en génie chimique, en mettant l'accent sur le développement de modèles prédictifs pouvant être intégrés à des flux de travail plus larges pour l'analyse des procédés, l'optimisation et l'aide à la décision en ingénierie.

La thèse est organisée autour de deux principaux groupes d'études. Le premier groupe porte sur des applications à l'échelle du procédé, en utilisant le réacteur de production du styrène comme étude de cas représentative. Dans ce groupe, plusieurs modèles d'apprentissage automatique supervisé sont développés afin de prédire la conversion de l'éthylbenzène et le rendement en styrène, puis les modèles entraînés sont intégrés à une optimisation multiobjectif et à une méthode d'aide multicritère à la décision afin d'identifier des conditions opératoires favorables du réacteur en présence d'objectifs concurrents. Ces travaux montrent comment le choix du modèle, la qualité des données et l'incertitude influencent la fiabilité des recommandations Pareto-optimales, et introduisent une stratégie fondée sur le consensus pour identifier des conditions opératoires plus robustes à travers différents modèles de substitution et scénarios de données. Une étude connexe porte sur l'imputation de données manquantes dans un jeu de données de production du styrène comprenant variables de procédé, à l'aide d'un réseau neuronal autoassociatif. Les résultats montrent que les variables de procédé manquantes peuvent être reconstruites avec une précision acceptable, même lorsque des jeux de données relativement petits ou bruités sont utilisés, bien que cette précision dépende fortement de la corrélation entre la variable manquante et les autres variables de procédé. Ensemble, ces études démontrent comment les méthodes fondées sur les données peuvent soutenir à la fois l'optimisation du réacteur et la reconstruction des données, renforçant ainsi la compréhension du procédé tout en améliorant la base de données nécessaire à une modélisation en aval et à une aide à la décision fiables.

Le second groupe porte sur des systèmes catalytiques et réactionnels. Un ensemble d'études examine la conversion catalytique du dioxyde de soufre en trioxyde de soufre, dans laquelle des modèles prédictifs sont développés pour soutenir la conception de catalyseurs et de procédés à partir de données expérimentales issues de la littérature. Dans ce travail, différentes stratégies de prétraitement et représentations des variables sont évaluées afin de déterminer comment les choix liés à la préparation des données influencent la précision des modèles, leur interprétabilité et la fiabilité des conclusions d'ingénierie. Les modèles obtenus sont intégrés à une procédure d'optimisation visant à considérer conjointement la conversion, la productivité et le coût, et l'étude montre que la combinaison des informations issues de différentes stratégies de prétraitement après l'optimisation peut réduire la dépendance à une seule représentation des données et conduire à des recommandations d'ingénierie moins biaisées. Une autre étude porte sur la chimiosélectivité dans des réactions d'hydrogénation impliquant des substrats organiques concurrents. Dans ce travail, des modèles d'apprentissage automatique sont développés afin de prédire la conversion et le rendement pour différents couples substrat–catalyseur et différentes conditions opératoires, puis les modèles entraînés sont intégrés à une procédure d'optimisation afin de recommander les catalyseurs et les conditions favorisant la transformation souhaitée tout en limitant la conversion indésirable. Ce cadre fournit des cartes directionnelles de recommandation de catalyseurs et identifie des fenêtres opératoires favorables pour l'hydrogénation sélective. Dans l'ensemble de ces études, l'accent est mis non seulement sur la précision prédictive, mais aussi sur la traduction des résultats des modèles en orientations pratiquement utiles pour la sélection des catalyseurs, l'exploration des conditions opératoires et la planification des réactions.

Dans l'ensemble, cette thèse démontre que les approches fondées sur les données peuvent contribuer bien au-delà de la seule prédiction. À travers la modélisation des procédés, la reconstruction des données, la conception de procédés catalytiques et la planification des réactions, les travaux présentés montrent comment des cadres prédictifs peuvent être utilisés pour soutenir des décisions d'ingénierie plus éclairées, efficaces et transparentes. Cette thèse présente ainsi une perspective cohérente sur le rôle des méthodes fondées sur les données en génie chimique, en montrant leur valeur comme outils permettant de transformer diverses sources de données en connaissances exploitables pour l'analyse, la conception et l'aide à la décision.

Dedications

I dedicate this thesis, first and foremost, to my beloved wife, Mina. Your love, patience, and unwavering support have carried me through every stage of this journey. In moments of stress, uncertainty, and exhaustion, you were my source of comfort, strength, and hope. Your belief in me never faded, even when I doubted myself, and your presence made every challenge feel lighter. This achievement is deeply shared with you, because without your love and sacrifice, this journey would not have been the same.

I also dedicate this thesis to my dear parents, to whom I owe more than words can ever express. Your endless love, sacrifices, and prayers have been the foundation of my life and every success I have achieved. Everything I am today has been shaped by your guidance, your values, and your unconditional support. You gave me the strength to dream, the courage to continue, and the faith to persevere. No accomplishment can ever repay what I owe you, but I hope this work reflects even a small part of the gratitude and love I carry for you in my heart.

To my sisters, I dedicate this work with love and appreciation. Your kindness, encouragement, and constant presence in my life have meant more than I can fully express. Whether through words of support, shared memories, or simply knowing that you were always there for me, you brought me comfort and strength throughout this journey. I am truly grateful to have you by my side.

I also dedicate this thesis to my friends, whose friendship, encouragement, and support have brightened this long and demanding path. In times when the journey felt overwhelming, your presence reminded me that I was never alone. Your understanding, care, and companionship gave me strength, balance, and many moments of joy along the way.

To all of you, this thesis is dedicated with my deepest love and gratitude. Whatever I have achieved in these pages has been shaped, supported, and made meaningful by your presence in my life.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Jules Thibault, for his invaluable guidance, support, and encouragement throughout my doctoral studies. His knowledge, insight, and dedication to research have been a constant source of inspiration. I am especially grateful for his patience, trust, and the many opportunities I was given to grow as a researcher under his supervision. His mentorship has had a lasting impact on both this work and my academic development.

I would also like to sincerely thank my co-supervisor, Professor Clémence Fauteux-Lefebvre, for her guidance, encouragement, and thoughtful feedback throughout this journey. Her support and perspective have contributed greatly to this research, and I have truly appreciated her kindness, time, and willingness to help whenever needed. I am very grateful for her mentorship and for the positive role she has played in my PhD experience.

I would like to warmly thank Ariel Boucheikhchoukh and Laure Nshimiye Hakizumwami for their valuable contributions to the collection of literature data for the sulfur dioxide to sulfur trioxide projects. Their efforts and collaboration were important to the progress of this work, and I sincerely appreciate the time and care they devoted to it.

I am also sincerely grateful to Kostiantyn Prystupa and Professor Stephen Newman from the Department of Chemistry and Biomolecular Sciences for providing the experimental data used in the hydrogenation project. Their support and collaboration are greatly appreciated, and their contribution played an important role in making this part of the research possible.

I gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Eurofins CDMO Alphora. Their support made it possible to carry out the projects presented in this thesis, and I am truly thankful for the opportunities this funding created.

I would also like to thank the Digital Research Alliance of Canada for providing access to the computational resources and platform that supported this research. These resources were essential for carrying out the computational aspects of this work.

Finally, I would like to express my sincere appreciation to the Department of Chemical and Biological Engineering for its support throughout my doctoral studies. I am grateful for the

academic environment, resources, and sense of community that the department provided during this journey.

Statement of contributions

I hereby declare that this thesis represents my own original research and scholarly writing, except where the contributions of supervisors and collaborators are explicitly identified. I was responsible for formulating the research objectives, conducting the literature review, developing the methodologies, implementing the computational workflows, performing the data analysis and model development, interpreting the results, and preparing the thesis manuscript.

The work presented in Chapters 1–6, 8, and Appendices A and B was conducted by me under the supervision of Professor Jules Thibault and Professor Clémence Fauteux-Lefebvre. Professor Thibault and Professor Fauteux-Lefebvre provided supervisory guidance, research direction, technical advice, feedback on the interpretation of results, and critical review of the associated manuscripts and thesis chapters. I was responsible for carrying out the analyses, implementing the methods, interpreting the findings, and writing these chapters and appendices.

For the work presented in Chapter 7 and Appendix C, I was responsible for developing and implementing the machine learning, optimization, and decision-support framework, as well as performing the analysis, interpreting the results, and writing the chapter and appendix. Kostiantyn Prystupa and Professor Stephen Newman contributed to the preparation of the experimental dataset used in this part of the thesis. Professor Thibault and Professor Fauteux-Lefebvre provided supervisory guidance, technical feedback, and critical review of this work.

All final analyses, interpretations, written content, and conclusions presented in this thesis are my own, except where the contributions of supervisors and collaborators have been acknowledged.

Table of Contents

Abstract ii

Dedications vi

Acknowledgements vii

Statement of contributions ix

Table of Contents x

List of Figures xvi

List of Tables xxv

Nomenclature xxvii

Abbreviations xxix

Chapter 1: Introduction 1

1.1 Machine learning as a predictive framework in chemical engineering 2

1.2 From prediction to engineering improvement: The role of multi-objective optimization and decision-making 4

1.3 Scope, methodological positioning, and organization of the thesis 6

1.4 References 9

Chapter 2: Background 13

2.1 Introduction 14

2.2 Machine learning models 14

2.2.1 Artificial neural networks 14

2.2.2 Support vector regression 16

2.2.3 Gaussian process regression 18

2.2.4 K-nearest neighbours 20

2.2.5 Decision tree 21

2.2.6 Bagging 23

2.2.7 Random forest 25
2.2.8 Extremely randomized trees 26
2.2.9 eXtreme gradient boosting 27
2.3 Multi-objective optimization 31
2.4 Multi-criteria decision-making 36
2.5 References 38
Chapter 3: Harnessing Machine Learning and Optimization for Informed Chemical Engineering Decisions: A Styrene Reactor Analysis 46
3.1 Abstract 47
3.2 Graphical abstract 48
3.3 Introduction 49
3.4 Methodology 53
3.4.1 Styrene production process and dataset generation 53
3.4.2 Preprocessing and model development 57
3.4.3 Machine learning models 59
3.4.3.1 Artificial neural networks 59
3.4.3.2 Support vector regression 59
3.4.3.3 Gaussian process regression 59
3.4.3.4 K-nearest neighbours 60
3.4.3.5 Decision tree 60
3.4.3.6 Bagging 61
3.4.3.7 Random forest 61
3.4.3.8 Extremely randomized trees 61
3.4.3.9 eXtreme gradient boosting 62
3.4.4 Hyperparameter tuning 62

3.4.5 Optimization 62
3.4.6 Multi-criteria decision-making 63
3.5 Results and discussion 64
3.5.1 Model development 64
3.5.2 Optimization 64
3.5.2.1 NSGA-II Pareto domains and NFM ranking for each model and dataset 64
3.5.2.2 Comparison of dominancy of models 74
3.5.2.3 Computation time 76
3.5.2.4 Consensus approach to use the information from all the trained models 79
3.6 Conclusion 86
3.7 References 87
Chapter 4: Autoassociative Neural Network for Missing Data Imputation: A Case Study via the Styrene Production Process 96
4.1 Abstract 97
4.2 Introduction 98
4.3 Methodology 101
4.3.1 Autoassociative neural network (AANN) 101
4.3.2 Styrene production and dataset generation 103
4.3.3 Noisy dataset 106
4.3.4 Data preprocessing 107
4.4 Results and discussion 108
4.4.1 Final AANN architecture 108
4.4.2 Performance of AANN models for the seven dataset sizes 110
4.4.3 Estimation of one missing value for different sizes of the datasets 111
4.4.4 Performance of AANNs for two missing process variables 119

4.4.5 Performance of AANNs for three missing process variables 123
4.4.6 Building AANN models on noisy data 127
4.5 Conclusion 129
4.6 References 130
Chapter 5: Optimal Design of Catalytic Conversion of SO_2 to SO_3 via Machine Learning Modelling 137
5.1 Abstract 138
5.2 Introduction 139
5.3 Methodology 142
5.3.1 Artificial neural network model development 142
5.3.2 Multi-objective optimization 149
5.3.3 Multi-criteria decision-making 152
5.4 Results and discussion 154
5.4.1 Final ANN model 154
5.4.2 Case study 1: Vanadium-potassium catalyst 157
5.4.3 Case study 2: Platinum catalyst 161
5.5 Conclusion 166
5.6 References 167
Chapter 6: Machine Learning–Driven Design of Catalytic Processes for Sulfur Dioxide Oxidation: Lessons from the Trenches 176
6.1 Abstract 177
6.2 Graphical abstract 178
6.3 Introduction 179
6.4 Methodology 183
6.4.1 Problem definition 183
6.4.2 Literature data 183

6.4.3 Data preprocessing 185
6.4.3.1 Missing data handling 186
6.4.3.2 Feature engineering 186
6.4.3.3 Additional preprocessing 186
6.4.4 Model development 187
6.4.5 Optimization 188
6.4.6 Ranking Pareto-optimal solutions 191
6.5 Results and discussion 192
6.5.1 Scenario 1: Implementation of the uniform quantile transformer 193
6.5.2 Scenario 2: Implementation of the normal quantile transformer 195
6.5.3 Scenario 3: Implementation of the Box-Cox transformer 195
6.5.4 Scenario 4: Dataset with no data transformer 198
6.6 Challenges and recommendations for development of ML models for catalytic processes…. 201
6.7 Conclusion 205
6.8 References 206
Chapter 7: A Data-Driven Approach to Competitive Hydrogenation: Structure-Informed Machine Learning with Chemoselectivity-Based Optimization 214
7.1 Abstract 215
7.2 Introduction 217
7.3 Methodology 219
7.3.1 Problem statement, experimental setup, and dataset 219
7.3.2 Machine learning models 221
7.3.3 Data preprocessing 224
7.3.4 Data splitting and hyperparameter tuning 227
7.3.5 Optimization 228

7.4 Results and discussion 229
7.4.1 Model development and predictive performance 229
7.4.2 Chemoselectivity optimization and catalyst recommendation 234
7.4.3 Interpretable analysis of optimized operating-condition regimes 239
7.5 Conclusion 242
7.6 References 244
Chapter 8: Conclusion 251
8.1 Overview of the thesis 252
8.2 Summary of main findings 252
8.3 Main contributions of the thesis 254
8.4 Limitations and future research directions 256
8.5 Final remarks 257
8.6 References 258
Appendix A: Supporting Information of Chapter 3 260
Appendix B: Supporting Information of Chapter 6 300
Appendix C: Supporting Information of Chapter 7 315

List of Figures

Figure 2.1. A simple representation of a typical ANN structure. 15
Figure 2.2. Visualization of a simple linear SVR model. 17
Figure 2.3. Comparison of the GP prior and posterior distributions. The left panel shows random function samples from the prior with zero mean. The right panel illustrates the posterior distribution after conditioning on five observations, showing reduced uncertainty and updated function samples. 20
Figure 2.4. A simple representation of KNN. 21
Figure 2.5. Partitioning of the input space by a DT (left) and its corresponding hierarchical decision structure (right), where each split is based on feature thresholds and each leaf predicts a constant output. 23
Figure 2.6. Schematic representation of the Bagging regressor. The model generates multiple bootstrapped training datasets from the original data and trains a separate decision tree on each subset. The individual predictions are then averaged to produce a final, robust, and low-variance regression output. 24
Figure 2.7. Sequential tree-building process in XGBoost where each tree corrects the errors of the previous ones, and the final prediction is obtained by summing all outputs. 29
Figure 2.8. Diagram illustrating Pareto optimal solutions. a) Feasible, infeasible, and optimal solutions [68], b) decision and objective spaces [71]. 33
Figure 2.9. Zones of a design problem [72]. 33
Figure 2.10. a) Procedure of NSGA-II, b) Crowding distance [69]. 35
Figure 3.1. Annual number of publications involving the combined use of ML, MOO, and MCDM: (a) across all fields and (b) specifically within chemical engineering, based on Scopus and Web of Science data. 51
Figure 3.2. Workflow of the proposed framework integrating ML, NSGA-II, and NFM for optimizing styrene production based on simulation data. 55
Figure 3.3. Pareto fronts and NFM rankings for models trained on the complete dataset. Each plot corresponds respectively to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost. 71

Figure 3.4. Pareto fronts and NFM rankings for models trained on the 1000-record clean dataset. Each plot corresponds respectively to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost. 72
Figure 3.5. Pareto fronts and NFM rankings for models trained on the 1000-record noisy dataset. Each plot corresponds respectively to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost. 73
Figure 3.6. Non-dominated Pareto-optimal solutions for nine ML models across three training scenarios: (a) complete dataset, (b) 1000-record clean dataset, and (c) 1000-record noisy dataset. 75
Figure 3.7. Comparison of hyperparameter tuning and training time plotted on a logarithmic scale. 78
Figure 3.8. Comparison of optimization time plotted on a logarithmic scale. 78
Figure 3.9. Number of Pareto-optimal (non-dominated) solutions generated by each model across the three datasets. Results were obtained using the NSGA-II algorithm with a population size of 1000. 79
Figure 3.10. Champion points identified by the NFM for each model across both the objective and decision spaces using the complete dataset. Subplots (a) and (c) depict the optimal decision spaces, while (b) and (d) present the corresponding optimal objective values, which have been further validated through simulation. 81
Figure 3.11. Champion points identified by the NFM for each model across both the objective and decision spaces using the 1000-record clean dataset. Subplots (a) and (c) depict the optimal decision spaces, while (b) and (d) present the corresponding optimal objective values, which have been further validated through simulation. 82
Figure 3.12. Champion points identified by the NFM for each model across both the objective and decision spaces using the 1000-record noisy dataset. Subplots (a) and (c) depict the optimal decision spaces, while (b) and (d) present the corresponding optimal objective values, which have been further validated through simulation. 83
Figure 4.1. Simple example of an AANN model. 102
Figure 4.2. Styrene production process diagram. 103
Figure 4.3. Pearson correlation matrix between all of the variables. 106
Figure 4.4. AANN architecture used in this investigation. 110

Figure 4.5. MSE values for the training, validation, and test data for seven datasets. 111
Figure 4.6. Histogram of the estimated errors of the reactor length as one missing process variable for the AANN models obtained with different sizes of the datasets. (A) 29000; (B) 25000; (C) 15000; (D) 5000; (E) 1000; (F) 500; (G) 100. 114
Figure 4.7. Histogram of the estimated errors of the outlet styrene flow rate as one missing process variable for the AANN models obtained with different sizes of the datasets. (A) 29000; (B) 25000; (C) 15000; (D) 5000; (E) 1000; (F) 500; (G) 100. 115
Figure 4.8. (A) Predicted values of a sample of the normalized styrene flow rate obtained from two models trained with 29000 and 100 records; (B) MSE of predicted styrene flow rates of the 1000-record evaluation dataset for two models trained with 29000 and 100 records at each iteration. 116
Figure 4.9. Heat map of the standard deviations of the effect of one missing variable on itself and on the other 21 variables for the model developed with the 29000-record dataset. 118
Figure 4.10. Heat map of the standard deviations of the effect of one missing variable on itself and on the other 21 variables for the model developed with the 100-record dataset. 119
Figure 4.11. Mean and standard deviation of the prediction errors from the 1000-record dataset for two missing variables as a function of the size of the dataset used to train AANNs. (A) outlet flow rate of ethylbenzene and conversion; (B) length of the reactor and outlet flow rate of water.... 121
Figure 4.12. Heat map of the standard deviations of the differences between the estimations with and without the missing values, showing the effect of two missing variables on themselves and other variables. (A) 29000 record AANN; (B) 100 record AANN. 123
Figure 4.13. Mean and standard deviation of the prediction errors for three missing variables as a function of the size of the dataset used to train AANNs. (A) SOR, inlet flow rate of ethylbenzene, and inlet flow rate of water; (B) length of reactor, outlet flow rate of water, and outlet flow rate of CO. 125
Figure 4.14. Heat map of the standard deviations of the differences between the estimations with and without the missing values showing the effect of three missing variables on themselves and other variables. (A) 29000 record AANN; (B) 100 record AANN. 126
Figure 5.1. Flowchart of the complete steps toward the optimization of the catalytic conversion of SO2. 143
Figure 5.2. (A) FFNN with three layers and (B) a sample node of the network. 143

Figure 5.3. Schematic illustration of the procedure of NSGA-II. 150
Figure 5.4. Performance of the final ANN model: (A) Training and validation MSE as a function of the number of epochs, and the parity plots of the conversion for the (B) training dataset, (C) validation dataset, and (D) testing dataset. 156
Figure 5.5. Pareto domain: (A) conversion vs productivity, (B) conversion vs cost, and (C) cost vs productivity for vanadium and potassium case study. 160
Figure 5.6. Decision space: (A) vanadium wt% vs potassium wt% and (B) temperature vs W/V for vanadium and potassium case study. 162
Figure 5.7. Pareto domain: (A) conversion vs productivity, (B) conversion vs cost, and (C) cost vs productivity for the platinum case study. 164
Figure 5.8. Decision space: (A) platinum wt% vs temperature and (B) temperature vs W/V for the platinum case study. 165
Figure 6.1. Proposed framework of optimized catalysts with knowledge extraction. 184
Figure 6.2. Raw dataset without any missing variables. 185
Figure 6.3. Feature importance visualization of the 29 variables (conversion excluded) determined by (a) random forest regressor and (b) PCA. 187
Figure 6.4. Parity plots of the ANN model for scenario 1 (uniform quantile transformer). 194
Figure 6.5. Results of MOO for vanadium-potassium case study and scenario 1 (uniform quantile transformer) (a) conversion vs productivity, (b) conversion vs cost, (c) productivity vs cost, (d) vanadium wt% vs potassium wt%, and (e) W/V vs temperature. 194
Figure 6.6. Parity plots of the ANN model for scenario 2 (normal quantile transformer). 195
Figure 6.7. Results of MOO for vanadium-potassium case study for scenario 2 (normal quantile transformer) (a) conversion vs productivity, (b) conversion vs cost, (c) productivity vs cost, (d) vanadium wt% vs potassium wt%, and (e) W/V vs temperature. 196
Figure 6.8. Parity plots of the ANN model for scenario 3 (Box-Cox transformer). 197
Figure 6.9. Results of MOO for vanadium-potassium case study and scenario 3 (Box-Cox transformer) (a) conversion vs productivity, (b) conversion vs cost, (c) productivity vs cost, (d) vanadium wt% vs potassium wt%, and (e) W/V vs temperature. 197
Figure 6.10. Parity plots of the ANN model for scenario 4 (Dataset without transformation). . 199

Figure 6.11. Results of MOO for vanadium-potassium case study and scenario 4 (Dataset without transformation) (a) conversion vs productivity, (b) conversion vs cost, (c) productivity vs cost, (d) vanadium wt% vs potassium wt%, and (e) W/V vs temperature. 199
Figure 7.1. Chemical space of hydrogenation reactions covered in this work. Relevant functional group classes were represented by model substrates and reduced under a diverse set of catalytic reaction conditions. 221
Figure 7.2. Overview of the ML framework used in this study. Selected experimental inputs are provided to a suite of regression models, which are trained and compared to predict key reaction performance metrics. 222
Figure 7.3. RF–based feature-importance ranking of substrate molecular descriptors for predicting conversion and yield. Importance scores were computed separately for each target to account for target-specific descriptor relevance. The top ten descriptors per target were selected; four descriptors were common to both lists, and the remaining descriptors were target-specific, yielding sixteen unique descriptors used as the shared substrate descriptor set for model training. 225
Figure 7.4. Conversion model performance under Strategy 1 (one-hot substrate and catalyst): training vs. test MSE and R^2 across sixteen models using leave-one-experiment-out evaluation. 230
Figure 7.5. Yield model performance under Strategy 1 (one-hot substrate and catalyst): training vs. test MSE and R^2 across sixteen models using leave-one-experiment-out evaluation. 231
Figure 7.6. Conversion model performance under Strategy 2 (RDKit substrate descriptors + one-hot catalyst): training vs. test MSE and R^2 across sixteen models using leave-one-experiment-out evaluation. 233
Figure 7.7. Yield model performance under Strategy 2 (RDKit substrate descriptors + one-hot catalyst): training vs. test MSE and R^2 across sixteen models using leave-one-experiment-out evaluation. 234
Figure 7.8. GA-optimised chemoselectivity landscape for ordered substrate pairs using the selected best ML predictors from Strategy 2. Each cell represents a substrate pair in which Substrate A is targeted for hydrogenation and Substrate B is intended to remain unreacted. Cell colour indicates the maximum chemoselectivity score identified by the GA, while the cell label indicates the corresponding optimal catalyst. 235

Figure 7.9. Distributions of genetically algorithm–optimized Conversion of A (blue) and Conversion of B (orange) for each catalyst across all 90 ordered substrate pairs (Substrate A versus Substrate B; n = 90 per catalyst). For every catalyst–pair combination, temperature, pressure, and flow rate were optimized independently, and the histograms summarize the resulting conversion values (10% bins). The dashed line indicates 50% conversion. Catalyst panels are arranged according to their frequency of selection as the optimal catalyst. 238
Figure 7.10. PCA visualization of k-means clustering (k = 10) for GA-optimal operating conditions (temperature, pressure, flow rate), with point size proportional to the chemoselectivity score. 239
Figure 7.11. k-means clustering results (k = 10) in the original operating-variable space, showing chemoselectivity score versus temperature, pressure, and flow rate for the GA-optimal solutions, coloured by cluster. 240
Figure A.1. Keyword co-occurrence network showing dominant research themes and interconnections in studies combining ML, MOO, and MCDM. Node size indicates frequency, and colours represent thematic clusters. 262
Figure A.2. Annual keyword frequency trends (2010–2021) showing the rise of key concepts such as machine learning, optimization, and artificial intelligence in the literature combining ML, MOO, and MCDM. 263
Figure A.3. Word cloud illustrating the most frequent keywords in the reviewed literature on ML, MOO, and MCDM. Larger font size indicates higher frequency of occurrence. 263
Figure A.4. Histogram of the complete dataset. 273
Figure A.5. Histogram of the 1000-record clean dataset. 273
Figure A.6. Histogram of the 1000-record noisy dataset. 274
Figure A.7. Amount of noise added to conversion and yield. 274
Figure A.8. Decision space plots (steam vs ethylbenzene flow rates) for all models for the complete dataset. Subplots correspond to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost. 283
Figure A.9. Decision space plots (steam vs ethylbenzene flow rates) for all models for the 1000-record clean dataset. Subplots correspond to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost. 284

Figure A.10. Decision space plots (steam vs ethylbenzene flow rates) for all models for the 1000-record noisy dataset. Subplots correspond to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost. 285
Figure A.11. Decision space plots (inlet pressure vs inlet temperature) for all models for the complete dataset. Subplots correspond to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost. 286
Figure A.12. Decision space plots (inlet pressure vs inlet temperature) for all models for the 1000-record clean dataset. Subplots correspond to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost. 287
Figure A.13. Decision space plots (inlet pressure vs inlet temperature) for all models for the 1000-record noisy dataset. Subplots correspond to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost. 288
Figure A.14. Pareto fronts generated from 10 repeated NSGA-II runs using different random seeds for the ANN model on the 1000-record clean dataset. The almost indistinguishable fronts demonstrate that the optimization results are highly stable with respect to seed selection under the adopted optimization settings. 289
Figure A.15. Full overlay of the Pareto fronts generated by all nine ML models together with the corresponding global Pareto front for the three data scenarios: (a) complete dataset, (b) 1000-record clean dataset, and (c) 1000-record noisy dataset. The black curve represents the global Pareto front obtained from the pooled non-dominated solutions, allowing direct visual comparison of the deviation of each model-specific front from the overall best trade-off surface. 290
Figure A.16. Predictive uncertainty bands for the GPR-based Pareto fronts. Panels (a) and (b) correspond to the 1000-record clean dataset, and panels (c) and (d) correspond to the 1000-record noisy dataset. Panels (a) and (c) show the 95% confidence bands for yield, while panels (b) and (d) show the 95% confidence bands for conversion, all plotted in the standard Pareto orientation of yield versus conversion. 293
Figure B.1. Number and amount of missing values in the databank related to catalyst physical properties and synthesis techniques. 306
Figure B.2. Parity plots of the AANN model for training, validation, and testing for the prediction of the two randomly selected variables of the 30-variable dataset (a) normalized conversion and (b) normalized temperature. 307

Figure B.3. Parity plots of the estimation of the missing values by the AANN model for conversion and temperature (a) normalized conversion and (b) normalized temperature. 307
Figure B.4. Heat map of the correlation matrix using the Pearson correlation coefficients for the 30-variable dataset. 308
Figure B.5. Parity plots of the ANN model for scenario 1 (with logical data). 310
Figure B.6. Results of MOO for vanadium-potassium case study for scenario 1 (with logical data) (a) conversion vs productivity, (b) conversion vs cost, (c) productivity vs cost, (d) vanadium wt% vs potassium wt%, and (e) temperature vs W/V. 311
Figure C.1. Overall interface of the developed Streamlit-based chemoselectivity web application for hydrogenation, showing the sidebar for substrate selection and the two main analysis sections, Catalyst Search and Full Optimization. 318
Figure C.2. Sidebar of the chemoselectivity web application showing an example substrate-pair selection, with 1-Nitronaphthalene chosen as Substrate A and 2-Phenylacetophenone chosen as Substrate B, together with the two analysis options, Catalyst search and Full optimization. 319
Figure C.3. Overview of the Catalyst Search section in the web application, showing the on-screen guidance provided to the user before running the screening analysis. 320
Figure C.4. Expandable More Info panel in the Catalyst Search section, showing the 21 operating conditions used in the screening workflow. The bold rows indicate the original experimental operating conditions used for model training, while the remaining rows represent interpolated intermediate conditions introduced to create a continuous operating path. 322
Figure C.5. Result of the Catalyst Search workflow for the selected substrate pair, showing the maximum chemoselectivity score achieved by each candidate catalyst across the 21 pre-defined operating conditions. The highlighted bar identifies the catalyst with the highest score and therefore the selected best catalyst candidate. 324
Figure C.6. Metric-selection menu in the Catalyst Search workflow, allowing the user to choose the performance indicator to be visualized. Based on the selected metric, the application generates the corresponding heat map and score plot for catalyst comparison across the pre-defined operating conditions. 324
Figure C.7. Heat map of catalyst performance for the selected substrate pair based on the chosen metric, shown here for chemoselectivity score. The heat map compares all available catalysts

across the five representative operating conditions corresponding to the mildest, mild, intermediate, harsh, and harshest levels of operating severity..... 326
Figure C.8. Score plot of the selected metric across all 21 pre-defined operating conditions for each available catalyst. In this example, the plotted metric is chemoselectivity score, allowing comparison of catalyst performance over the full screening range. 327
Figure C.9. Full results table generated by the Catalyst Search workflow, summarizing the predicted performance of all catalysts across the 21 pre-defined operating conditions for the selected substrate pair. The table can also be downloaded by the user as a CSV file for further analysis..... 329
Figure C.10. Introductory section of the Full Optimization module in the web application, presenting the optimization objective and the user instructions for retrieving the saved genetic algorithm results for the selected substrate pair..... 330
Figure C.11. Initial optimization result displayed in the Full Optimization module, showing the catalyst identified by the genetic algorithm as the best candidate for the selected substrate pair, together with the maximum selectivity score achieved by each catalyst under optimized operating conditions..... 332
Figure C.12. Sankey-style summary of the best optimization result, showing the catalyst selected by the genetic algorithm together with the corresponding optimal temperature, pressure, flow rate, and maximum selectivity score for the chosen substrate pair..... 333
Figure C.13. Genetic algorithm landscape showing, for each catalyst, the operating conditions at which its maximum selectivity score is achieved. The three panels separately display the optimal temperature, pressure, and flow rate associated with the best optimization result for each catalyst. 334
Figure C.14. Final tabulated output of the Full Optimization workflow, listing for each catalyst the operating conditions associated with its maximum selectivity score, together with the corresponding predicted yield, conversion values, and selectivity score. The table can also be downloaded by the user for further analysis and recordkeeping. 335

List of Tables

Table 2.1. Brief description, advantages, and disadvantages of each model used in this work. . 29
Table 3.1. Summary of selected input and output variables used in the ML models, including their minimum and maximum values as observed in the complete dataset. 58
Table 3.2. Model performance comparison for all models on the complete and 1000-record datasets with and without noise on both training and test data. 66
Table 3.3. NFM parameters for ethylbenzene conversion and styrene yield..... 67
Table 3.4. Majority and minority consensus operating points for each dataset, derived from the champion solutions using the consensus-based selection approach under the imposed process constraints. 84
Table 4.1. Description of the generated dataset. 105
Table 4.2. Description of the noisy dataset. 107
Table 4.3. Final neural network parameters..... 109
Table 4.4. MSE of AANN models for two cases: clean data and noisy data. 127
Table 4.5. MAE between denormalized observed data and predicted missing values of different AANN models of clean and noisy datasets..... 128
Table 5.1. SO_2 conversion databank description with 22 input variables and one output variable used to develop the ANN model to predict conversion. 146
Table 5.2. Estimated price of each catalyst..... 152
Table 5.3. Structure of the final ANN model..... 155
Table 5.4. Comparative performance of the final ANN and traditional regression models on the testing data. 156
Table 5.5. Values of the decision parameters for the catalytic combination of vanadium and potassium. 158
Table 5.6. NFM parameters for vanadium and potassium case study. 159
Table 5.7. NFM parameters for platinum case study. 162
Table 6.1. Summary of ML applications in catalysis using literature data..... 181
Table 6.2. The estimated price of each metal. 191
Table 6.3. NFM parameters for the vanadium and potassium case study. 192
Table 6.4. Highlights of the Pareto fronts obtained from the different scenarios..... 202

Table 7.1. Summary of the selected RDKit substrate descriptors used for model development, including a brief definition of each descriptor and its corresponding descriptor category [55]. .. 226
Table A.1. Searched keywords in Scopus and Web of Science. 261
Table A.2. Summary of previous studies. 266
Table A.3. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the ANN models. 275
Table A.4. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the SVR models. 276
Table A.5. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the KNN models. 277
Table A.6. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the GPR models. 277
Table A.7. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the DT models. 278
Table A.8. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the Bagging models. 279
Table A.9. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the ET models. 280
Table A.10. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the RF models. 281
Table A.11. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the XGBoost models. 282
Table A.12. Comparison of champion points (obtained by NFM) of each model for both objectives and decision space for all three datasets. 291
Table A.13. Local inter-model uncertainty around the majority and minority consensus points for the complete, 1000-record clean, and 1000-record noisy datasets. For each consensus point, the nearest Pareto-optimal solution from each ML model was identified in normalized objective space using conversion and yield, and the resulting dispersion across models was summarized by the mean, standard deviation, and range. 292
Table B.1. Summary of elements removed during preprocessing. 309

Nomenclature

b^i	Bias
C	Cumulative cost of catalysts
c_i	Price of each catalyst
$C[i,j]$	Global concordance index
$c_m[i,j]$	Individual concordance index
d_i	Crowding distance
$D_m[i,j]$	Discordance index
f	Cost function
$F(X)$	Objective function space
$F_{EB,in}$	Inlet ethylbenzene molar flow rate
$F_{EB,out}$	Outlet ethylbenzene molar flow rate
F_{in}	Total inlet molar flow rate
$F_{SO_2,in}$	Inlet sulfur dioxide molar flow rate
$F_{SO_2,out}$	Outlet sulfur dioxide molar flow rate
F_{ST}	Styrene flow rate
$g_h(X)$	Inequality
$h_g(X)$	Equality
m_i	Mass fraction of each catalyst
N	Number of training points
q	Number of output neurons
P	Productivity
P_k	Preference threshold
P_t	Parent population
Q_k	Indifference threshold
Q_t	Offspring population
R^2	Coefficient of determination
R_t	New population
V_k	Veto threshold

v_t	Sum of squares of the previous gradients
W	Weight vector
W	Catalyst weight
w_j^i	Weight assigned to each connection
W_k	Relative weight
W/V	Ratio of catalyst weight to volumetric flow rate
X	Conversion
X	Input vector and decision-variable space
$X_{protected}$	Conversion of a substrate to protect
x_i^{LL}	Lower limit
x_i^{UL}	Upper limit
x_p	Input signal
Y	Yield
$Y_{desired}$	Yield from a substrate to react
y_i^{noisy}	Output with noise
y_i	Excitation output of a node
$\hat{y}_{ij}^t$	Output of the FNN
y_{ij}^t	Desired output
y_{SO_2}	Input sulfur dioxide mole fraction
λ	Box-Cox transformation parameter
ε	Small positive constant
ε_i	Gaussian noise
$\sigma[i,j]$	Outranking matrix
σ_i	Ranking score
φ_i	Activation function
$\Delta_m[i,j]$	Difference between the objective m of solutions i and j

Abbreviations

Adam	Adaptive moment estimation
AANN	Autoassociative neural network
AI	Artificial intelligence
ANN	Artificial neural network
Bagging	Bagging regression
BRR	Bayesian Ridge regression
CART	Classification and Regression Trees
CS	Chemoselectivity score
DT	Decision tree
EA	Evolutionary algorithms
EN	Elastic Net regression
ET	Extremely randomized trees
FGD	Flue-gas desulphurization
FFNN	Feedforward neural network
GA	Genetic algorithm
GB	Gradient boosting
GPR	Gaussian process regression
KNN	k-nearest neighbours
LightGBM	Light gradient boosting machine
MAE	Mean absolute error
MCDM	Multi-criteria decision-making
ML	Machine Learning
MLP	Multilayer perceptron
MOEA	Multi-objective evolutionary algorithm
MOO	Multi-objective optimization
MSE	Mean squared error
NFM	Net flow method
NSCR	Non-selective catalytic reduction
NSGA-II	Non-dominated sorting genetic algorithm

OOB	Out of bag
PCA	Principal component analysis
RF	Random forest
RMSE	Root mean squared error
SOR	steam-to-reactant ratio
SVM	Support vector machine
XGBoost	eXtreme gradient boosting

Chapter 1

Introduction

1.1 Machine learning as a predictive framework in chemical engineering

Chemical and process engineering has long relied on mathematical modelling to describe transport phenomena, reaction kinetics, thermodynamic behaviour, control interactions, and overall process performance. This tradition remains fundamental, as it underpins process design, scale-up, operation, and continuous improvement. However, the systems encountered by contemporary engineers are significantly more complex than those historically represented by compact analytical or semi-empirical models. Modern processes are highly nonlinear, strongly coupled, increasingly data-rich, and subject to economic, environmental, and operational constraints that must often be addressed simultaneously. In many cases, first-principles models remain indispensable, yet their routine application can be hindered by high computational cost, uncertain parameters, or incomplete knowledge of the underlying mechanisms. This disconnect between mechanistic rigor and practical tractability has created a growing need for complementary data-driven tools that can learn from experiments, simulations, and industrial operations while remaining aligned with engineering objectives [1–3].

Within this broader transition, artificial intelligence (AI) has emerged as an enabling paradigm, with machine learning (ML) becoming its most practically relevant branch for engineering applications. From an engineering perspective, the value of ML lies in its ability to extract relationships directly from data and to build predictive models for systems that are multivariable, noisy, nonlinear, and dynamic. Rather than requiring all relationships to be specified explicitly, ML methods infer patterns from historical observations and apply them to tasks such as prediction, classification, monitoring, or decision support. Reviews spanning process industries, biochemical systems, combustion applications, and intelligent manufacturing show that ML is now widely used for soft sensing, fault diagnosis, predictive maintenance, quality estimation, process monitoring, control support, and digital-twin-oriented analytics [3–6]. The current prominence of ML therefore reflects not only advances in algorithms, but also the convergence of large datasets, stronger computing infrastructure, open-source software ecosystems, and Industry 4.0 environments that have made implementation far more feasible than in earlier decades [7,8].

The chemical engineering literature reflects this shift particularly clearly. Early applications of AI in the field relied mainly on expert systems and symbolic reasoning, but these approaches proved difficult to scale because they required extensive manual encoding of domain expertise. Subsequent advances in neural networks broadened the range of potential applications by enabling empirical learning directly from process data; however, earlier generations were constrained by limited data availability, insufficient computing resources, and poor interpretability. More recent work has repositioned ML as a practical engineering tool rather than a speculative alternative. Contemporary surveys report successful applications in process monitoring, soft sensing, control-oriented prediction, catalyst screening, synthesis planning, product design, and surrogate modelling of complex process systems [1,4,6,7]. Importantly, these studies do not frame ML as a replacement for engineering knowledge; instead, they emphasize that ML is most effective when grounded in physically meaningful variables, high-quality curated data, and well-defined engineering performance objectives.

A major factor driving this growing role is speed. Chemical and process systems generate vast quantities of information from sensors, laboratory studies, pilot plants, simulations, and historical operating records, yet these data yield value only when translated into actionable engineering knowledge. ML facilitates this translation by learning predictive mappings between process inputs and key outputs such as conversion, yield, selectivity, energy consumption, emissions, product quality, equipment state, or control performance. In many applications, these learned models can be evaluated far more rapidly than rigorous simulations or repeated experiments, making them especially attractive for screening, monitoring, and rapid analysis [6-10]. As a result, reviews in process systems engineering and sustainable process design increasingly frame ML as a means of improving responsiveness while reducing computational demands in data-rich engineering environments [4,8,9]. At the same time, the literature consistently emphasizes that predictive accuracy alone is insufficient: careful data preprocessing, model validation, awareness of extrapolation limits, and adherence to physical plausibility remain essential for learned models to be credible and useful in engineering practice [1,10,11].

ML has also become increasingly important in materials discovery and development, an area of particular relevance to chemical engineering because materials properties often determine process feasibility and performance. The search spaces associated with catalysts, polymers, solvents, adsorbents, battery materials, and other functional materials are far too large to be

explored exhaustively through experiments alone. As a result, ML is increasingly used for tasks such as property prediction, inverse design, descriptor generation, high-throughput screening, and the prioritization of promising candidates prior to laboratory validation [12,13]. Similar trends are visible in catalyst discovery, where ML has been used to identify structure–property relationships and to guide the targeted development of improved catalytic materials [14,15]. From the perspective of this thesis, this extension toward materials discovery is important because it shows that ML contributes not only to downstream process analysis but also to upstream design choices that fundamentally shape process behaviour. In this sense, ML assists engineers addressing two closely related questions: how a system is expected to behave, and which systems or materials are worth exploring in the first place.

The main emphasis of this thesis is therefore on the application of ML in general, and more specifically on its use within chemical engineering and materials-related contexts. The central theme is the capability of ML to build predictive representations of complex engineering systems from available data, especially in situations where direct mechanistic evaluation is slow, uncertain, or impractical for repeated use. Such predictive models form the analytical core of many modern engineering workflows because they facilitate rapid response estimation, support sensitivity analysis, and enable efficient exploration of process or material design spaces. For this reason, ML is not treated here as a secondary computational convenience; it serves as the main methodological framework through which complex relationships become accessible for subsequent engineering analysis.

1.2 From prediction to engineering improvement: The role of multi-objective optimization and decision-making

Once a predictive layer is established, optimization naturally becomes the next step. In engineering, models are developed not merely to describe systems, but to improve their performance. Chemical processes rarely involve a single objective; instead, engineers must typically balance multiple, often competing goals such as yield, selectivity, cost, energy efficiency, emissions, safety, or controllability. Multi-objective optimization (MOO) is therefore essential, as it represents these trade-offs explicitly and identifies non-dominated alternatives rather than collapsing complex design problems to a single oversimplified objective [16–18]. Within the context of this thesis, optimization should be viewed as operating on top of the predictive capability

provided by ML. The ML model provides a computationally efficient representation of the system, and optimization exploits this representation to search for favourable operating conditions or design regions. This layered structure is especially valuable when rigorous evaluations are computationally expensive, which explains why surrogate-based optimization has emerged as a key interface between ML methods and chemical engineering analysis [9–11].

Recent studies in chemical engineering illustrate this integration in a specific way. ML-assisted frameworks have been applied to the MOO of systems such as thermal cracking, gas sweetening, pressure swing adsorption, and control-oriented process applications. In these cases, predictive surrogate models reduce computational demand while preserving the essential relationships between inputs (decision variables) and system responses [19–22]. Related reviews in biofuel systems and in engineering optimization more broadly reveal the same pattern: ML strengthens optimization by providing fast and informative estimates of system behaviour, while optimization provides a structured framework for identifying superior solutions [6,23]. For this reason, optimization remains an important component of the present work. Although it is not the primary analytical focus, it serves as the mechanism through which predictive ML models are translated into actionable engineering improvements.

Decision making represents the final stage of this methodological chain. Even after MOO has generated a Pareto set of viable alternatives, engineering practice typically requires the selection of a single implementable solution. Multi-criteria decision-making (MCDM) methods are useful at this stage because they make the basis of selection more transparent. Methods such as TOPSIS, VIKOR, AHP, NFM, and related ranking approaches help structure trade-offs and support systematic, transparent selection among technically acceptable solutions [24–26]. In the present thesis, decision making is therefore not treated as a separate dominant topic, but as an important concluding layer that translates prediction and optimization results into a single preferred engineering option. Although its role is smaller than that of ML, it remains essential, since engineering work does not end with generating alternatives, but selecting one.

Taken together, the literature points to a coherent methodological trajectory. ML has evolved from a promising auxiliary technique into a central analytical framework for chemical engineering and materials discovery. Its value extends beyond pattern recognition or improved prediction to its ability to convert large, diverse, and otherwise underutilized datasets into actionable

engineering knowledge. In chemical engineering, this supports faster and more adaptive representations of complex processes. In materials discovery, it accelerates screening, prediction, and inverse design within search spaces that are too large for exhaustive exploration. Optimization and decision making remain important components, but they are most effective when built upon a strong predictive foundation. For that reason, this thesis is positioned first and foremost within the growing use of ML in chemical engineering, while recognizing its full engineering impact emerges when this predictive core is coupled, more selectively but meaningfully, with optimization and final decision support. This framing preserves the fundamental logic of the field: accurate and physically meaningful learning comes first, followed by optimization and decision making as downstream instruments for turning learned knowledge into improved engineering choices.

1.3 Scope, methodological positioning, and organization of the thesis

The main objective of this thesis is to develop and evaluate data-driven frameworks that use ML as the central predictive layer for chemical engineering applications, while integrating optimization and decision-support methods where needed to translate model outputs into practical engineering guidance. More specifically, the thesis aims to: (i) develop supervised ML models capable of predicting key process and reaction outcomes such as conversion, yield, selectivity, productivity, and cost-related indicators; (ii) examine how data quality, missing values, noise, preprocessing strategies, feature representation, and model selection influence the reliability of predictive and optimization outcomes; (iii) integrate trained ML models with MOO techniques to identify favourable operating conditions under competing objectives; (iv) apply MCDM methods to support transparent selection among Pareto-optimal alternatives; and (v) demonstrate the usefulness of these integrated workflows across process-level, catalytic-process, and reaction-planning case studies. The central research question guiding this thesis is therefore: How can ML-based predictive models be systematically developed and integrated with optimization and decision-making tools to support reliable, interpretable, and practically useful decisions in chemical engineering systems?

The preceding section established the conceptual foundation of this thesis by identifying ML as the central predictive framework for representing complex systems in chemical engineering and materials science, while recognizing MOO and MCDM as essential complementary stages for transforming predictive knowledge into practical engineering decisions. Building on this

foundation, the remainder of the thesis is organized to systematically develop and apply this integrated methodological perspective across a series of interconnected studies. The subsequent chapters progress from theoretical background to methodological development and, finally, to application-driven case studies, demonstrating how data-driven predictive modelling can be combined with optimization and decision-support tools to address increasingly complex chemical engineering problems. In this way, the thesis is structured as a coherent body of work in which ML provides the analytical core, MOO enables the exploration of trade-offs among conflicting objectives, and MCDM supports the selection of suitable solutions for implementation in engineering practice.

Following this introductory chapter, Chapter 2 provides the theoretical and methodological background required for the studies presented in this thesis. It reviews the main ML models applied throughout the work, including artificial neural networks (ANNs), support vector regression (SVR), Gaussian process regression (GPR), k-nearest neighbours (KNN), decision trees (DT), bagging, random forests (RF), extremely randomized trees (Extra Trees or ET), and eXtreme Gradient Boosting (XGBoost). In addition, the chapter introduces the fundamentals of MOO and MCDM, with particular emphasis on the non-dominated sorting genetic algorithm II (NSGA-II) as the optimization solver used to identify Pareto-optimal solutions and the net flow method (NFM) as the MCDM technique adopted to rank and select preferred alternatives. This chapter, therefore, establishes the analytical basis upon which the later chapters are developed.

Chapter 3 presents an integrated framework for informed decision-making in chemical engineering, using the styrene production reactor as a case study. In this chapter, several ML models are developed to predict key process responses, and their predictive performance is systematically evaluated and compared. The selected trained models are then coupled with MOO, with NSGA-II employed to generate Pareto-optimal operating solutions for competing process objectives. Subsequently, NFM is applied as a MCDM tool to rank these non-dominated alternatives and support the final selection. This chapter demonstrates how the integration of ML, MOO, and MCDM forms a structured and effective framework for process analysis, optimization, and engineering decision support.

Chapter 4 addresses data quality challenges through the development of an autoassociative neural network (AANN) for missing-data imputation, again using the styrene production process

as a case study. The chapter investigates the ability of the proposed model to reconstruct missing process variables under different dataset sizes, noise levels, and missing-data scenarios. Since the reliability of any data-driven framework depends strongly on the quality and completeness of the underlying data, this chapter makes an important methodological contribution by strengthening the data-preprocessing foundation required for robust ML applications in chemical engineering.

Chapter 5 focuses on the optimal design of the catalytic conversion of sulfur dioxide to sulfur trioxide through a data-driven framework that integrates ML, MOO, and MCDM. An ANN model is developed to represent the catalytic conversion process, after which NSGA-II is used to solve the MOO problem and identify Pareto-optimal operating conditions. The resulting non-dominated alternatives are then ranked using NFM, providing a transparent and systematic basis for selecting preferred solutions. By applying the framework to different catalyst systems, the chapter illustrates the benefits of combining predictive modelling, optimization, and ranking methods in catalytic process design.

Chapter 6 extends the sulfur dioxide to sulfur trioxide case study by addressing the practical and methodological challenges associated with building ML models for catalytic processes using literature-derived data. In addition to model development, the chapter examines the effects of different preprocessing strategies, feature engineering choices, and data transformation techniques on predictive performance and optimization outcomes. NSGA-II is again used as the MOO solver, while NFM is employed to rank Pareto-optimal solutions. Beyond presenting technical results, this chapter offers broader insights into the challenges, limitations, and recommended practices for developing robust and reliable ML-based catalytic design frameworks for catalytic process design.

Chapter 7 examines the application of ML for predicting reaction conversion and yield in organic substrate systems under different catalysts and operating conditions. In this chapter, predictive models are developed to estimate reaction outcomes and are subsequently integrated with a genetic algorithm (GA) to identify suitable catalysts for reactions in which one functional group is intended to react selectively while another must remain protected. In this context, catalyst selection is treated as a reaction-specific decision problem. To support interpretation of the modelling results, k-means clustering is employed to categorize operating-condition patterns and identify meaningful groupings among reaction conditions. In addition, a web-based application is developed to translate the proposed framework into a practical decision-support tool that assists

chemists in selecting suitable catalysts for specific reactions. Overall, this chapter illustrates how ML-based prediction, optimization, and data-driven categorization can be combined to support decision-making in catalytic reaction design.

Finally, the thesis concludes with a general summary that synthesizes the principal findings and highlights the methodological and practical contributions of the research. Taken together, the chapters form a coherent progression from foundational theory to advanced applications, demonstrating how ML can function as the predictive core of modern chemical engineering analysis, while MOO and MCDM provide the tools needed to convert predictive capability into optimized and actionable engineering decisions.

1.4 References

1. Dobbelaere, M. R., Plehiers, P. P., Van de Vijver, R., Stevens, C. V., & Van Geem, K. M. (2021). Machine Learning in Chemical Engineering: Strengths, Weaknesses, Opportunities, and Threats. *Engineering*, *7*(9), 1201–1211. https://doi.org/10.1016/j.eng.2021.03.019
2. Venkatasubramanian, V. (2019). The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal*, *65*(2), 466–478. https://doi.org/10.1002/aic.16489
3. Chowdhury, S., Ghosh, A., Acharya, S., Saha, S., Pal, P. K., Roy, S., & Lahiri, S. K. (2026). Transforming chemical process engineering: The role of AI and machine learning in revolutionizing process systems. *The Canadian Journal of Chemical Engineering*, *104*(2), 699–731. https://doi.org/10.1002/cjce.70032
4. Khan, N., & Ammar Taqvi, S. A. (2023). Machine Learning an Intelligent Approach in Process Industries: A Perspective and Overview. *ChemBioEng Reviews*, *10*(2), 195–221. https://doi.org/10.1002/cben.202200030
5. Mowbray, M., Savage, T., Wu, C., Song, Z., Cho, B. A., Del Rio-Chanona, E. A., & Zhang, D. (2021). Machine learning for biochemical engineering: A review. *Biochemical Engineering Journal*, *172*, 108054. https://doi.org/10.1016/j.bej.2021.108054
6. Aliramezani, M., Koch, C. R., & Shahbakhti, M. (2022). Modeling, diagnostics, optimization, and control of internal combustion engines via modern machine learning techniques: A review and future directions. *Progress in Energy and Combustion Science*, *88*, 100967. https://doi.org/10.1016/j.pecs.2021.100967

7. He, C., Zhang, C., Bian, T., Jiao, K., Su, W., Wu, K.-J., & Su, A. (2023). A Review on Artificial Intelligence Enabled Design, Synthesis, and Process Optimization of Chemical Products for Industry 4.0. *Processes*, *11*(2), 330. https://doi.org/10.3390/pr11020330
8. Torres, A. I., Ferreira, J., & Pedemonte, M. (2025). Machine learning and process systems engineering for sustainable chemical processes–A short review. *Current Opinion in Green and Sustainable Chemistry*, *51*, 100982. https://doi.org/10.1016/j.cogsc.2024.100982
9. Ning, C., & You, F. (2019). Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming. *Computers & Chemical Engineering*, *125*, 434–448. https://doi.org/10.1016/j.compchemeng.2019.03.034
10. Bhosekar, A., & Ierapetritou, M. (2018). Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Computers & Chemical Engineering*, *108*, 250–267. https://doi.org/10.1016/j.compchemeng.2017.09.017
11. McBride, K., & Sundmacher, K. (2019). Overview of surrogate modeling in chemical process engineering. *Chemie Ingenieur Technik*, *91*(3), 228–239. https://doi.org/10.1002/cite.201800091
12. Papadimitriou, I., Gialampoukidis, I., Vrochidis, S., & Kompatsiaris, I. (2024). AI methods in materials design, discovery and manufacturing: A review. *Computational Materials Science*, *235*, 112793. https://doi.org/10.1016/j.commatsci.2024.112793
13. Madika, B., Saha, A., Kang, C., Buyantogtokh, B., Agar, J., Wolverton, C. M., Voorhees, P., Littlewood, P., Kalinin, S., & Hong, S. (2025). Artificial Intelligence for Materials Discovery, Development, and Optimization. *ACS Nano*, *19*(30), 27116–27158. https://doi.org/10.1021/acsnano.5c04200
14. Goldsmith, B. R., Esterhuizen, J., Liu, J.-X., Bartel, C. J., & Sutton, C. (2018). Machine learning for heterogeneous catalyst design and discovery. *AIChE Journal*, *64*(7), 2311–2323. https://doi.org/10.1002/aic.16198
15. Guan, Y., Chaffart, D., Liu, G., Tan, Z., Zhang, D., Wang, Y., Li, J., & Ricardez-Sandoval, L. (2022). Machine learning in solid heterogeneous catalysis: Recent developments, challenges and perspectives. *Chemical Engineering Science*, *248*, 117224. https://doi.org/10.1016/j.ces.2021.117224

16. Rangaiah, G. P., Feng, Z., & Hoadley, A. F. (2020). Multi-Objective Optimization Applications in Chemical Process Engineering: Tutorial and Review. *Processes*, *8*(5), 508. https://doi.org/10.3390/pr8050508
17. Cerda-Flores, S. C., Rojas-Punzo, A. A., & Nápoles-Rivera, F. (2022). Applications of Multi-Objective Optimization to Industrial Processes: A Literature Review. *Processes*, *10*(1), 133. https://doi.org/10.3390/pr10010133
18. Ma, H., Zhang, Y., Sun, S., Liu, T., & Shan, Y. (2023). A comprehensive survey on NSGA-II for multi-objective optimization and applications. *Artificial Intelligence Review*, *56*(12), 15217–15270. https://doi.org/10.1007/s10462-023-10526-z
19. Wang, Z., Li, J., Rangaiah, G. P., & Wu, Z. (2022). Machine learning aided multi-objective optimization and multi-criteria decision making: Framework and two applications in chemical engineering. *Computers & Chemical Engineering*, *165*, 107945. https://doi.org/10.1016/j.compchemeng.2022.107945
20. Wang, Z., Tan, W. G. Y., Rangaiah, G. P., & Wu, Z. (2023). Machine learning aided model predictive control with multi-objective optimization and multi-criteria decision making. *Computers & Chemical Engineering*, *179*, 108414. https://doi.org/10.1016/j.compchemeng.2023.108414
21. Patel, S. P., Gujarathi, A. M., & Vanzara, P. B. (2025). Machine Learning Predictions of Best Optimal Solutions in Evolutionary Multi-Objective Optimization of Industrial Gas Sweetening Process. *2025 International Conference for Artificial Intelligence, Applications, Innovation and Ethics (AI2E)*, 1–5. https://doi.org/10.1109/AI2E64943.2025.10983276
22. Stander, L., Woolway, M., & Van Zyl, T. L. (2025). Surrogate-assisted evolutionary multi-objective optimisation applied to a pressure swing adsorption system. *Neural Computing and Applications*, *37*(2), 739–755. https://doi.org/10.1007/s00521-022-07295-1
23. Malashin, I. P., Martysyuk, D. A., Tynchenko, V. S., Gantimurov, A. P., Nelyub, V. A., & Borodulin, A. S. (2025). Integrating Machine Learning and Multi-Objective Optimization in Biofuel Systems: A Review. *IEEE Access*, *13*, 81983–82002. https://doi.org/10.1109/ACCESS.2025.3568297
24. Azhar, N. A., Radzi, N. A., & Wan Ahmad, W. S. H. M. (2021). Multi-criteria decision making: A systematic review. *Recent Advances in Electrical & Electronic Engineering*

(Formerly Recent Patents on Electrical & Electronic Engineering), *14*(8), 779–801. https://doi.org/10.2174/2352096514666211029112443

25. Liao, H., He, Y., Wu, X., Wu, Z., & Bausys, R. (2023). Reimagining multi-criterion decision making by data-driven methods based on machine learning: A literature review. *Information Fusion*, *100*, 101970. https://doi.org/10.1016/j.inffus.2023.101970
26. Wang, Z., Nabavi, S. R., & Rangaiah, G. P. (2024). Multi-Criteria Decision Making in Chemical and Process Engineering: Methods, Progress, and Potential. *Processes*, *12*(11), 2532. https://doi.org/10.3390/pr12112532

Chapter 2

Background

2.1 Introduction

This chapter presents the methodological background that underpins the research framework developed in this thesis. It first introduces several of the principal ML models considered in this work, with emphasis on their fundamental concepts, learning mechanisms, and predictive structures. In this section, several supervised ML algorithms used for regression are discussed. Although some of these algorithms are applicable to both classification and regression problems, the focus of this work is on regression tasks. These models are discussed as the main ML approaches forming the basis of the developed framework. While additional ML models are employed in later chapters, they are generally derived from, related to, or conceptually connected with the principal models introduced here. Accordingly, this chapter establishes the essential ML foundation required for the subsequent analyses.

The chapter then focuses on MOO, with particular attention to the NSGA-II, which constitutes the primary optimization algorithm used in this thesis. The main principles of MOO are reviewed, followed by a discussion of the structure and operation of NSGA-II and its suitability for solving problems involving competing objectives. In the final part of the chapter, the NFM is presented as the MCDM technique adopted in this research. Its role in evaluating, ranking, and selecting preferred alternatives from the set of optimal solutions is explained in the context of the overall framework. Collectively, the topics covered in this chapter provide the theoretical and methodological basis for the modelling, optimization, and decision-making procedures applied in the following chapters.

2.2 Machine learning models

2.2.1 Artificial neural networks

Artificial neural networks (ANN) are computational models inspired by the structural and functional principles of the human brain [1]. They consist of multiple interconnected processing elements, or neurons, arranged into layers that communicate through weighted connections. Each neuron aggregates its inputs via a weighted sum, transforms the result through a nonlinear activation function, and passes the output forward through the network [2]. A typical ANN

architecture comprises an input layer that receives data, one or more hidden layers for intermediate processing, and an output layer that generates the final result (Figure 2.1), whether it is used for a prediction, classification, or other outcome [3]. Among the various ANN configurations, feedforward neural networks stand out for their straightforward, unidirectional data flow and structural simplicity. These characteristics make them highly suitable for approximating complex, continuous functions—a property known as universal approximation [2,4]. Other widely-used architectures include convolutional neural networks and recurrent neural networks, which are especially prevalent in domains such as computer vision, natural language processing, and time-series analysis [5].

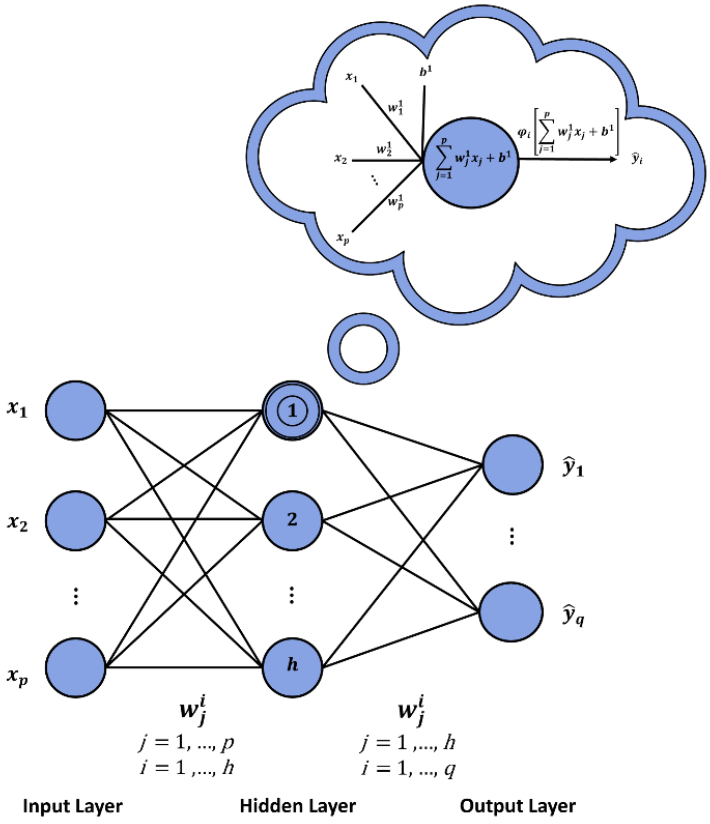


Figure 2.1. A simple representation of a typical ANN structure.

The optimization of ANNs has been extensively investigated across multiple dimensions, which can be broadly categorized into two principal domains: network design and network training. Network design involves several foundational considerations, including the architectural configuration, weight initialization strategies, selection of learning algorithms and activation

functions, input layer specification, and the tuning of hyperparameters such as learning rate and momentum [6]. On the other hand, network training focuses on minimizing a cost function that quantifies the error between predicted and actual outputs, typically using gradient-based optimization algorithms. Among these, backpropagation [7] remains the most widely used method, using gradient descent to iteratively update weights. However, it is prone to challenges such as vanishing gradients and convergence to local minima. To address these issues, more advanced optimization techniques have been used, including Quasi-Newton methods [8], the Levenberg-Marquardt algorithm [9], and adaptive moment estimation (Adam) [10], which are particularly effective in large-scale, high-dimensional contexts. Furthermore, nature-inspired metaheuristic algorithms like genetic algorithms, differential evolution, simulated annealing, artificial bee colony, and particle swarm optimization have been proposed as robust alternatives to traditional training methods [6,11]. These approaches enhance convergence behaviour, improve learning efficiency, and enable ANNs to model complex nonlinear relationships with greater fidelity.

2.2.2 Support vector regression

Support vector regression (SVR) is a supervised learning algorithm tailored for regression tasks, evolving from the foundational principles of support vector machines (SVMs), which were initially developed for binary classification problems. The origins of SVM can be traced back to the 1960s [12,13], with its theoretical framework significantly refined in the 1990s by Vapnik and his colleagues [14,15]. SVMs gained prominence for their effectiveness in handling high-dimensional datasets by constructing optimal hyperplanes that maximize the margin between distinct classes. This approach was later extended to accommodate continuous output predictions, leading to the development of SVR, formally introduced as the ε-SVR model [15].

While SVM classification seeks to identify the hyperplane that best separates distinct classes with maximum margin, SVR repurposes this concept for regression by estimating a function that approximates the relationship between input features and a continuous target variable. The core idea behind SVR is to find a function that deviates from the actual observed outputs by no more than a predefined margin, denoted as ε, for the majority of data points. This tolerance creates a margin of indifference, known as the ε-insensitive tube, within which prediction errors are not penalized [16,17]. SVR frames this regression task as a convex optimization problem, seeking the flattest possible function—effectively the narrowest tube around the regression curve—while

ensuring that most of the training data falls within this tube. Model complexity is controlled by minimizing the norm of the weight vector (‖w‖), which governs the flatness of the regression function. Training instances that lie outside this ε-margin contribute to the loss and are referred to as support vectors. These points play a critical role in shaping the final regression model. The optimization process thus strikes a balance between simplicity and prediction accuracy, resulting in a sparse solution where only a select subset of data, the support vectors, defines the model [18]. Figure 2.2 presents a simple linear representation of an SVR model.

A major advantage of SVR is its kernel-based framework, which allows effective modelling of both linear and nonlinear regression relationships. Through the use of kernel functions, such as radial basis functions and polynomial kernels, SVR can project input data into a higher-dimensional feature space. Within this transformed space, complex relationships among variables can become linearly separable, without the need to compute the transformation explicitly. This property makes SVR particularly effective for capturing nonlinear patterns in high-dimensional datasets [19].

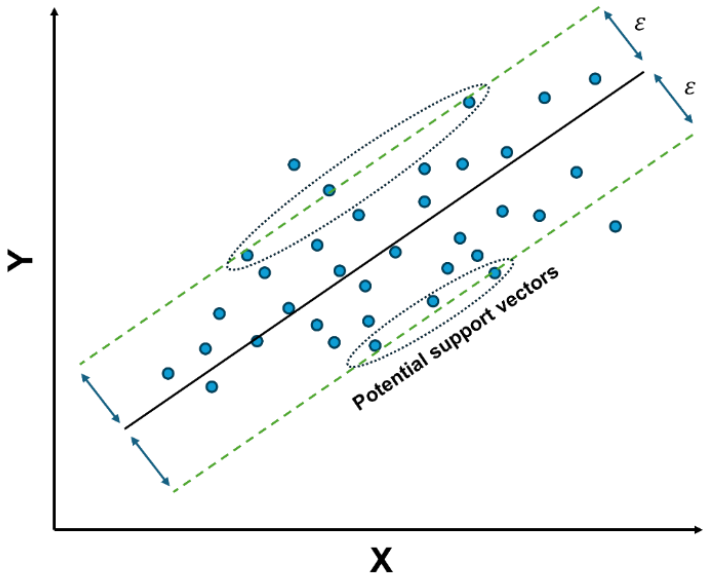


Figure 2.2. Visualization of a simple linear SVR model.

The parameter ε plays a crucial role in SVR by setting the width of the insensitive tube. A smaller ε results in tighter error bounds, often increasing the number of support vectors and leading to a more complex model. Conversely, a larger ε widens the tube, potentially reducing the model

complexity but at the expense of accuracy. Beyond ε, SVR introduces other important hyperparameters. The regularization parameter (C) controls the trade-off between model flatness and the allowance for deviations exceeding ε, effectively balancing complexity and error tolerance. Additionally, the choice of kernel function and its associated parameters determines how input data is mapped into higher-dimensional space, enabling SVR to capture both linear and nonlinear relationships [16,18]. In summary, SVR extends the margin-maximization philosophy of SVM to regression tasks by constructing a function that approximates the data within a flexible margin of error. It retains the core strength of SVM, namely good generalization, sparsity, and robustness in high-dimensional spaces, while being tailored for continuous output prediction. Utilizing convex optimization and kernel methods, SVR offers a robust and versatile framework for regression modelling.

2.2.3 Gaussian process regression

Gaussian process regression (GPR) is a nonparametric, Bayesian approach for supervised learning that not only generates predictions but also principled quantification of uncertainty. Unlike traditional regression methods that fit a predetermined functional form to data, GPR models a distribution over possible functions and updates this distribution as new observations are introduced [20,21]. This probabilistic nature makes GPR particularly well-suited for tasks involving limited data, noisy measurements, or situations where uncertainty estimation is essential [22].

A Gaussian process (GP) is formally defined as a collection of random variables, where any finite subset follows a joint multivariate Gaussian distribution [23,24]. In the context of regression, this means that the outputs corresponding to different input points are assumed to be jointly Gaussian distributed. This assumption allows us to treat function values as random variables and define a prior distribution over functions before observing any data. This prior distribution represents the initial beliefs about the function's shape or behaviour, typically informed by general assumptions such as smoothness, continuity, or stationarity. These assumptions are encoded through the covariance function, or kernel, which quantifies the similarity between outputs based on their input locations [25].

The prior distribution in GP is defined by two key components: the mean function, which specifies the expected value of the function at any input (often set to zero for simplicity), and the kernel function, which determines the degree of correlation between function values at different

input points. A widely used kernel is the radial basis function, which assumes that inputs located closer together in the input space yield more similar outputs. The kernel consists of several hyperparameters, including the length-scale, signal variance, and noise variance, which control the smoothness and variability of the modelled function. These hyperparameters are typically estimated by maximizing the marginal likelihood of the observed data, striking a balance between accurately fitting the data and avoiding excessive model complexity [25,26].

As new data are observed, the prior distribution over functions in GPR is updated to form a posterior distribution. This posterior distribution reflects the refined beliefs about the underlying function, incorporating the evidence provided by the observed data [27]. Importantly, the posterior mean represents the predicted value at a given input, while the posterior variance provides a measure of confidence in that prediction. Thanks to its Bayesian framework, GPR explicitly models uncertainty, reducing it in regions where data are available and maintaining higher uncertainty in areas with sparse or no observations [28].

This prior–posterior framework enables GPR to interpolate between known data points in a smooth and statistically consistent manner, without assuming a predefined form for the target function. It also allows predictions to degrade gracefully in regions of uncertainty, making the method particularly valuable for scientific applications where reliability and interpretability are as important as accuracy [25].

Despite its strengths, GPR faces practical limitations, most notably its computational cost. Training requires inverting a covariance matrix whose size grows with the number of data points, resulting in a time complexity of $O(n^3)$, where n is the number of data points. As a result, GPR becomes inefficient for large-scale datasets. To address this limitation, researchers have developed various sparse and approximate methods that significantly reduce the computational burden [29].

Figure 2.3 presents a comparison between the prior and posterior distributions of a GPR model. In the left panel, 20 sample functions are drawn from the GP prior distribution, which assumes a zero-mean function and encodes prior uncertainty solely through the kernel function, since no data has been observed at this stage. By contrast, the right panel displays the posterior distribution after conditioning on five observed data points, shown as black markers. As expected, the posterior mean (dashed black line) aligns more closely with the observed data, while predictive uncertainty (gray shaded region) is significantly reduced in regions surrounding those

observations. The red triangle denotes a reference input point consistently highlighted in both panels.

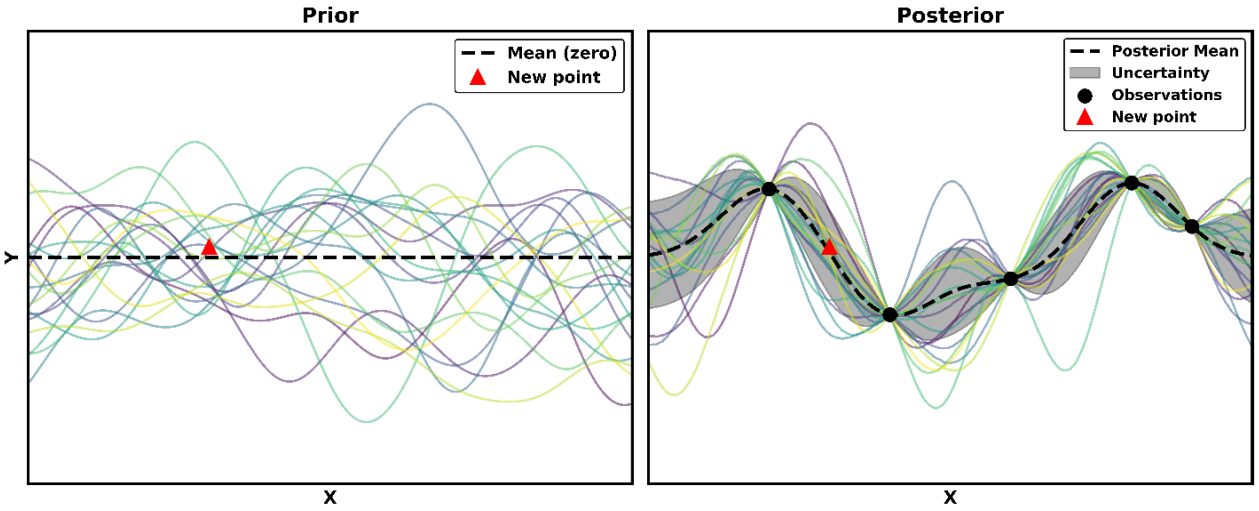


Figure 2.3. Comparison of the GP prior and posterior distributions. The left panel shows random function samples from the prior with zero mean. The right panel illustrates the posterior distribution after conditioning on five observations, showing reduced uncertainty and updated function samples.

2.2.4 K-nearest neighbours

K-nearest neighbours (KNN) is a well-established method in supervised ML. It is a non-parametric, instance-based learning algorithm commonly used for both classification and regression tasks [30]. Unlike model-based approaches that build an explicit function during training, KNN is categorized as a lazy learning algorithm, meaning it performs minimal computation during training and defers most processing to the prediction phase. KNN operates on the intuitive principle of similarity, assuming that data points with similar features tend to exist near each other in the feature space. In the context of regression, KNN predicts a continuous outcome for a new data point by averaging the outcomes of its k nearest neighbours in the training data [31].

The prediction process begins by quantifying the similarity, or "closeness", between data points using a chosen distance metric. Euclidean distance is most commonly employed, although alternatives like Manhattan or Minkowski distances may also be used. Once distances are computed, the algorithm identifies the k closest training instances to the new input. The final prediction is then obtained by averaging the target values of these neighbours (Figure 2.4). In some

cases, a distance-weighted average is applied, giving greater influence to neighbours that are closer to the input and less to those farther away [31,32].

A key hyperparameter in KNN regression is *k*, which determines the number of neighbours considered when making predictions. This parameter plays a critical role in model performance: a smaller *k* can capture finer local structures in the data but may lead to overfitting, especially in noisy datasets. In contrast, a larger *k* yields smoother, more generalized predictions and enhances robustness to noise, though it may lead to underfitting by oversimplifying complex patterns. One of the major strengths of KNN regression lies in its flexibility. As a non-parametric method, it makes no assumptions about the underlying data distribution or functional form of the relationship between inputs and outputs. This enables KNN to effectively model complex, nonlinear relationships, making it a valuable tool in scenarios where the data structure is intricate or poorly understood [33].

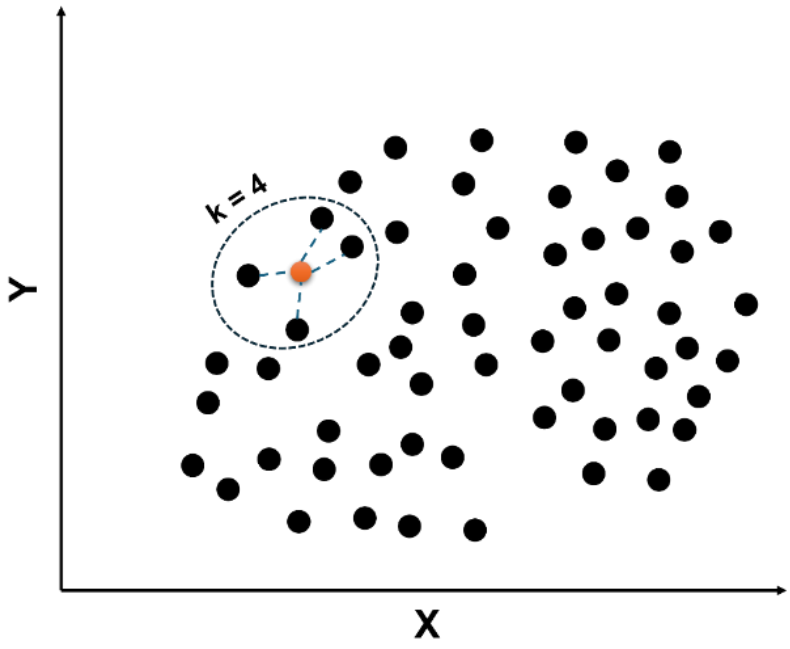


Figure 2.4. A simple representation of KNN.

2.2.5 Decision tree

Decision tree regression (DT) is a widely used non-parametric method for predicting continuous variables. Rooted in ML theory and formalized through the CART (Classification and Regression

Trees) framework by [34], DT models a hierarchical, tree-like structure to partition the input space into distinct regions based on feature values.

The model begins at the root node, which contains the entire dataset. The model then recursively splits it into branches using binary decisions. At each step, the algorithm selects the feature and split point that results in the greatest reduction in prediction error, typically measured by minimizing the sum of squared deviations from the mean within each resulting subset. This top-down, greedy approach continues until a predefined stopping criterion is met, such as a minimum number of samples per node or a specified tree depth. Each terminal node, or leaf, represents a region in the feature space and holds a constant prediction value, which is usually the average of the target variable within that node. This piecewise approach enables DT to model complex and nonlinear relationships without requiring prior assumptions about the data distribution [35–37].

Figure 2.5 illustrates the manner in which a DT partitions a feature space and maps this partitioning through a hierarchical tree structure. In the left panel, the 2D feature space is recursively divided into five rectangular regions (Y_1 to Y_5) using axis-aligned splits based on the input variables X_1 and X_2. Each region groups data points with similar target values. On the right, this special partitioning is represented as a binary tree. Each internal node performs a decision based on a threshold (e.g., $X_1 \leq a$), while each leaf node predicts a constant output value (Y_1 to Y_5). The tree structure reflects the step-by-step decision process used to assign a prediction to any new data point by following the appropriate path from the root to the appropriate leaf.

One of the key strengths of DTs is their interpretability [38]. Their hierarchical structure closely mimics human decision-making, enabling users to follow the prediction path from root to leaf. This transparency makes DTs intuitive and easy to understand. Additionally, a DT performs implicit feature selection during the splitting process by prioritizing the most informative variables and ignoring those with little predictive value. This not only enhances interpretability but also reduces dimensionality and mitigates the "curse of dimensionality," improving computational efficiency [39]. However, DTs are prone to overfitting when allowed to grow too deep, as they may capture noise in the training data rather than the underlying patterns. To address potential overfitting, pruning is applied after the full tree is built [40]. Pruning involves removing branches

that do not significantly improve performance, using a validation set and a cost-complexity measure to balance model accuracy with simplicity.

DT models require minimal data preprocessing—no feature scaling or normalization is needed—and they are robust to a wide range of data types [41]. In summary, a DT offers a flexible, interpretable, and efficient approach for modelling complex relationships in data. Its ability to manage nonlinear relationships, select relevant features, and deliver clear, interpretable decision logic makes it an invaluable asset for various regression tasks.

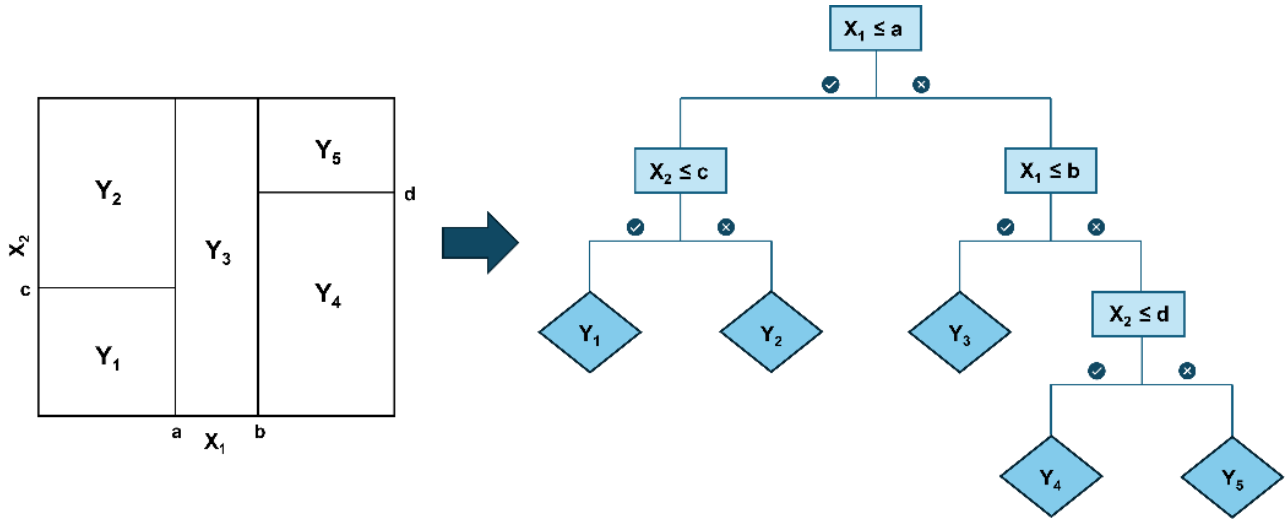


Figure 2.5. Partitioning of the input space by a DT (left) and its corresponding hierarchical decision structure (right), where each split is based on feature thresholds and each leaf predicts a constant output.

2.2.6 Bagging

Bagging regressor (Bagging), short for bootstrap aggregating regressor, is an ensemble learning method introduced to improve the predictive performance of machine learning models by reducing variance and enhancing stability [42]. This technique is particularly effective for algorithms that are highly sensitive to fluctuations in the training data, such as DTs. The fundamental principle of Bagging lies in generating multiple versions of a predictive model by training each one on a different subset of the original data. These subsets are created through bootstrap sampling, a method in which data points are randomly drawn with replacement, resulting in diverse training sets that may include repeated instances while omitting others [43].

Each base learner, typically a fully-grown decision tree, is independently trained on its respective bootstrapped dataset of the original dataset. These trees are left unpruned, allowing them to capture complex patterns within the data, albeit at the cost of increased variance. To mitigate this, the predictions from all models are aggregated—most commonly through averaging in regression tasks—which helps smooth out individual fluctuations and results in a more stable and generalizable final model [44]. This aggregation step is grounded in the statistical principle that the variance of the mean of multiple estimators is lower than that of any individual estimator, provided the estimators are not perfectly correlated. Figure 2.6 presents the architecture of Bagging, in which multiple DTs are trained in parallel on different bootstrapped samples drawn with replacement from the original dataset. Each tree generates its own prediction based on the subset it was trained on, and these individual outputs are subsequently aggregated, most often through averaging, to yield a final prediction.

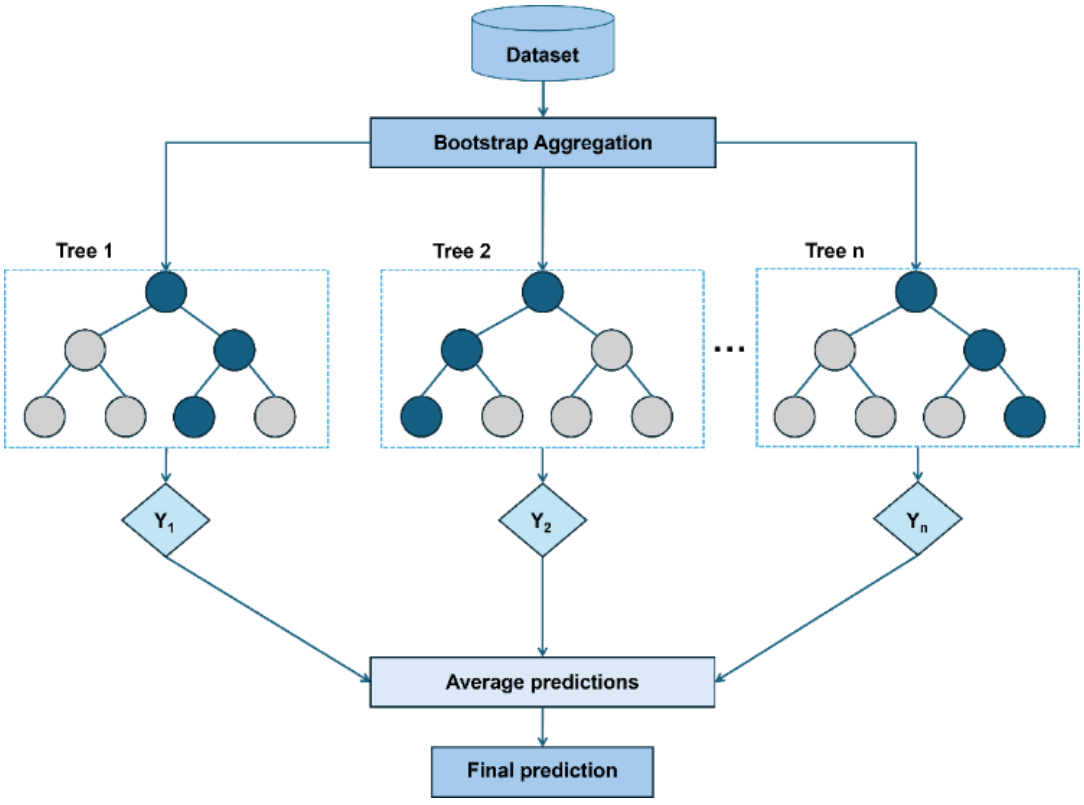


Figure 2.6. Schematic representation of the Bagging regressor. The model generates multiple bootstrapped training datasets from the original data and trains a separate decision tree on each subset. The individual predictions are then averaged to produce a final, robust, and low-variance regression output.

Bagging is founded on the principle that the randomness introduced via bootstrapped sampling leads to a diverse set of models, each capturing different aspects of the data. When these models are combined, typically through averaging, their individual errors tend to offset one another, especially when the underlying base learners are uncorrelated or weakly correlated. This ensemble method not only reduces overfitting compared to a single high-variance model but also increases predictive accuracy without increasing bias [45]. In practice, the performance of Bagging depends on several factors, including the number of base estimators and the nature of the base model itself. Typically, DTs are employed due to their sensitivity to data variations, making them ideal candidates for variance reduction through bagging. The approach also allows for out-of-bag (OOB) estimation, where approximately 37% of the data not included in a particular bootstrap sample is used to evaluate model performance without requiring a separate validation set [46]. In conclusion, Bagging offers a scientifically robust approach to ensemble learning by leveraging the power of statistical resampling and model averaging. Through its capacity to reduce variance while maintaining model complexity, it serves as an effective strategy for enhancing the accuracy and reliability of regression models, particularly when applied to inherently unstable learners.

2.2.7 Random forest

The random forest (RF) regressor is a powerful ensemble learning method that builds upon the principles of DTs and the Bagging method. Like Bagging, RF constructs multiple DTs using bootstrapped samples of the original dataset and combines their outputs through averaging to produce a robust and low-variance prediction. However, RF introduces an additional layer of randomness that sets it apart: instead of considering all predictor variables at each decision node, it selects a random subset of features to determine the optimal split. This added randomness reduces the decorrelation among individual trees in the ensemble, thereby minimizing overfitting and enhancing the model's ability to generalize to unseen data [48,49].

The key innovation of the RF approach lies in this dual-randomization strategy (both in data sampling and in predictor selection), making it especially effective for handling high-dimensional and noisy datasets. Each tree in the forest is grown to its full depth without pruning, enabling the model to capture intricate patterns and interactions within the data. While fully grown trees tend to exhibit high variance, their ensemble average balances this variance, resulting in a model with both low bias and low generalization error [46,50]. Additionally, the algorithm measures the

importance of each feature by evaluating the decrease in prediction accuracy when the values of that feature are permuted. This capability makes RF particularly valuable in complex, high-dimensional problems where interpretability and variable selection are essential [48].

Although RF is considered a "gray box" model, less transparent than individual DTs, it retains a meaningful level of interpretability through its variable importance metrics. It avoids the extreme complexity of ANNs while still delivering strong predictive power [51]. Moreover, the method relies on relatively few hyperparameters—primarily the number of trees in the forest and the number of features considered at each split—making it both user-friendly and computationally efficient [52]. As a result, RF has become one of the most widely used and reliable tools in modern predictive modelling, especially in domains where both accuracy and robustness are critical.

Figure 2.6, which illustrates the architecture of Bagging, also effectively represents the overall procedure of RF. Both models share the same ensemble framework based on training multiple DTs on bootstrapped samples and aggregating their outputs. However, it should be noted that the figure does not explicitly capture RF's additional mechanism of random feature selection at each split, which plays a crucial role in reducing the correlation among trees and enhancing the generalization performance.

2.2.8 Extremely randomized trees

The extremely randomized trees regressor (Extra Trees or ET) is a powerful ensemble learning technique that extends the RF algorithm by injecting even greater randomness into the tree-building process. Like Bagging and RF, the ET algorithm builds an ensemble of DTs and aggregates their predictions to produce a stable and robust final output. Figure 2.6, which illustrates the overall procedure of the Bagging and RF regressors, also effectively captures the structural framework of the ET regressor. All three methods share a common strategy: training multiple trees in parallel and combining their outputs to reduce model variance. However, ET introduces distinct methodological changes that set it apart in terms of both design and predictive performance [54,55].

In ET, each DT is trained not on the full original training dataset but on a bootstrapped subset, as in Bagging and RF. This strategy reduces the bias introduced by resampling and ensures that all trees have access to the complete set of training examples [56]. More critically, ET introduces maximal randomness during the split selection process. While RF selects the best split from a randomly chosen subset of features based on criteria such as information gain or variance

reduction, ET takes this process a step further by also selecting the split threshold at random for each candidate feature. Among these randomly generated thresholds, the best is chosen to split the node. This results in greater diversity among the trees, which reduces correlation and variance across the ensemble, though potentially at the cost of a slight increase in bias [53,57,58].

Even with their increased randomness, ET models effectively capture intricate nonlinear relationships within the data. Like RF and Bagging, they support OOB evaluation when bootstrap sampling is enabled. Additionally, they provide feature importance metrics that highlight the most influential variables in the prediction process. Although their accuracy may not always rival that of highly specialized or finely tuned models for certain applications, ETs are valued for their nearly parameter-free, fast to compute, and capable of handling high-dimensional, noisy data with minimal preprocessing [57].

In conclusion, the ET stands as a highly randomized and computationally efficient alternative within the family of tree-based ensemble methods. By combining the structural ensemble approach illustrated in Figure 2.6 with full-dataset training and randomized split selection, ET achieves a favourable balance between accuracy, speed, and robustness—making it a valuable off-the-shelf method for a wide range of regression problems.

2.2.9 eXtreme gradient boosting

The eXtreme gradient boosting (XGBoost) regressor is an advanced ensemble learning algorithm that builds upon the foundational principles of DTs while introducing a highly optimized gradient boosting framework. In contrast to Bagging and RF, which train multiple decision trees in parallel and aggregate their outputs to reduce the variance, XGBoost adopts a sequential learning approach where trees are added one at a time [60]. Each successive tree is trained to correct the residual errors of the ensemble formed by the previous trees. This additive model refinement enables XGBoost to produce highly accurate predictions and makes it particularly effective for complex, nonlinear regression tasks [61].

At its core, XGBoost constructs an ensemble of regression trees, where each tree contributes a weighted prediction that cumulatively improves the model's overall accuracy. Unlike RF, which selects splits based on local criteria and randomness, XGBoost formulates tree construction as a numerical optimization problem. It utilizes both the first and second derivatives of a differentiable loss function, such as MSE, via a second-order Taylor expansion, allowing for precise control of

model updates. This formulation enables the algorithm to choose optimal tree structures and leaf weights at each stage, maximizing the reduction in prediction errors [59].

One key feature that distinguishes XGBoost is the inclusion of a regularization term in its objective function. This term penalizes model complexity by limiting the number of leaves and constraining the magnitude of leaf weights, effectively reducing overfitting and enhancing generalization. As a result, XGBoost remains highly robust, even when applied to high-dimensional spaces and when handling noisy or sparse data [59,62].

Beyond its boosting strategy and regularization framework, XGBoost incorporates a range of hyperparameters that significantly impact model performance. Among the most critical are the number of trees, the maximum depth of each tree, the learning rate (which determines the contribution of each tree), and subsampling ratios for both rows and features. These parameters provide substantial flexibility, enabling practitioners to fine-tune the model to balance bias, variance, and computational efficiency [63,64].

Figure 2.7 illustrates the underlying architecture of the XGBoost, which differs from traditional ensemble methods by adopting a sequential, additive learning process. In this framework, multiple regression trees are constructed in succession, with each new tree trained to minimize the residual errors left by the preceding trees. As illustrated, each tree contributes a prediction (Y_1, Y_2, ..., Y_n), and the final output is obtained by summing these individual predictions. This iterative boosting strategy allows the model to continuously refine its performance by focusing on the errors of previous learners.

In summary, the XGBoost represents a powerful and efficient evolution of tree-based ensemble learning. Its sequential boosting process, gradient-driven optimization, and embedded regularization enable it to consistently deliver state-of-the-art performance across a wide range of supervised learning tasks. Widely adopted in both academic research and competitive ML, XGBoost is now regarded as one of the most effective tools for developing high-accuracy regression models, especially in settings that demand both precision and scalability [61].

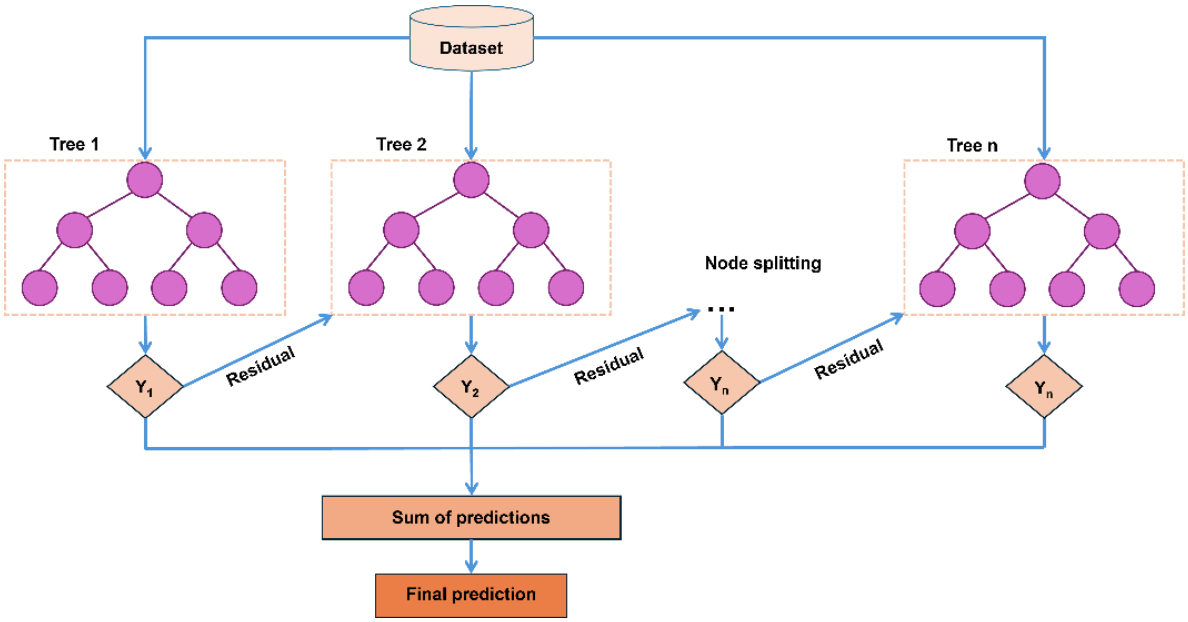


Figure 2.7. Sequential tree-building process in XGBoost where each tree corrects the errors of the previous ones, and the final prediction is obtained by summing all outputs.

Table 2.1 presents a comprehensive overview of the nine supervised ML regression models examined in this study. For each model, the table presents a brief description of its underlying methodology, along with its primary strengths and potential limitations. This comparative summary clarifies the distinguishing features of each approach and supports informed model selection tailored to specific application requirements and data characteristics.

Table 2.1. Brief description, advantages, and disadvantages of each model used in this work.

Model	Description	Advantages	Disadvantages
ANN	Computational models inspired by the brain, composed of layers of interconnected neurons that approximate nonlinear functions	Universal function approximator, good for nonlinear relationships, flexible architectures, strong performance in many tasks, generally suitable for deterministic optimization because of its smooth nonlinear mapping	Requires tuning, prone to overfitting, slow training, less interpretable, sensitive to initial weights, nonconvex structure may lead to local optima in deterministic optimization

SVR	Regression method based on the support vector machine concept, using an ε-insensitive loss to fit a function within a margin	Effective in high-dimensional spaces, good generalization, kernel trick enables nonlinear regression, often suitable for deterministic optimization when smooth kernels are used	Sensitive to hyperparameters (C, ε, kernel), slow with large datasets, limited interpretability, kernel complexity may complicate deterministic optimization
GPR	Bayesian non-parametric method modelling a distribution over functions, providing prediction mean and uncertainty estimates	Provides uncertainty estimation, strong in small/noisy datasets, flexible kernel choice for modelling structure, smooth predictions are well-suited to deterministic optimization	High computational cost ($O(n^3)$), less suitable for large datasets, kernel/hyperparameter selection crucial, repeated evaluations may become expensive in optimization
KNN	Instance-based method that predicts output by averaging the k nearest neighbours based on a distance metric	Simple to implement, no training phase, flexible for complex patterns, no distribution assumptions	Sensitive to noise/outliers, computationally expensive for large datasets, lacks interpretability, non-smooth local prediction structure is not well suited to deterministic optimization
DT	Tree-based model that splits the feature space based on threshold rules to minimize prediction error	Interpretable, no feature scaling required, handles mixed data types, performs implicit feature selection	Prone to overfitting without pruning, instability to small data changes, biased toward dominant features, piecewise-constant prediction structure limits suitability for deterministic optimization
Bagging	Ensemble of models trained on bootstrapped samples,	Reduces variance, increases stability, supports OOB evaluation, well-suited for	Increased training time, model complexity, less interpretable than single trees, ensemble tree

	reducing variance through averaging predictions	high-variance models like DTs	structure remains non-smooth for deterministic optimization
RF	Improved bagging method using random feature subsets at each split to enhance generalization and reduce overfitting	High accuracy, robust to overfitting, ranks feature importance, performs well on many real-world problems	Less interpretable, may not perform well with extrapolation, requires tuning number of trees and features, discontinuous tree-based predictions reduce suitability for deterministic optimization
ET	Variant of RF using full data (not bootstrapped) and random thresholds for splits, increasing diversity among trees	Fast, highly randomized, robust to noise and high-dimensional data, minimal tuning needed	Slightly more biased than RF, randomness may hurt accuracy in some settings, non-smooth prediction surface is less convenient for deterministic optimization
XGBoost	Gradient boosting framework that sequentially builds trees to minimize loss using first and second derivatives	High accuracy, fast and scalable, regularization to prevent overfitting, widely adopted in competitions, can still be usable in optimization because of its strong predictive capability	Complex tuning, less interpretable, sensitive to noisy data if not regularized properly, tree-based piecewise structure is less straightforward for deterministic optimization than smooth surrogate models

2.3 Multi-objective optimization

MOO problems, which typically have multiple optimum solutions, are optimization problems that must satisfy multiple objectives [28]. The solution is defined as a vector of decision variables $X = \{x_1, x_2, x_3, \dots, x_n\}$ (variable decision space) that optimizes the vector of objective functions $F(X) = \{f_1(X), f_2(X), f_3(X), \dots, f_m(X), \dots, f_M(X)\}$ (objective function space) within a feasible region of solutions which is subjected to equality $h_g(X)$ or inequality $g_h(X)$ constraints. The decision variables have also lower limits (x_i^{LL}) and upper limits (x_i^{UL}), which constrain the search space for each variable [65–67]. This process can mathematically be formulated in Equation (2.1).

$$\text{Min } F(X) = \{f_1(X),\ f_2(X),\ f_3(X), \dots, f_m(X), \dots, f_M(X)\},$$
$$m = 1, 2, 3, \dots, M$$
Subject to :
$$h_k(X) = 0\,, \quad g = 1, 2, 3, \dots, G \tag{2.1}$$
$$g_h(X) \leq 0\,, \quad h = 1, 2, 3, \dots, H$$
$$x_i^{LL} \leq x_i \leq x_i^{UL}\,, \quad i = 1, 2, 3, \dots, n$$

where M is the number of objectives. Each of the objective functions $f_m(X)$, also referred to as criteria, payoff functions, cost functions, or value functions, can be either minimised or maximised [68]. However, we can transform a maximisation problem into a minimization problem using the duality principle by multiplying -1 to the objective function [69]. When objectives in Equation (2.1) conflict with one another, the results can be quite different. For example, improving one objective may worsen the other objectives. Therefore, the concept of Pareto optimality can address this issue [69,70]. The approach that is most frequently used to evaluate solutions is the Pareto dominance relationship, which generates a set of optimal solutions between the objectives rather than presenting a single optimal solution. These solutions are also known as non-dominated or Pareto-optimal solutions, and it is up to the problem's decision maker to select the best one based on his/her preferences [65]. As an example, two objectives are shown schematically in Figure 2.8a, along with an accompanying Pareto front. Point D, which is a Pareto solution and one of the best solutions, is located on the border known as the Pareto front. Point C is another solution, but it is not considered as optimal as we can find another solution that is better for the two criteria. Point B is an impossible and unreachable point, whereas Point A is the utopia. Figure 2.8b illustrates the mapping between the decision space (three variables) and objective space (two objectives).

The overall goal of a MOO technique can be summarized as finding a set of solutions that lie in the optimal Pareto front and assembling a set of solutions that are sufficiently varied to reflect the Pareto front correctly. Algorithms for an evolutionary MOO attempt to achieve both of the aforementioned subjects. Figure 2.9 illustrates how various separate parametric single-objective algorithms may generate numerous Pareto-optimal solutions. The Pareto-optimal front correlates to the overall optimal solutions for several scalarized objectives. Several challenges, such as infeasible regions, local optimum solutions, and smooth regions of the objective function, must be surmounted by algorithms during an optimization process in order to converge to the global

solution. Furthermore, an optimization process must also be performed at a sensible computational cost due to real restrictions. In order to address the aforementioned challenges consistently and swiftly, an algorithm must find a reasonable equilibrium between the extent of these processes [72,73].

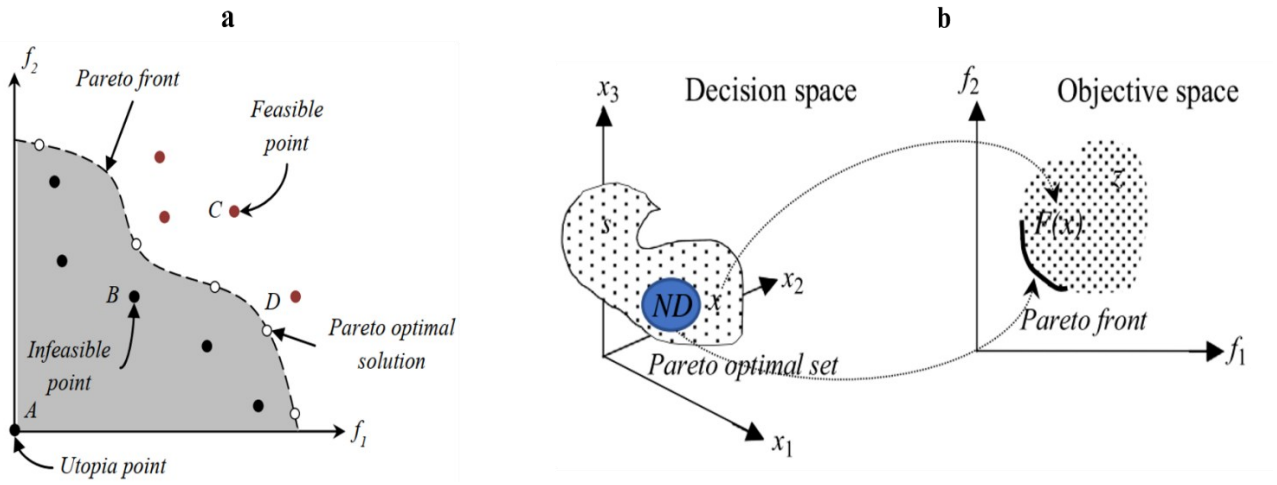


Figure 2.8. Diagram illustrating Pareto optimal solutions. a) Feasible, infeasible, and optimal solutions [68], b) decision and objective spaces [71].

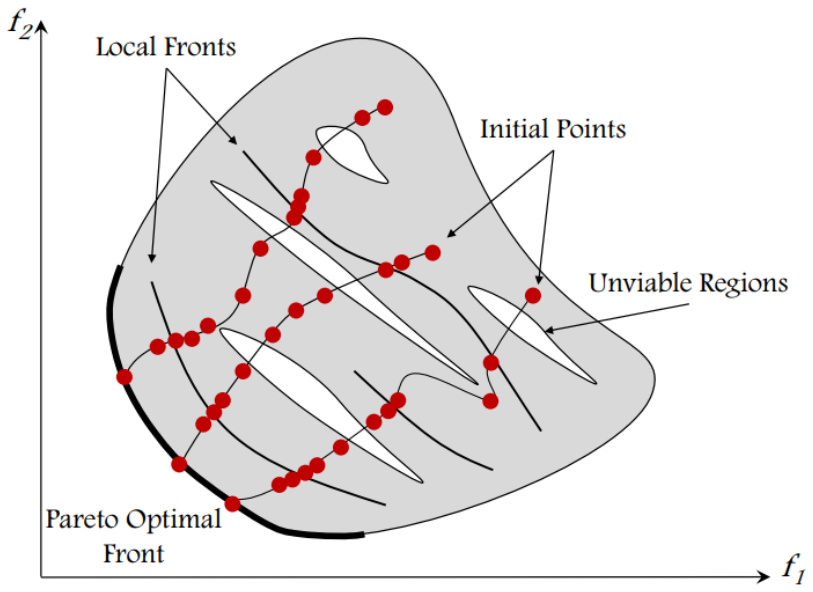


Figure 2.9. Zones of a design problem [72].

There are plenty of methods that can be used to solve MOO problems. The global criterion technique converts the complexity of plural problem optimization into a single problem

optimization by reducing the distance between various reference locations and feasible destination regions [70,74]. A weighted vector is used in the weighted-sum technique to combine all problems into a single problem [75,76]. Typically, the weights are normalized such that their sum is equal to one.

Although the weighted-sum technique is straightforward and simple to apply, it is plagued by two main issues. First, selecting weights for problems of varying magnitudes is challenging. As a result, the process of coming up with a trade-off solution will be biased. Second, this method cannot be used if the optimized multiple problem is non-convex [77]. The ε-constraint technique is applied to solve challenges in plural non-convex problems. While the other problems are converted into limits, the ε-constraint technique only optimizes one problem [77–79]. In the lexicographic approach, decision-makers are requested to control objective functions by relying solely on their absolute interests. Each objective is optimized separately in accordance with its relative significance. If only one solution is returned after optimizing the first objective—the most important one—then the solution is the optimal one. On the other hand, optimization will proceed on the second objective while imposing new constraints on the first objective's solution. This pattern continues until the last objective is achieved [77,80]. Another method is goal programming, in which the decision-maker selects the desired degree of the objective function. The purpose is to maximise the objective function while maintaining the desired level [81]. One of the popular methods is the multi-objective evolutionary algorithm (MOEA), which is a stochastic technique. MOEAs are utilised to find the optimal Pareto solutions for particular problems, much like other optimization methods. The majority of MOEAs in existence employ the idea of dominance. With the exception of the use of dominance relationships, the MOEA's optimization process is very comparable to that of Evolutionary algorithms (EAs). To choose a better solution for the generation of the hereditary population, the objective value is specifically computed for each individual at each iteration and used to establish the relationship of dominance in the population. To create the population for the following generation, this population may be merged with parent groups. Additionally, the presence of the objective space might allow MOEAs the freedom to use some traditional support techniques, such as niching [82]. In comparison to more traditional techniques, MOEAs have several discernible benefits. They are effective at finding a global optimum, are not susceptible to the non-smoothness of the objective functions and have a built-in resilience to noise. MOEAs, same as other approaches, have limitations. One significant drawback is that a large

number of solutions are taken into account, which may result in redundancy. This also implies that the method for resolving MOO problems is a time-consuming technique [83].

NSGA-II is one of the widely used MOEA methods, which seeks to solve a MOO problem and find Pareto-optimal solutions. This method uses an elitist principle, an explicit diversity mechanism, and emphasises non-dominated solutions. The parent population (P_t) and the standard genetic operators (crossover, mutation, etc.) are first used to produce the offspring population (Q_t) at any generation t. The two populations are then merged to create a new population (R_t) with a capacity of $2N$ individuals. The population R_t is subsequently divided into various non-dominance groups. The new population is then filled, one at a time, by points of various non-domination fronts. The first non-domination front is filled first, followed by the points of the second non-domination front, and so forth. Not all fronts can fit in the N spaces accessible for the new population because the total population of R_t is $2N$. All fronts that cannot fit will be removed. There may be more points in the front than open spaces in the new population when the last permitted front is taken into account. Figure 2.10a provides an illustration of this procedure. The points that increase the most the variety of the selected points are chosen rather than randomly eliminating some members from the previous front. The last front's points that could not be completely accommodated are crowded-sorted in ascending order of their crowding distance values, and the points at the top of the ordered list are selected. Figure 2.10b illustrates the calculation of crowding distance [69]. The objective area surrounding point i that is unoccupied by any additional solutions in the population is measured by the crowding distance d_i of point i.

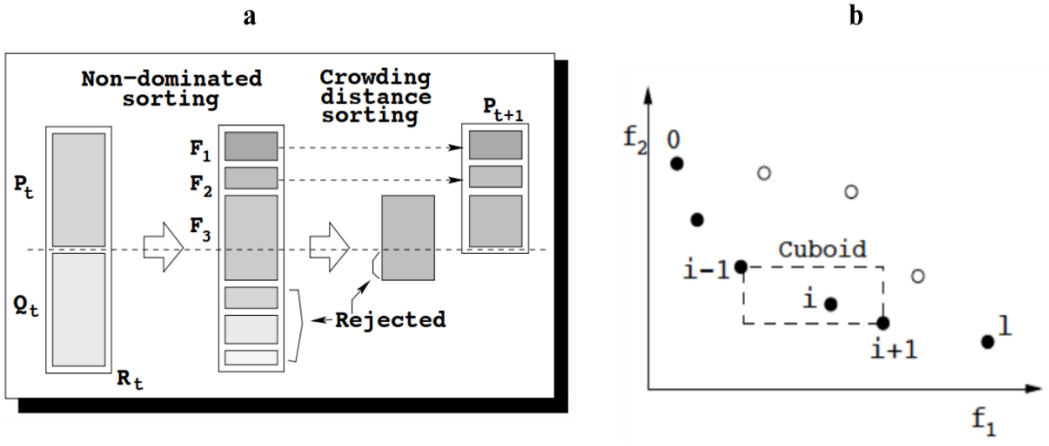


Figure 2.10. a) Procedure of NSGA-II, b) Crowding distance [69].

2.4 Multi-criteria decision-making

Once the Pareto front has been identified using dominance-based criteria, the next critical step is to rank the non-dominated solutions to support final decision-making. This step incorporates the preferences of the decision-maker and applies a systematic evaluation framework to differentiate between competing optimal solutions. In this work, the NFM—a hybrid approach that combines elements of the ELECTRE and PROMETHEE MCDM schemes—is employed to rank the Pareto-optimal set. NFM leverages the strengths of both parent methodologies: it adopts ELECTRE's use of concordance and discordance indices for outranking while incorporating PROMETHEE's net flow scoring logic for solution ranking. Through this integration, NFM enables a more balanced and discriminating evaluation of optimal solutions [84,85].

NFM integrates subjective preferences through four essential parameters that must be defined for each objective function. First, relative weights (W_m) are allocated to each objective, reflecting their importance in the overall evaluation. These weights must collectively sum to one, ensuring proportional contribution across objectives. Second, the indifference threshold (Q_m) is specified, which denotes a range of differences between two solutions' objective values where no strong preference can be asserted. Third, the preference threshold (P_m) indicates the minimum difference in objective values required to favour one solution over another. Finally, the veto threshold (V_m) serves as a cut-off point; if the gap between two solutions for one objective exceeds (V_m), the inferior solution is excluded, regardless of its performance on other criteria. The three thresholds are established individually for each criterion to satisfy Equation (2.2):

$$0 \leq Q_m \leq P_m \leq V_m \tag{2.2}$$

These thresholds play a crucial role in generating pairwise comparison metrics. Using Equations (2.3) to (2.6), the difference $\Delta_m[i,j]$ between the objective of solutions i and j is first computed. Based on this difference, the individual concordance index $c_m[i,j]$ is derived: it takes the value of 1 if the difference between values is less than the indifference threshold for a given criterion, decreases linearly between Q_m and P_m, and becomes 0 when the difference exceeds P_m. The global concordance index $C[i,j]$ is then aggregated across all objectives.

Similarly, the discordance index $D_m[i,j]$ is calculated for each criterion. It takes a value of 0 when the difference falls within the P_m, increases linearly between P_m and V_m, and reaches 1 once

the difference surpasses V_m. These concordance and discordance measures allow for nuanced and discriminating comparisons between pairs of solutions.

$$\Delta_m[i,j] = F_m(j) - F_m(i) \qquad \begin{cases} i \in [1,N] \\ j \in [1,N] \\ m \in [1,M] \end{cases} \tag{2.3}$$

$$c_m[i,j] = \begin{cases} 1 & if \Delta_m[i,j] \leq Q_m \\ \dfrac{P_m - \Delta_m[i,j]}{P_m - Q_m} & if\ Q_m < \Delta_m[i,j] \leq P_m \\ 0 & if\ \Delta_m[i,j] > P_m \end{cases} \tag{2.4}$$

$$C[i,j] = \sum_{m=1}^{M} W_m c_m[i,j] \qquad \begin{cases} i \in [1,N] \\ j \in [1,N] \end{cases} \tag{2.5}$$

$$D_m[i,j] = \begin{cases} 0 & if \Delta_m[i,j] \leq P_m \\ \dfrac{\Delta_m[i,j] - P_m}{V_m - P_m} & if\ P_m < \Delta_m[i,j] \leq V_m \\ 1 & if\ \Delta_m[i,j] > V_m \end{cases} \tag{2.6}$$

Once both indices are known, the outranking matrix $\sigma[i,j]$ is formed using Equation (2.7). This matrix encapsulates the extent to which one solution is judged to outperform another, offering a complete pairwise dominance landscape across the Pareto set.

$$\sigma[i,j] = C[i,j]\left(\prod_{m=1}^{M}[1 - (D_m[i,j])^3]\right) \qquad \begin{cases} i \in [1,N] \\ j \in [1,N] \end{cases} \tag{2.7}$$

To finalize the ranking, the overall score for each solution *i* is calculated using Equation (2.8).

$$\sigma_i = \sum_{j=1}^{M} \sigma[i,j] - \sum_{j=1}^{M} \sigma[j,i] \tag{2.8}$$

This score includes two components: the degree to which solution *i* dominates all others, and the degree to which other solutions dominate solution *i*. The net difference of these two terms yields a comprehensive performance score. Sorting the solutions in descending order based on these scores provides a prioritized ranking, with the top-ranked solution representing the most preferred option within the Pareto domain.

2.5 References

1. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*, 115–133. https://doi.org/10.1007/BF02478259
2. Ojha, V. K., Abraham, A., & Snášel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, *60*, 97–116. https://doi.org/10.1016/j.engappai.2017.01.013
3. Plumb, A. P., Rowe, R. C., York, P., & Brown, M. (2005). Optimisation of the predictive ability of artificial neural network (ANN) models: A comparison of three ANN programs and four classes of training algorithm. *European Journal of Pharmaceutical Sciences*, *25*(4–5), 395–405. https://doi.org/10.1016/j.ejps.2005.04.010
4. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8
5. Shrestha, A., & Mahmood, A. (2019). Review of Deep Learning Algorithms and Architectures. *IEEE Access*, *7*, 53040–53065. https://doi.org/10.1109/ACCESS.2019.2912200
6. Ünal, H. T., & Başçiftçi, F. (2022). Evolutionary design of neural network architectures: A review of three decades of research. *Artificial Intelligence Review*, *55*(3), 1723–1802. https://doi.org/10.1007/s10462-021-10049-5
7. Werbos, P. J. (1994). *The roots of backpropagation: From ordered derivatives to neural networks and political forecasting* (Vol. 1). John Wiley & Sons.
8. Gill, P. E., & Murray, W. (1972). Quasi-Newton methods for unconstrained optimization. *IMA Journal of Applied Mathematics*, *9*(1), 91–108. https://doi.org/10.1093/imamat/9.1.91
9. Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, *11*(2), 431–441. https://doi.org/10.1137/0111030
10. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv Preprint arXiv:1412.6980*. https://doi.org/10.48550/arXiv.1412.6980
11. Abdolrasol, M. G. M., Hussain, S. M. S., Ustun, T. S., Sarker, M. R., Hannan, M. A., Mohamed, R., Ali, J. A., Mekhilef, S., & Milad, A. (2021). Artificial Neural Networks Based

Optimization Techniques: A Review. *Electronics*, *10*(21), 2689. https://doi.org/10.3390/electronics10212689

12. Vapnik, V. N. (1964). A note on one class of perceptrons. *Automat. Rem. Control*, *25*, 821–837.
13. Vapnik, V. N. (1963). Pattern recognition using generalized portrait method. *Automat. Rem. Control*, *24*(6), 774–780.
14. Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144–152.
15. Vapnik, V. (1999). *The nature of statistical learning theory*. Springer.
16. Awad, M., & Khanna, R. (2015). Support Vector Regression. In M. Awad & R. Khanna, *Efficient Learning Machines* (pp. 67–80). Apress. https://doi.org/10.1007/978-1-4302-5990-9_4
17. Drucker, H., Burges, C. J., Kaufman, L., Smola, A., & Vapnik, V. (1996). Support vector regression machines. *Advances in Neural Information Processing Systems*, *9*, 155–161.
18. Zhang, F., & O'Donnell, L. J. (2020). Support vector regression. In *Machine learning* (pp. 123–140). Academic Press. https://doi.org/10.1016/B978-0-12-815739-8.00007-9
19. Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*, 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88
20. Gibbs, M. N. (1998). *Bayesian Gaussian processes for regression and classification* [PhD Thesis, University of Cambridge].
21. Williams, C., & Rasmussen, C. (1995). Gaussian processes for regression. *Advances in Neural Information Processing Systems*, *8*.
22. Boyle, P. (2007). *Gaussian processes for regression and optimisation* [PhD Thesis, Open Access Te Herenga Waka-Victoria University of Wellington]. https://doi.org/10.26686/wgtn.16934869.v1
23. Rasmussen, C. E. (2004). Gaussian Processes in Machine Learning. In O. Bousquet, U. Von Luxburg, & G. Rätsch (Eds.), *Advanced Lectures on Machine Learning* (Vol. 3176, pp. 63–71). Springer. https://doi.org/10.1007/978-3-540-28650-9_4
24. Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian processes for machine learning* (Vol. 2). MIT press.

25. Schulz, E., Speekenbrink, M., & Krause, A. (2018). A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, *85*, 1–16. https://doi.org/10.1016/j.jmp.2018.03.001
26. Wang, J. (2023). An intuitive tutorial to Gaussian process regression. *Computing in Science & Engineering*, *25*(4), 4–11. https://doi.org/10.1109/MCSE.2023.3342149
27. Deringer, V. L., Bartók, A. P., Bernstein, N., Wilkins, D. M., Ceriotti, M., & Csányi, G. (2021). Gaussian Process Regression for Materials and Molecules. *Chemical Reviews*, *121*(16), 10073–10141. https://doi.org/10.1021/acs.chemrev.1c00022
28. Shi, J. Q., & Choi, T. (2011). *Gaussian process regression analysis for functional data* (1st ed.). Chapman and Hall/CRC.
29. Gunnell, L. L., Manwaring, K., Lu, X., Reynolds, J., Vienna, J., & Hedengren, J. (2022). Machine Learning with Gradient-Based Optimization of Nuclear Waste Vitrification with Uncertainties and Constraints. *Process.*, *10*(11). https://doi.org/10.3390/pr10112365
30. Ortiz-Villaseñor, D., Trujillo-Hernández, G., Real-Moreno, O., Castro-Toscano, M. J., Medina-Madrazo, L. D., & Barrera-Román, D. (2025). K-Nearest Neighbors Regression and Applications. In *Exploring Psychology, Social Innovation and Advanced Applications of Machine Learning* (pp. 295–316). IGI Global Scientific Publishing.
31. Halder, R. K., Uddin, M. N., Uddin, Md. A., Aryal, S., & Khraisat, A. (2024). Enhancing K-nearest neighbor algorithm: A comprehensive review and performance analysis of modifications. *Journal of Big Data*, *11*(1), 113. https://doi.org/10.1186/s40537-024-00973-y
32. Srisuradetchai, P., & Suksrikran, K. (2024). Random kernel k-nearest neighbors regression. *Frontiers in Big Data*, *7*, 1402384. https://doi.org/10.3389/fdata.2024.1402384
33. Wekalao, J., & Mandela, N. (2024). Graphene metasurface-based biosensor for direct dopamine detection utilizing surface Plasmon resonance in the terahertz regime with machine learning optimization via K-nearest neighbors regression. *Plasmonics*, 1–29. https://doi.org/10.1007/s11468-024-02570-4
34. Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees* (First). Chapman and Hall/CRC.
35. Hastie, T., Friedman, J., & Tibshirani, R. (2001). *The Elements of Statistical Learning*. Springer. https://doi.org/10.1007/978-0-387-21606-5

36. Rathore, S. S., & Kumar, S. (2016). A Decision Tree Regression based Approach for the Number of Software Faults Prediction. *ACM SIGSOFT Software Engineering Notes*, *41*(1), 1–6. https://doi.org/10.1145/2853073.2853083
37. Xu, M., Watanachaturaporn, P., Varshney, P. K., & Arora, M. K. (2005). Decision tree regression for soft classification of remote sensing data. *Remote Sensing of Environment*, *97*(3), 322–336. https://doi.org/10.1016/j.rse.2005.05.008
38. Balcan, M.-F., & Sharma, D. (2024). Learning accurate and interpretable decision trees. *arXiv Preprint arXiv:2405.15911*. https://doi.org/10.48550/arXiv.2405.15911
39. Bittencourt, H. R., & Clarke, R. T. (2004). Feature selection by using classification and regression trees (CART). *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.
40. Mienye, I. D., & Jere, N. (2024). A survey of decision trees: Concepts, algorithms, and applications. *IEEE Access*, *12*, 86716–86727. https://doi.org/10.1109/ACCESS.2024.3416838
41. De'ath, G., & Fabricius, K. E. (2000). Classification and regression trees: a powerful yet simple technique for ecological data analysis. *Ecology*, *81*(11), 3178–3192. https://doi.org/10.1890/0012-9658(2000)081%5B3178:CARTAP%5D2.0.CO;2
42. Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, *24*(6), 2350–2383. https://doi.org/10.1214/aos/1032181158
43. James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). Tree-Based Methods. In G. James, D. Witten, T. Hastie, R. Tibshirani, & J. Taylor, *An Introduction to Statistical Learning* (pp. 331–366). Springer. https://link.springer.com/10.1007/978-3-031-38747-0_8
44. Bakır, R., Orak, C., & Yüksel, A. (2024). Optimizing hydrogen evolution prediction: A unified approach using random forests, lightGBM, and Bagging Regressor ensemble model. *International Journal of Hydrogen Energy*, *67*, 101–110. https://doi.org/10.1016/j.ijhydene.2024.04.173.
45. Sutton, C. D. (2005). Classification and regression trees, bagging, and boosting. *Handbook of Statistics*, *24*, 303–329. https://doi.org/10.1016/S0169-7161(04)24011-1
46. Prasad, A. M., Iverson, L. R., & Liaw, A. (2006). Newer Classification and Regression Tree Techniques: Bagging and Random Forests for Ecological Prediction. *Ecosystems*, *9*(2), 181–199. https://doi.org/10.1007/s10021-005-0054-1

47. Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324
48. Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random Forests. In C. Zhang & Y. Ma (Eds.), *Ensemble Machine Learning* (pp. 157–175). Springer. https://doi.org/10.1007/978-1-4419-9326-7_5
49. Hastie, T., Tibshirani, R., & Friedman, J. (2009). Random Forests. In T. Hastie, R. Tibshirani, & J. Friedman, *The Elements of Statistical Learning* (pp. 587–604). Springer. https://doi.org/10.1007/978-0-387-84858-7_15
50. Segal, M. R. (2004). *Machine learning benchmarks and random forest regression*. UCSF: Center for Bioinformatics and Molecular Biostatistics. https://escholarship.org/uc/item/35x3v9t4
51. Fawagreh, K., Gaber, M. M., & Elyan, E. (2014). Random forests: From early developments to recent advancements. *Systems Science & Control Engineering*, *2*(1), 602–609. https://doi.org/10.1080/21642583.2014.956265
52. Genuer, R., & Poggi, J.-M. (2020). Random Forests. In R. Genuer & J.-M. Poggi, *Random Forests with R* (pp. 33–55). Springer International Publishing. https://doi.org/10.1007/978-3-030-56485-8_3
53. Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1
54. Ahmad, M. W., Reynolds, J., & Rezgui, Y. (2018). Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. *Journal of Cleaner Production*, *203*, 810–821. https://doi.org/10.1016/j.jclepro.2018.08.207
55. John, V., Liu, Z., Guo, C., Mita, S., & Kidono, K. (2016). Real-Time Lane Estimation Using Deep Features and Extra Trees Regression. In T. Bräunl, B. McCane, M. Rivera, & X. Yu (Eds.), *Image and Video Technology* (Vol. 9431, pp. 721–733). Springer. https://doi.org/10.1007/978-3-319-29451-3_57
56. Cao, L., Ji, X., Cao, Y., Li, Y., Siang, L. C., Li, J., Pediredla, V. K., & Gopaluni, R. B. (2023). Interpretable Soft Sensors using Extremely Randomized Trees and SHAP. *IFAC-PapersOnLine*, *56*(2), 8000–8005. https://doi.org/10.1016/j.ifacol.2023.10.922

57. Geurts, P., & Louppe, G. (2011). Learning to rank with extremely randomized trees. *Proceedings of the Learning to Rank Challenge*, 49–61.
58. Louppe, G., & Geurts, P. (2012). Ensembles on Random Patches. In P. A. Flach, T. De Bie, & N. Cristianini (Eds.), *Machine Learning and Knowledge Discovery in Databases* (Vol. 7523, pp. 346–361). Springer. https://doi.org/10.1007/978-3-642-33460-3_28
59. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785
60. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *39*(5), 1189–1232.
61. Huber, F., Yushchenko, A., Stratmann, B., & Steinhage, V. (2022). Extreme Gradient Boosting for yield estimation compared with Deep Learning approaches. *Computers and Electronics in Agriculture*, *202*, 107346. https://doi.org/10.1016/j.compag.2022.107346
62. Nielsen, D. (2016). *Tree boosting with xgboost-why does xgboost win" every" machine learning competition?* [Master's Thesis, NTNU].
63. Peng, J., Sun, W., Zhou, G., Xie, L., Han, H., & Xiao, Y. (2022). The accurate prediction and analysis of bed expansion characteristics in liquid–solid fluidized bed based on machine learning methods. *Chemical Engineering Science*, *260*, 117841. https://doi.org/10.1016/j.ces.2022.117841
64. Sakheta, A., Raj, T., Nayak, R., O'Hara, I., & Ramirez, J. A. (2025). Development and assessment of hybrid machine learning model of biomass pyrolysis process. *Chemical Engineering Science*, *310*, 121552. https://doi.org/10.1016/j.ces.2025.121552
65. Pereira, J. L. J., Oliver, G. A., Francisco, M. B., Cunha, S. S., & Gomes, G. F. (2021). A review of multi-objective optimization: Methods and algorithms in mechanical engineering problems. *Archives of Computational Methods in Engineering*, *29*(4), 2285–2308. https://doi.org/10.1007/s11831-021-09663-x
66. Chiandussi, G., Codegone, M., Ferrero, S., & Varesio, F. E. (2012). Comparison of multi-objective optimization methodologies for engineering applications. *Computers & Mathematics with Applications*, *63*(5), 912–942. https://doi.org/10.1016/j.camwa.2011.11.057

67. Baril, C., Yacout, S., & Clément, B. (2011). Design for Six Sigma through collaborative multiobjective optimization. *Computers & Industrial Engineering*, *60*(1), 43–55. https://doi.org/10.1016/j.cie.2010.09.015
68. Ojha, M., Singh, K. P., Chakraborty, P., & Verma, S. (2019). A review of multi-objective optimisation and decision making using evolutionary algorithms. *International Journal of Bio-Inspired Computation*, *14*(2), 69–84. https://doi.org/10.1504/IJBIC.2019.101640
69. Deb, K. (2011). *Multi-objective optimisation using evolutionary algorithms: An introduction*. Springer.
70. Miettinen, K. (1999). *Nonlinear multiobjective optimization* (Vol. 12). Springer.
71. Afshari, H., Hare, W., & Tesfamariam, S. (2019). Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures. *Applied Soft Computing*, *83*, 105631. https://doi.org/10.1016/j.asoc.2019.105631
72. Deb, K. (2008). Introduction to evolutionary multiobjective optimization. *Multiobjective Optimization: Interactive and Evolutionary Approaches*, 59–96. https://doi.org/10.1007/978-3-540-88908-3_3
73. Gomes, G. F., de Almeida, F. A., da Silva Lopes Alexandrino, P., da Cunha, S. S., de Sousa, B. S., & Ancelotti, A. C. (2019). A multiobjective sensor placement optimization for SHM systems considering Fisher information matrix and mode shape interpolation. *Engineering with Computers*, *35*, 519–535. https://doi.org/10.1007/s00366-018-0613-7
74. Zeleny, M. (1973). Multiple criteria decision making. *Compromise Programming*, 262–301.
75. Odu, G. O., & Charles-Owaba, O. E. (2013). Review of multi-criteria optimization methods–theory and applications. *IOSR Journal of Engineering*, *3*(10), 01–14. https://doi.org/10.9790/3021-031020114
76. Triantaphyllou, E., Shu, B., Sanchez, S. N., & Ray, T. (1998). Multi-criteria decision making: An operations research approach. *Encyclopedia of Electrical and Electronics Engineering*, *15*(1998), 175–186.
77. Gunantara, N. (2018). A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, *5*(1), 1502242. https://doi.org/10.1080/23311916.2018.1502242
78. Chankong, V., & Haimes, Y. Y. (2008). *Multiobjective decision making: Theory and methodology*. Courier Dover Publications.

79. Haimes, Y. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, (3), 296–297. https://doi.org/10.1109/tsmc.1971.4308298
80. Fishburn, P. C. (1974). Exceptional paper—Lexicographic orders, utilities and decision rules: A survey. *Management Science*, *20*(11), 1442–1471. https://doi.org/10.1287/mnsc.20.11.1442
81. Chang, C.-T. (2007). Multi-choice goal programming. *Omega*, *35*(4), 389–396. https://doi.org/10.1016/j.omega.2005.07.009
82. Bui, L. T., & Alam, S. (2008). *Multi-objective optimization in computational intelligence: Theory and practice*. IGI Global.
83. Cáceres Sepúlveda, G. (2019). *Relevance of multi-objective optimization in the chemical engineering field* [Master thesis, University of Ottawa]. https://doi.org/10.20381/ruor-24026
84. Brans, J.-P., Vincke, P., & Mareschal, B. (1986). How to select and how to rank projects: The PROMETHEE method. *European Journal of Operational Research*, *24*(2), 228–238. https://doi.org/10.1016/0377-2217(86)90044-5
85. Mareschal, B., Brans, J. P., & Vincke, P. (1984). *PROMETHEE: A new family of outranking methods in multicriteria analysis*. 2013/9305, ULB–Université Libre de Bruxelles. https://ideas.repec.org/s/ulb/ulbeco.html

Chapter 3

Harnessing Machine Learning and Optimization for Informed Chemical Engineering Decisions: A Styrene Reactor Analysis

Farough Agin, Clémence Fauteux-Lefebvre, and Jules Thibault*

Department of Chemical and Biological Engineering

University of Ottawa, Ottawa, Ontario, K1N 6N5, Canada

*Corresponding Author: Jules Thibault

This chapter is a manuscript published in The Canadian Journal of Chemical Engineering (2026).

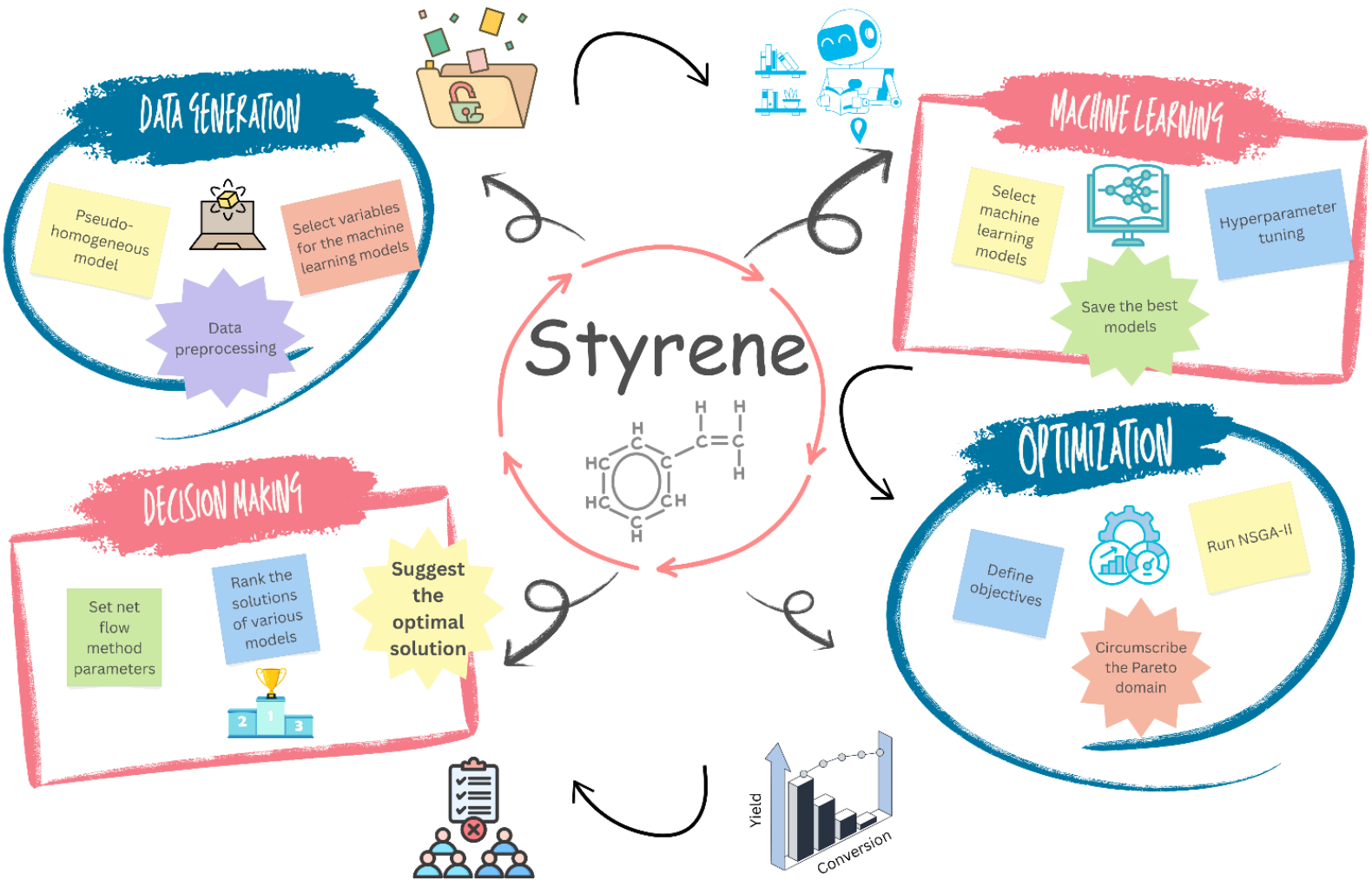
3.1 Abstract

Machine learning (ML) has become a powerful asset in chemical engineering, offering accurate, data-driven approaches for modelling and process optimization. While ML enhances the prediction of complex system behaviours beyond traditional empirical models, its integration with optimization frameworks is essential to identify operating conditions that align with multiple industrial goals. This study presents an integrated approach that combines various supervised ML models with multi-objective optimization (MOO) and multi-criteria decision-making (MCDM) techniques, applied to a simulated styrene production reactor. The goal is to train various ML models and use them to simultaneously optimize ethylbenzene conversion and styrene yield, using datasets of varying size and quality. Model performance is evaluated under three data conditions: a complete dataset, a reduced clean dataset, and a noisy dataset. While individual ML models generate different Pareto fronts and optimal solutions, this work demonstrates that a novel consensus-based strategy that leverages recurring patterns across the full ensemble of ML models can better predict outcomes and produce more robust and reliable recommendations for process conditions than relying on any single model. By aggregating agreement predictions across diverse learners, the consensus approach reduces model-specific bias and sensitivity to data imperfections, yielding recommendations that are more stable across data quality scenarios and more defensible for decision-making.

By highlighting both the predictive power and optimization potential of ML models across data environments, this research provides a practical framework for integrating ML with MOO/MCDM in chemical process design. The findings emphasize the importance of robustness and interpretability in model-driven decision-making and contribute to the advancement of sustainable and adaptive strategies in chemical engineering practice.

Keywords: Machine learning, multi-criteria optimization, multi-criteria decision-making, styrene production

3.2 Graphical abstract



3.3 Introduction

The rapid advancement of technology continues to transform the modern world, with one of the most notable developments in the past decade being the widespread integration of artificial intelligence (AI) into daily life. Many real-world challenges are characterized by uncertainty and complex probabilistic data, which are often difficult for humans to interpret and process effectively. In this context, machine learning (ML) serves as an essential tool for converting raw data into actionable insights, facilitating a broad spectrum of applications [1,2]. AI and ML have emerged as highly effective and economical approaches, often surpassing the performance of traditional methods [3]. In the field of engineering, AI has found broad applications, including engineering design [4], manufacturing [5], geotechnical engineering [6], civil engineering [7], materials engineering [8], and many other engineering fields. In the field of chemical engineering, ML has demonstrated significant potential across a range of applications [9,10], including fault detection [11], process monitoring [12], process control [13], catalyst discovery [14,15], soft-sensing [16], operational optimization [17], and batch process monitoring [18]. Trained ML models can serve as surrogate objective functions for optimization tasks [19], particularly in chemical engineering, where problems are often highly nonlinear, high-dimensional, and computationally expensive to solve using traditional mechanistic models [20,21].

Optimization is essential in chemical processes due to evolving economic pressures, environmental regulations, and rising energy costs [22]. It supports key stages such as process modelling, design, operation, and retrofitting across industries like chemical, petrochemical, pharmaceutical, and energy [23,24]. Engineers use optimization to enhance performance, reduce costs, and ensure safety and reliability. However, chemical systems are often complex, nonlinear, and constrained, making optimization challenging. Additionally, objectives often conflict, requiring multi-objective optimization (MOO) to generate Pareto-optimal solutions that offer balanced trade-offs for informed decision-making [25,26].

While MOO provides a set of Pareto-optimal solutions that represent trade-offs among conflicting objectives, practical implementation ultimately requires the selection of a single, most appropriate solution. This critical step—identifying one final option from a range of non-dominated alternatives—is where multi-criteria decision-making (MCDM) becomes essential [27]. MCDM provides a systematic framework for evaluating and ranking solutions based on multiple,

often conflicting, criteria. By quantifying the performance of each solution against a set of defined objectives, MCDM facilitates informed decision-making in complex scenarios [28].

In the context of chemical and process engineering, although MOO has gained considerable attention, the integration of MCDM for selecting a particular solution among Pareto-optimal solutions remains relatively underexplored [29]. Nevertheless, MCDM plays a crucial role in translating optimization outcomes into actionable decisions, particularly in high-stakes applications where economic, environmental, operational, and safety considerations must be balanced [30]. These mathematical approaches typically involve several analytical steps to rank solutions, from which the highest-ranked option can be selected for implementation. In this way, MCDM serves as a valuable complement to MOO, bridging the gap between optimization and practical decision-making.

To assess the research trends in the combined application of ML, MOO, and MCDM techniques, a comprehensive bibliometric analysis was conducted across all fields, with a specific focus on chemical engineering. Relevant publications were retrieved from Scopus and Web of Science using predefined search keywords (listed in Appendix A, Table A.1). Both databases were searched to improve coverage and reduce the risk of missing relevant studies, as Scopus and Web of Science differ in journal inclusion and indexing breadth. The results from both databases were merged and processed using the Covidence tool (Covidence) to eliminate duplicates and ensure a systematic review. The initial search retrieved 460 records from Scopus and 212 records from Web of Science, giving 672 records before deduplication. After import into Covidence, 96 duplicate records were removed, leaving 576 unique records for manual title-and-abstract screening. During this manual screening step, 532 records were excluded because they did not sufficiently match the combined methodological focus on ML, MOO, and, where applicable, MCDM. The remaining 44 representative studies were retained for detailed classification in Table A.2.

Figure 3.1 presents the annual number of publications involving ML, MOO, and MCDM together across all fields (Figure 3.1a) and within chemical engineering (Figure 3.1b). While the overall volume of publications is considerably higher in the broader scientific literature, a noticeable rise in chemical engineering publications in recent years confirms growing interest in applying this integrated approach within the discipline. Despite this growth, the combined application of ML, MOO, and MCDM remains relatively new in chemical engineering and

warrants further attention and exploration. Additional bibliometric visualizations, including the keyword co-occurrence network, annual keyword trends, and word cloud are provided in the support information file (Figure A.1 to Figure A.3). The keyword co-occurrence revealed interdisciplinary clusters linking ML, optimization, and engineering applications. The word cloud highlights core themes like decision-making, modelling, and data-driven optimization.

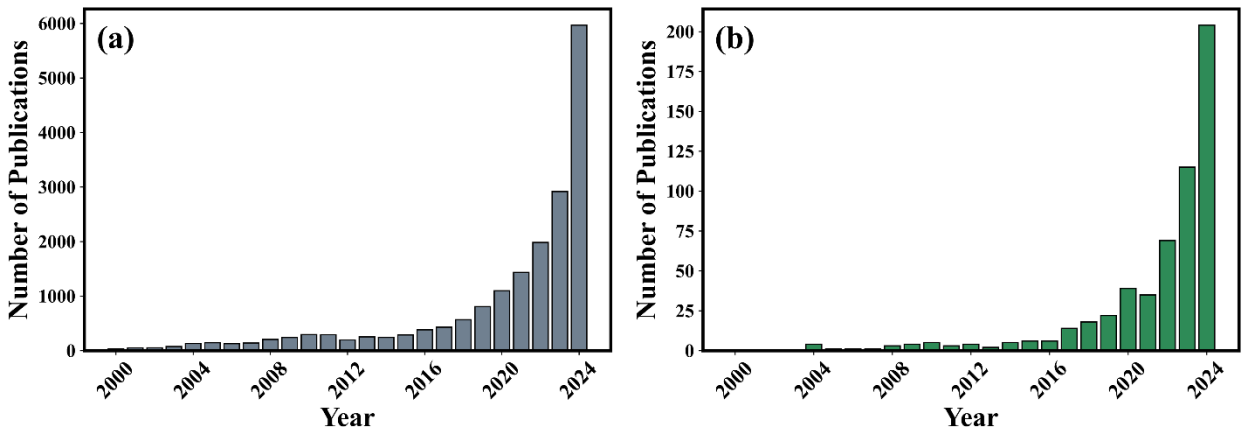


Figure 3.1. Annual number of publications involving the combined use of ML, MOO, and MCDM: (a) across all fields and (b) specifically within chemical engineering, based on Scopus and Web of Science data.

Drawing from the findings from the bibliometric analyses, the relevant and representative papers that integrate ML, MOO, and MCDM within chemical engineering were selected for in-depth review. For each paper, key information was extracted and organized into a structured table, including the case study, data origin, input and output variables, ML models used, hyperparameter tuning method, final selected ML model, optimization algorithm, and MCDM technique, if applied. The papers were further categorized by their year of publication, providing a chronological perspective on the development of this integrated approach. To better capture the breadth of applications, each paper was grouped into one of the following application domains: (1) energy systems, (2) heat transfer and thermal management, (3) process simulation and optimization, and (4) materials and construction. This structured categorization enables a cross-disciplinary comparative analysis of emerging trends and methodological approaches, emphasizing the evolving integration of ML, MOO, and MCDM in addressing domain-specific challenges. A detailed summary of this classification is presented in Table A.2.

This analysis highlights that while the integration of these methodologies is gaining traction, there is still a lack of comprehensive, structured studies demonstrating how to systematically apply this combined approach within chemical processes. To bridge this gap, the present study introduces a comprehensive, step-by-step methodology encompassing data collection, preprocessing, variable selection, ML model development, MOO formulation, and MCDM-based decision-making. A key novelty of this work is the introduction of a consensus-based modelling layer that synthesizes information across all trained ML models to yield more reliable outcome predictions and optimization recommendations. Rather than relying on a single "best" model (which may be sensitive to data quality, noise, or model-specific bias), the proposed consensus approach exploits recurring patterns and agreement signals across diverse learners to identify stable, defensible process-condition recommendations, improving robustness and reducing the risk of overfitting-driven decisions, especially when Pareto fronts differ across models.

The styrene production process is selected as the case study, with the objective of optimizing continuous performance indicators through supervised regression models. Nine widely adopted ML algorithms are implemented and compared—namely artificial neural networks (ANN), bagging regressors (Bagging), decision trees (DT), extra trees (ET), eXtreme gradient boosting (XGBoost), Gaussian process regression (GPR), k-nearest neighbours (KNN), random forest (RF), and support vector regression (SVR). Comprehensive details are provided on the model training procedures, hyperparameter tuning strategies, and performance evaluation. Following the identification of the best-performing models, their outputs are used to define objective functions within a MOO framework, resulting in a Pareto front that captures optimal trade-offs. In parallel, the consensus strategy is used to consolidate model outputs into a robust set of recommendations that remains consistent across modelling assumptions and data environments, strengthening the reliability of the final operating-condition selection. These solutions are then ranked using an MCDM technique. While an exhaustive comparison of MOO and MCDM algorithms falls outside the scope of this work, NSGA-II [31] was selected as a widely used optimization method, and the net flow method (NFM) [32] was adopted as a suitable outranking-based MCDM technique for ranking Pareto-optimal solutions in the present study. This study serves both as a practical demonstration of chemical process optimization using advanced data-driven tools and as a user guide for researchers seeking to apply ML, MOO, and MCDM in an integrated manner.

3.4 Methodology

The proposed methodology (Figure 3.2) presents an integrated workflow combining simulation-based data generation, ML model development, MOO, and MCDM. It begins with problem definition and data generation using a pseudo-homogeneous adiabatic reactor model of the styrene production process. Key input variables include ethylbenzene flow rate, steam flow rate, inlet temperature, and inlet pressure, while ethylbenzene conversion and styrene yield are selected as outputs. Nine supervised regression models (ANN, Bagging, DT, ET, XGBoost, GPR, KNN, RF, and SVR) are trained, normalized, and tuned to evaluate their predictive performance and identify the most accurate models. The best-performing models are then used to formulate a two-objective optimization problem that maximizes both conversion and yield, solved using NSGA-II to generate a Pareto optimal front. These solutions are further evaluated using MCDM through the NFM technique to rank the alternatives and identify the most preferred operating point. Overall, this framework demonstrates a cohesive integration of ML, MOO, and decision-making methods for analyzing and optimizing the styrene production reactor.

3.4.1 Styrene production process and dataset generation

Styrene ($C_6H_5CHCH_2$) is a key aromatic monomer extensively used in the production of polymer-based materials, including polystyrene, acrylonitrile-butadiene-styrene, styrene-acrylonitrile, styrene-butadiene rubber, latexes, and unsaturated polyester resins [33–36]. Its industrial significance is reflected in its broad range of applications across the packaging, construction, electronics, and automotive sectors. The global styrene market is projected to grow significantly, rising from USD 63.7 billion in 2025 to over USD 108 billion by 2034 [37].

Over 90% of styrene is produced via the catalytic dehydrogenation of ethylbenzene ($C_6H_5CH_2CH_3$), carried out in the presence of superheated steam over an iron oxide-based catalyst. This reaction is both endothermic and reversible, with thermodynamic conditions favouring styrene formation at high temperatures and low pressures. Superheated steam plays a critical role in this reaction by supplying the required thermal energy for the endothermic reaction, inhibiting coke formation, and lowering the partial pressures of the reaction products, thereby enhancing the equilibrium conversion toward styrene [35,38]. The primary reaction involves the conversion of ethylbenzene into styrene and hydrogen (Equation (3.1)). However, this process is accompanied

by several side reactions (Equations (3.2)-(3.6)) that negatively impact product selectivity and overall efficiency [33].

$$C_6H_5CH_2CH_3 \rightleftharpoons C_6H_5CHCH_2 + H_2 \quad (3.1)$$
$$C_6H_5CH_2CH_3 \rightarrow C_6H_6 + C_2H_4 \quad (3.2)$$
$$C_6H_5CH_2CH_3 + H_2 \rightarrow C_6H_5CH_3 + CH_4 \quad (3.3)$$
$$2H_2O + C_2H_4 \rightarrow 2CO + 4H_2 \quad (3.4)$$
$$H_2O + CH_4 \rightarrow CO + 3H_2 \quad (3.5)$$
$$H_2O + CO \rightarrow CO_2 + H_2 \quad (3.6)$$

These reactions occur over a promoted iron oxide catalyst, such as Shell 105®, which comprises 62% Fe_2O_3, 36% K_2CO_3, and 2% Cr_2O_3 [38]. Under optimal operating conditions, the conversion of ethylbenzene can reach up to 80%. However, this high conversion must be balanced against catalyst degradation due to thermal cracking and the formation of by-products, including benzene (C_6H_6), toluene ($C_6H_5CH_3$), methane (CH_4), and carbon oxides (CO, CO_2). The process typically operates at temperatures exceeding 857 K, with an inlet steam-to-ethylbenzene molar ratio ranging from 7 to 20 [33,39].

In this study, the styrene production process is used as a case study to demonstrate the integration of ML, MOO, and MCDM methodologies in the context of chemical process engineering. A comprehensive dataset was generated by simulating an adiabatic plug flow reactor using the pseudo-homogeneous kinetic model developed by Sheel and Crowe [38]. This model, based on industrial reactor data, assumes ideal plug flow behaviour and incorporates experimentally derived rate expressions that account for heat and mass transfer limitations within catalyst pellets. The steady state simulation used the finite difference method, implemented in Fortran, to solve material and energy balances along a fixed reactor length and diameter while randomly varying four industrial operating parameters: reactor inlet temperature, inlet pressure, ethylbenzene flow rate, and steam flow rate [33]. Each run generated detailed reactor outputs, including outlet flow rates of ten chemical species, as well as outlet temperature and pressure, from which ethylbenzene conversion and styrene yield were calculated. Conversion was determined from the change in ethylbenzene flow rate (Equation (3.7)), and yield from the ratio of styrene produced to ethylbenzene converted (Equation (3.8)). This dataset provided the basis for developing ML models capable of capturing the reactor's nonlinear behaviour, with conversion and yield selected as the two-performance metrics for the optimization framework.

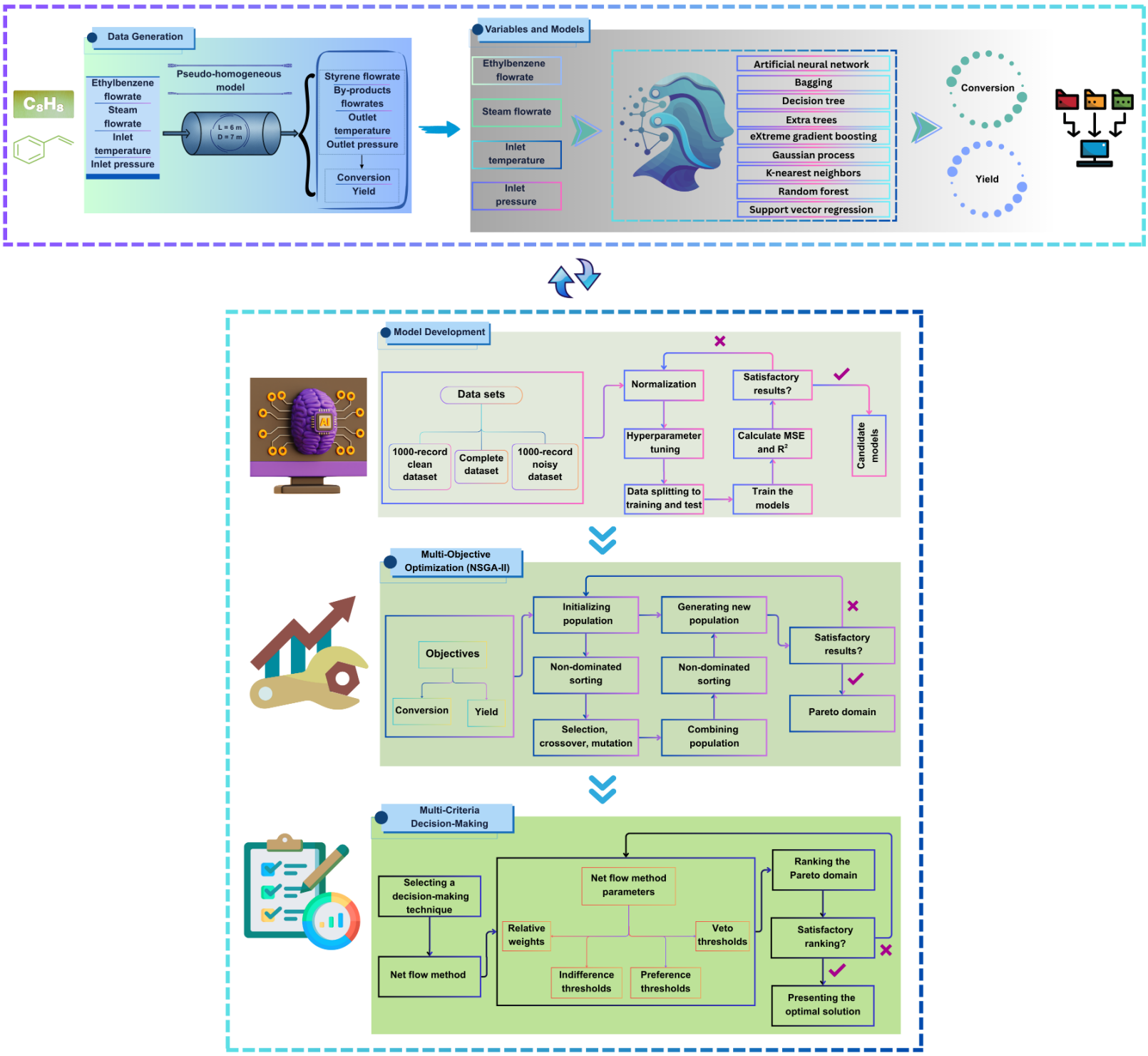


Figure 3.2. Workflow of the proposed framework integrating ML, NSGA-II, and NFM for optimizing styrene production based on simulation data.

$$X_{EB} = \frac{F_{EB,in} - F_{EB,out}}{F_{EB,in}} \tag{3.7}$$

$$Y_{ST} = \frac{F_{ST}}{F_{EB,in} - F_{EB,out}} \tag{3.8}$$

where $F_{EB,in}$ and $F_{EB,out}$ are the inlet and outlet ethylbenzene flow rates, respectively, and $F_{ST,out}$ is the outlet styrene flow rate, all expressed in mol/s. X_{EB} and Y_{ST} denote ethylbenzene conversion and styrene yield, respectively, and are dimensionless fractions.

A set of key operating parameters was selected as inputs for ML modelling, including ethylbenzene flow rate, steam flow rate, reactor inlet temperature, and reactor inlet pressure, because these variables directly govern the thermal and reaction environment within the reactor and strongly influence conversion, selectivity, and by-product formation. Ethylbenzene conversion and styrene yield were selected as the outputs because they are key reactor-level performance indicators that reflect both reactant utilization and formation of the desired product. Although economic indicators such as styrene production rate, operating cost, or profit are highly relevant for full process optimization, the present study was intentionally limited to reactor-level technical objectives in order to establish and compare the integrated ML–MOO–MCDM workflow using core performance indicators. Extending the framework to include economic objectives would be a valuable direction for future work, particularly in a broader process-level setting.

The simulation generated about 35,000 steady-state samples covering a wide range of operating conditions. The simulation dataset was generated in Fortran by pseudo-random sampling within predefined operating bounds rather than through a formal design of experiments technique. Specifically, the steam-to-ethylbenzene molar ratio was sampled between 8 and 12, the inlet temperature between 750 and 950 K, the inlet pressure between 150 and 300 kPa, and the ethylbenzene inlet flowrate between 50 and 80 mol/s; the corresponding steam flowrate was then calculated from the sampled steam-to-ethylbenzene ratio and ethylbenzene flowrate. This was done to obtain broad coverage of the feasible operating domain for subsequent ML model development. From this, three datasets were prepared: the *complete dataset*, a *1000-record clean dataset* selected from the full set using random sampling with stratified retention of boundary values, and a *1000-record noisy dataset* created by adding Gaussian noise with mean zero and standard deviation of 0.03 to the output variables to represent sensor variability. The 1000-record subset was chosen intentionally to create a substantially reduced-data scenario while still

preserving coverage of the full operating domain. A stratified sampling procedure ensured that both smaller datasets retained boundary values across all input variables, preserving the full operating range. The noisy dataset was generated from the same 1000-record subset to support a consistent comparison, whereas larger reduced datasets would have remained closer to the complete dataset and provided a less stringent test of model robustness to limited and noisy data. The Gaussian noise was added to the raw output variables before normalization using an additive noise model (Equation (3.9)):

$$y_i^{\text{noisy}} = y_i + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, 0.03^2) \tag{3.9}$$

where y_i denotes either ethylbenzene conversion or styrene yield expressed in fractional form. Any noise-affected output values outside the physically valid range of 0 to 1 were clipped before normalization and subsequent model preprocessing. The same absolute noise level was applied to both outputs in their original scale. The value of 0.03 was selected to represent a moderate level of output uncertainty consistent with plausible sensor or measurement variability, while still preserving meaningful structure in the data for model training and comparison. No additional output clipping or bound-enforcing treatment was applied during model training or optimization. This choice was made intentionally to assess whether the trained surrogate models and the optimization framework could maintain physically reasonable predictions without explicit output constraints.

Table 3.1 summarizes the selected input and output variables along with their respective minimum and maximum values across the complete dataset. These ranges define the domain of the simulated operating conditions and model outputs. Additionally, Figure A.4 to Figure A.6 provide visual insights into the distribution of variables across all three datasets, and Figure A.7 presents the characteristics of the noise applied to the outputs.

3.4.2 Preprocessing and model development

Developing a reliable and robust ML model begins with an essential phase of data preprocessing, vital in managing complex datasets and ensuring the quality and consistency of inputs used for training. Standard preprocessing typically involves five key tasks: data integration, data cleaning, normalization, handling missing values, and noise identification [40,41]. In this study, four input variables and two output variables, derived from the simulation of the styrene production reactor, were predefined as model features. Given that the dataset was generated synthetically, it was

inherently complete and free of missing values or outliers. Moreover, due to the relatively low dimensionality of the input space, dimensionality reduction techniques were deemed unnecessary. Consequently, normalization was the sole preprocessing step applied. All features were scaled to the range [0, 1], ensuring uniformity across input variables and preventing any single feature from disproportionately influencing the learning process.

Table 3.1. Summary of selected input and output variables used in the ML models, including their minimum and maximum values as observed in the complete dataset.

Variables	Mean	Std Dev	Min	Max
Inlet flow of ethylbenzene (mol/s)	64.75	8.63	50	80
Inlet flow of steam (mol/s)	645.70	113.35	401	957
Inlet temperature (K)	848.99	57.67	750	950
Inlet pressure (kPa)	225.86	42.69	150	300
Conversion (fraction)	0.38	0.14	0.12	0.68
Yield (fraction)	0.81	0.15	0.16	0.99

For each dataset, the samples were randomly partitioned to evaluate model generalization. For an ANN, an additional validation set is required during training to monitor learning progress and prevent overfitting. To maintain consistency across all models and ensure fair comparisons, a total of 20% of the data was reserved for testing and validation combined. Specifically, 80% of the data was allocated for training across all models. For models other than the ANN, the remaining 20% served exclusively as the test set. For the ANN model, this 20% was evenly divided: 10% for validation and 10% for testing. This strategy ensured that all models were trained on an identical number of samples, enabling a fair and unbiased comparison of model performance.

Since the modelling task involves supervised regression, performance metrics were selected to evaluate and compare predictive accuracy. In regression problems, model performance is commonly assessed by comparing predicted values against actual observations in the test set. A variety of error-based metrics are available for this purpose, including the root mean squared error (RMSE), the mean absolute error (MAE), the mean squared error (MSE), the Pearson correlation coefficient, and the coefficient of determination (R^2), each capturing different aspects of prediction quality [42]. In this work, MSE and R^2 were chosen as the primary evaluation metrics. Together, they provide a complementary perspective on both the magnitude of prediction errors and the proportion of variance explained by the models. Because the input and output variables were

normalized to the range [0,1] prior to model development, the reported MSE values were calculated on the normalized output scale.

All model development was implemented in Python. The ANN models were developed in TensorFlow, while SVR, GPR, KNN, DT, Bagging, RF, and ET were implemented using scikit-learn, and XGBoost was implemented using the XGBoost Python package.

3.4.3 Machine learning models

3.4.3.1 Artificial neural networks

Artificial neural networks (ANN) are computational models made of interconnected neurons that process information through weighted sums and nonlinear activation functions, allowing them to approximate complex relationships [43]. A typical ANN includes input, hidden, and output layers, with feedforward networks being the most widely used due to their simple structure and strong approximation capability [44]. ANN development involves designing the network architecture and training it to minimize prediction error, usually through backpropagation or more advanced optimizers such as Levenberg-Marquardt [45] and Adam [46].

3.4.3.2 Support vector regression

Support vector regression (SVR) extents the principles of support vector machines to predict continuous outcomes while maintaining strong generalization in high-dimensional feature spaces. It estimates a function constrained within an epsilon-wide margin around the true values, known as the epsilon-insensitive tube, where small errors are ignored [47,48]. Data points that fall outside this tube become support vectors, resulting in a sparse model trained through convex optimization that balances flatness of the function with prediction accuracy [49]. Through kernel functions such as radial basis and polynomial kernels, SVR can capture complex nonlinear relationships by implicitly mapping inputs into richer feature spaces [50]. Collectively, these characteristics make SVR a practical and robust tool for modelling nonlinear patterns in small or moderately sized datasets.

3.4.3.3 Gaussian process regression

Gaussian process regression (GPR) is a nonparametric Bayesian method that models a distribution over possible functions rather than fitting a single fixed equation. This framework enables both predictive outputs and principled uncertainty estimates [51,52]. In a Gaussian process, function values at different inputs are assumed to be jointly Gaussian distributed, with smoothness and

similarity encoded through kernels such as the radial basis function [53]. The model is defined by a mean function and a covariance function, whose hyperparameters are typically optimized by maximizing the marginal likelihood. As new observations are incorporated, the prior distribution is updated to a posterior that yields both a predicted mean and a variance reflecting confidence in the prediction [54]. This makes GPR especially useful for small or noisy datasets and for applications where uncertainty quantification is important. Its main limitation is the high computational cost, which scales cubically with the number of data points [55].

3.4.3.4 K-nearest neighbours

K-nearest neighbours (KNN) is a nonparametric, instance-based regression method that predicts the output of a new data point by averaging the values of its k most similar training samples, typically determined using Euclidean distance [56]. Because it builds no explicit model and instead relies directly on the structure of the data, KNN can capture complex, nonlinear relationships without assuming a predefined functional form. The choice of k is critical: small values risk overfitting, while larger values yield smoother but less detailed predictions. Accuracy can be further improved through distance-weighted variants, which assign greater influence to closer neighbours [57]. This simplicity and flexibility make KNN well-suited for regression problems where the data distribution is unknown or highly variable [58].

3.4.3.5 Decision tree

Decision tree (DT) regression is a nonparametric method for predicting continuous outcomes by recursively partitioning the data into subsets that minimize prediction error [59]. At each step, the algorithm selects the feature and threshold that most effectively reduce the sum of squared deviations, continuing until a stopping criterion is reached. The terminal leaf nodes provide constant predictions based on the average target value within each region, allowing DTs to capture complex, nonlinear relationships without assuming a specific data distribution [60–62]. Their hierarchical structure makes DTs highly interpretable, resembling human decision-making, while also performing implicit feature selection by prioritizing the most informative variables [63]. Although prone to overfitting when grown too deep, pruning techniques can enhance generalization [64]. Overall, DTs offer a simple, interpretable, and flexible approach that requires minimal preprocessing.

3.4.3.6 Bagging

Bagging regressor (Bagging) is an ensemble learning method designed to improve prediction stability by training multiple base models on different bootstrapped samples drawn with replacement from the original dataset [65]. Each base learner, typically an unpruned decision tree, is trained independently, and their predictions are averaged to reduce variance and improve generalization [66]. By combining several weakly correlated models, Bagging effectively lowers variance without increasing bias, making it particularly well-suited for high-variance learners such as decision trees [67]. The method also supports out-of-bag error estimation, which leverages data excluded from each bootstrap sample to evaluate performance without requiring a separate validation set [68]. Overall, Bagging provides a simple yet powerful approach for improving accuracy and robustness in regression tasks.

3.4.3.7 Random forest

Random forest (RF) regression extends Bagging by training multiple decision trees on bootstrapped samples while also selecting a random subset of features at each split. This strategy reduces correlation among trees and improves generalization [69,70]. Each tree is grown to full depth to capture complex patterns, and their averaged predictions lower variance and reduce overfitting, resulting in a model with strong accuracy and stability [68,71]. RF performs well on high-dimensional and noisy datasets and provides useful feature importance measures that enhance interpretability, even though it is less transparent than a single decision tree. With only a few key hyperparameters, RF remains a widely used and reliable regression method when both robustness and predictive performance are required [72].

3.4.3.8 Extremely randomized trees

Extremely randomized trees (ET) is an ensemble regression method that builds multiple decision trees with stronger randomness than RF, leading to more diverse models and lower variance [73,74]. Unlike RFs, which train each tree on bootstrapped samples, ET uses the entire dataset and randomly selects both the feature and the split threshold before choosing the best among them. This approach yields models that are fast to train, easy to tune, and well-suited for high-dimensional or noisy data, while still capturing nonlinear relationships and providing feature importance measures [75–77].

3.4.3.9 eXtreme gradient boosting

The eXtreme gradient boosting (XGBoost) regressor is a high-performance ensemble method that builds trees sequentially, with each new tree correcting the residual errors of its predecessors. This iterative process enables strong accuracy on complex regression tasks (Chen and Guestrin, 2016). Tree construction is optimized using first- and second-order derivatives of the loss function, allowing precise and efficient updates that improve the model at each stage. XGBoost also incorporates regularization to control tree complexity and mitigate overfitting, making it robust when handling high-dimensional, noisy, or sparse datasets [79]. Key hyperparameters, such as maximum tree depth, learning rate, and subsampling ratios, provide flexibility to balance bias, variance, and computational cost [80,81]. Collectively, these features have made XGBoost one of the most widely adopted and highly accurate regression techniques.

3.4.4 Hyperparameter tuning

Hyperparameters are external settings that shape the behaviour, complexity, and performance of ML models. Unlike internal parameters, which are learned directly from data, hyperparameters must be specified before training [82]. Careful selection is critical, and this selection process, known as hyperparameter optimization, aims to identify the combination that yields the best validation performance. Hyperparameter optimization is typically treated as a black box optimization problem [44,82,83]. Numerous hyperparameter optimization methods exist, including grid search, random search, evolutionary algorithms, Bayesian optimization, Hyperband, and racing strategies. Random search is often preferred for its simplicity and efficiency, particularly when paired with cross-validation [84]. In this study, random search with 5-fold cross validation was used to tune nine ML models across three datasets, leading to 27 total tuned models. Each model was evaluated with 50 hyperparameter configurations to ensure fair and optimized performance. The resulting search spaces and selected optimal hyperparameters are reported in Table A.3 to Table A.11 for transparency and reproducibility.

3.4.5 Optimization

ML is often integrated with optimization in engineering applications to fine-tune process parameters, especially for MOO, where several competing goals must be balanced simultaneously [24,33,85]. These problems rely on evaluating decision variables through multiple objective functions within a constrained search space, and conflicts between objectives are handled through Pareto

optimality, which returns a set of non-dominated trade-off solutions rather than a single aggregated answer [86]. Evolutionary algorithms are well-suited for exploring such complex landscapes, with the NSGA-II being one of the most influential methods because it uses non-dominance ranking, elitism, and crowding distance to produce a diverse and high-quality Pareto front [87]. In NSGA-II, parent and offspring populations are merged, sorted into dominance-based fronts, and the next generation is formed by selecting the best individuals while preserving diversity.

Although deterministic approaches such as the ε-constraint method can be effective for two-objective problems, their suitability depends strongly on the smoothness of the surrogate model. In the present framework, smoother models such as ANN, GPR, and SVR are more compatible with deterministic optimization, whereas tree-based and instance-based models such as DT, RF, ET, Bagging, and KNN have non-smooth or piecewise prediction structures that make them less straightforward for classical deterministic solvers.

The NSGA-II optimization was implemented in Python using the Pymoo package. For each case, the trained ML model was embedded directly as a surrogate evaluator within the optimization loop, so that candidate operating conditions generated by NSGA-II were automatically passed to the corresponding model to predict ethylbenzene conversion and styrene yield.

3.4.6 Multi-criteria decision-making

The NFM is used to rank the Pareto optimal solutions by comparing them pairwise through a structured MCDM process that blends the concordance–discordance logic of ELECTRE with the net flow scoring approach of PROMETHEE [88,89]. NFM incorporates decision-maker preferences through four parameters for each objective: relative weights, an indifference threshold that treats small differences as equivalent, a preference threshold that indicates when one solution is favoured, and a veto threshold that can exclude a solution if it performs unacceptably on any objective. Using these thresholds, NFM calculates concordance indices that measure agreement in favour of a solution and discordance indices that capture strong disagreements. These are combined to build an outranking matrix that quantifies how much each solution dominates others. The final ranking is obtained by computing the net flow score, which represents the difference between how strongly a solution outranks others and how strongly it is outranked by others.

3.5 Results and discussion

3.5.1 Model development

As outlined in Section 2.3, nine supervised regression models using the described ML methods were developed for each of the three datasets: the complete dataset, the 1000-record clean dataset, and the 1000-record noisy dataset. These models aimed to predict two key outputs of the styrene production process: ethylbenzene conversion and styrene yield. All models were trained following appropriate data normalization and extensive hyperparameter tuning, as previously detailed. Model performance was assessed using R^2 and MSE, for both training and test sets. A summary of these results is presented in Table 3.2.

Across all datasets, GPR and SVR delivered the strongest and most consistent predictive performance, achieving near-perfect R^2 and very low MSE. ANNs also performed well, though their accuracy declined slightly on noisy data. Ensemble methods such as Bagging, RF, and ET remained stable and resilient under data reduction and noise, whereas DT showed clear sensitivity to data quality, with reduced accuracy in the noisy dataset. XGBoost achieved excellent generalization with high test R^2 and low MSE across all datasets, making it well-suited for real-world conditions. In contrast, KNN displayed signs of overfitting, with perfect training accuracy but weaker test performance, especially in reduced and noisy datasets. Despite these differences, all nine models were carried forward to the MOO stage to enable a comprehensive comparison of both predictive capability and effectiveness in approximating Pareto optimal trade offs between conversion and yield.

3.5.2 Optimization

3.5.2.1 NSGA-II Pareto domains and NFM ranking for each model and dataset

Following the development and validation of the nine ML models for each of the three datasets, the subsequent step was to formulate the optimization problem within a MOO framework using the NSGA-II algorithm. This process required several preparatory stages. First, the two objective functions, namely ethylbenzene conversion and styrene yield, were defined using the trained and saved ML models. To ensure consistency across all optimization scenarios, the process input variables were constrained to their respective minimum and maximum bounds as observed in the original datasets. As previously noted, identical boundary values were applied across all three datasets to facilitate an unbiased comparative analysis. Beyond these bounds, two constraints were

incorporated to more accurately reflect the underlying process characteristics and to prevent extrapolation beyond the scope of the training data. The first constraint concerned the ratio of steam flow rate to ethylbenzene flow rate, which was restricted to the range of 8–12 in the original simulation data to align with real-world operating conditions. Accordingly, the optimization search space was constrained to ensure this ratio remained within that range. The second constraint addressed the issue of the outlet pressure, which was not included as an input variable during ML model training but was available in the simulation data. To ensure physically realistic solutions, a separate ANN with a single hidden layer was trained using the same four input variables employed in the main ML models (steam and ethylbenzene flow rates, inlet temperature, and inlet pressure) to predict the pressure drop and thus estimate the outlet pressure. This auxiliary model predicted the pressure drop with high accuracy (MSE of 5.53×10^{-5} and R^2 of 0.998 for the test data). The predicted pressure drop was then used to calculate the outlet pressure as the difference between the inlet pressure and the predicted pressure drop, with the constraint that the outlet pressure must remain at or above 100 kPa. A simple ANN was selected for this auxiliary task because it offered an accurate and computationally efficient nonlinear surrogate for repeated pressure-drop evaluations within the optimization loop. Since the pressure-drop model served only as a constraint-enforcement tool and not as part of the main model-comparison framework, a compact ANN architecture was adopted. Together, these two constraints confined the optimization search to a physically valid and data-supported operational space, thereby improving the reliability and robustness of the MOO results. Therefore, in the present framework, physical consistency was primarily supported through explicit process constraints on the optimization search space, while the surrogate outputs were left unconstrained to assess their behaviour under different data-quality conditions.

The NSGA-II algorithm was configured with a population size of 1,000 and 10,000 generations, providing sufficient search depth to capture a well-defined Pareto front. Although repeated independent runs could further characterize stochastic variability, the large population size and number of generations were selected to promote stable convergence and consistent front approximation across all cases. To further examine the effect of random seed on the NSGA-II results, the optimization was repeated 10 times for the ANN model trained on the 1000-record clean dataset. The overlaid Pareto fronts shown in Figure A.14 are nearly indistinguishable, demonstrating that seed-related variability is negligible under the adopted optimization settings.

This supports the robustness of the selected NSGA-II configuration and indicates that the observed Pareto fronts are stable for this representative model–dataset combination. Algorithm parameters were further tuned to balance convergence accuracy with solution diversity, thereby enabling efficient and robust identification of non-dominated solutions.

Table 3.2. Model performance comparison for all models on the complete and 1000-record datasets with and without noise on both training and test data.

		Conversion				**Yield**			
		Training		Test		Training		Test	
		R^2	MSE	R^2	MSE	R^2	MSE	R^2	MSE
Complete dataset	ANN	0.999	4.10×10^{-5}	0.999	4.00×10^{-5}	0.999	1.73×10^{-5}	1.000	1.55×10^{-5}
	GPR	1.000	1.65×10^{-6}	1.000	2.10×10^{-6}	1.000	1.11×10^{-6}	1.000	1.51×10^{-6}
	KNN	1.000	0	0.998	9.64×10^{-5}	1.000	0	0.997	8.09×10^{-5}
	SVR	1.000	1.89×10^{-6}	1.000	1.89×10^{-6}	1.000	1.69×10^{-6}	1.000	1.71×10^{-6}
	DT	1.000	2.94×10^{-5}	0.998	1.29×10^{-4}	0.998	7.20×10^{-5}	0.990	3.16×10^{-4}
	ET	1.000	1.63×10^{-5}	0.990	4.02×10^{-5}	0.999	3.08×10^{-5}	0.998	7.45×10^{-5}
	Bagging	1.000	8.77×10^{-6}	0.999	3.61×10^{-5}	0.999	1.82×10^{-5}	0.998	7.43×10^{-5}
	RF	1.000	5.27×10^{-6}	0.999	3.99×10^{-5}	1.000	1.11×10^{-5}	0.997	8.50×10^{-5}
	XGBoost	0.999	4.09×10^{-5}	0.999	4.83×10^{-5}	0.998	4.83×10^{-5}	0.998	7.26×10^{-5}
1000-record clean dataset	ANN	1.000	3.07×10^{-5}	1.000	2.77×10^{-5}	1.000	1.58×10^{-5}	0.999	1.45×10^{-5}
	GPR	1.000	1.33×10^{-6}	1.000	3.16×10^{-6}	1.000	9.01×10^{-7}	1.000	3.72×10^{-6}
	KNN	1.000	0	0.984	9.33×10^{-4}	1.000	0	0.976	5.72×10^{-4}
	SVR	1.000	2.84×10^{-6}	1.000	3.89×10^{-6}	1.000	6.19×10^{-6}	1.000	5.98×10^{-6}
	DT	0.997	2.27×10^{-4}	0.988	6.71×10^{-4}	0.986	4.37×10^{-4}	0.914	2.06×10^{-3}
	ET	0.993	4.62×10^{-4}	0.991	5.22×10^{-4}	0.989	3.53×10^{-4}	0.981	4.51×10^{-4}
	Bagging	0.999	6.12×10^{-5}	0.994	3.66×10^{-4}	0.996	1.28×10^{-4}	0.975	5.89×10^{-4}
	RF	0.995	3.63×10^{-4}	0.990	5.83×10^{-4}	0.996	1.18×10^{-4}	0.970	7.05×10^{-4}
	XGBoost	0.997	2.03×10^{-4}	0.993	3.86×10^{-4}	0.992	2.66×10^{-4}	0.982	4.26×10^{-4}
1000-record noisy dataset	ANN	0.961	2.45×10^{-3}	0.963	2.18×10^{-3}	0.962	9.09×10^{-4}	0.934	1.41×10^{-3}
	GPR	0.960	2.59×10^{-3}	0.964	2.08×10^{-3}	0.965	1.08×10^{-3}	0.943	1.31×10^{-3}
	KNN	1.000	0	0.947	3.00×10^{-3}	1.000	0	0.926	1.69×10^{-3}
	SVR	0.960	2.59×10^{-3}	0.964	2.08×10^{-3}	0.961	1.22×10^{-3}	0.942	1.34×10^{-3}
	DT	0.951	3.16×10^{-3}	0.954	2.64×10^{-3}	0.974	7.97×10^{-4}	0.875	2.86×10^{-3}
	ET	0.957	2.79×10^{-3}	0.956	2.53×10^{-3}	0.970	9.22×10^{-4}	0.927	1.67×10^{-3}
	Bagging	0.988	7.96×10^{-4}	0.950	2.87×10^{-3}	0.983	5.22×10^{-4}	0.928	1.66×10^{-3}
	RF	0.968	2.07×10^{-3}	0.953	2.69×10^{-3}	0.991	2.90×10^{-4}	0.922	1.79×10^{-3}
	XGBoost	0.966	2.21×10^{-3}	0.958	2.73×10^{-3}	0.971	9.01×10^{-4}	0.931	1.57×10^{-3}

Once the Pareto-optimal solution set was identified, the solutions were ranked using the NFM. The parameters applied in the NFM ranking procedure are summarized in Table 3.3. The NFM parameters were selected to provide a practical level of discrimination among Pareto-optimal

solutions while remaining consistent with the numerical spread of conversion and yield values observed along the Pareto fronts. Equal weights were assigned to both objectives because the study aimed to maximize ethylbenzene conversion and styrene yield simultaneously without prioritizing one over the other. The thresholds Q_m, P_m, and V_m were chosen so that small differences between solutions were treated as negligible, intermediate differences reflected a clear preference, and only sufficiently large deteriorations triggered a veto. Larger threshold values were assigned to conversion than to yield because conversion exhibited a broader variation range across the Pareto fronts. These parameters also reflect expert decision preference within the MCDM framework and can therefore be adjusted depending on the priorities and judgment of the decision-makers.

Before applying the NFM ranking procedure, the ML-predicted ethylbenzene conversion and styrene yield values were de-normalized from the normalized output scale and expressed as percentages. Therefore, the NFM parameters reported in Table 3 were defined and applied on the percentage scale of conversion and yield, and the following optimization results are presented as percentages from this point onward.

Table 3.3. NFM parameters for ethylbenzene conversion and styrene yield.

Parameters	Conversion	Yield
W_m **(relative weight)**	0.5	0.5
Q_m **(indifference threshold)**	5	3
P_m **(preference threshold)**	15	7
V_m **(veto threshold)**	25	12

Figure 3.3 to Figure 3.5 present the results of MOO using the NSGA-II algorithm, followed by solution ranking with the NFM, for the nine supervised ML models trained on the three datasets of varying size and quality: the complete dataset, the 1000-record clean dataset, and the 1000-record noisy dataset. The evaluated models include ANN, GPR, KNN, SVR, DT, bagging, RF, ET, and XGBoost. Each subplot displays the Pareto front corresponding to a specific model. Solution points are colour-coded according to their NFM rank, with the highest-ranked ("champion") solution highlighted by a red triangle.

Across the complete dataset, all nine models generated smooth, continuous, and closely comparable Pareto fronts. The distribution of ranked solutions along each front shows minimal structural differences between models, with both champion and top-tier solutions consistently

clustered within the same high-performance region in the conversion–yield space. This strong similarity suggests that, for the complete dataset, all models effectively captured the underlying process relationships with sufficient accuracy. As a result, the optimization outcomes were nearly identical, indicating a high degree of consistency in both predictive accuracy and generalization capability.

In the 1000-record clean dataset, although the number of training points was substantially reduced, most of the top-performing models preserved the overall structure of their Pareto fronts, indicating good robustness to data reduction. ANN and GPR maintained high-quality trade-off surfaces with dense clusters of high-ranking solutions, reinforcing their strong generalization capability under reduced-data conditions. SVR, XGBoost, and RF also continued to perform effectively, although a slight reduction in solution diversity was observed. In contrast, DT, ET, and particularly KNN showed greater sensitivity to the smaller sample size, leading to more fragmented Pareto fronts and fewer viable solutions identified by the NFM.

For the 1000-record noisy dataset, the Pareto fronts and NFM rankings exhibited the greatest variability and performance degradation among the three datasets. ANN and GPR maintained smooth, well-defined fronts with champion solutions still located in the high-performance region, though their curvature was less continuous than in the complete and clean dataset. SVR maintained a reasonable front shape but exhibited reduced density of high-ranking solutions. In contrast, KNN and DT suffered substantial performance deterioration: KNN's front became highly fragmented with large gaps and widely dispersed points, while DT produced only a small set of scattered elite solutions, failing to approximate a continuous trade-off surface. Bagging, RF, ET, and XGBoost also displayed reduced uniformity and density, with irregularities and shorter high-quality segments than those of the clean dataset. Compared to the complete dataset, where all models produced similarly smooth and dense Pareto fronts, and the clean 1000-record dataset, where only moderate differences emerged, the noisy dataset results demonstrate that noise in a reduced sample size disproportionately impacts algorithms that are particularly sensitive to training data quality, leading to weaker generalization and diminished coverage of the optimal conversion–yield trade-off space. These findings underscore the importance of considering both model performance and optimization outcomes to ensure a more comprehensive evaluation. For models with comparable predictive accuracy, selection should not rely solely on training metrics but must also account for

their effectiveness during the MOO phase. It is also noteworthy that, because no explicit output bounds were imposed during surrogate training or optimization, slight yield predictions above 100% were observed only for GPR, SVR, and XGBoost in the 1000-record noisy dataset, indicating that physically inconsistent predictions were limited to the most degraded data scenario.

Beyond the individual champion solution, the distribution of the top 10% ranked solutions also reveals the effect of ML model choice on optimization performance. In the complete dataset, the top-ranked solutions from most models were densely concentrated in the same high-performance region, indicating strong agreement across models. In the 1000-record clean dataset, ANN and GPR maintained relatively coherent top-ranked regions, while DT, ET, and especially KNN showed greater fragmentation and fewer highly ranked solutions. This effect became more pronounced in the 1000-record noisy dataset, where ANN and GPR remained comparatively stable, whereas the weaker models exhibited more scattered and less reliable elite solution sets. These results confirm that model choice influences not only the champion point but also the overall robustness and continuity of the high-performing region of the Pareto front.

These findings underscore the importance of considering both predictive performance and optimization behaviour when evaluating surrogate models for reactor optimization. Although each ML model produced distinct optimization outcomes under different data conditions, these differences were not limited to the simple presence of noise. In the complete dataset, the optimization outcomes were highly consistent across models, with smooth Pareto fronts, dense clusters of high-ranked solutions, and similar champion regions. This indicates that the large sample size provided sufficient coverage of the feasible operating domain for most models to learn the underlying response surface reliably. In the 1000-record clean dataset, the main change was not a loss of physical consistency but a reduction in front continuity, solution density, and ranking stability for the more data-sensitive models, indicating that the smaller sample size reduced the ability of some surrogates to represent the trade-off surface with the same fidelity as in the complete dataset. In the 1000-record noisy dataset, the degradation became more pronounced, as limited sample coverage and output perturbation acted together to produce fragmented fronts, more scattered elite solutions, and, in a few cases, slight physically inconsistent predictions.

These results show that the relationship between predictive accuracy and MOO effectiveness is not strictly one-to-one. Although poorer predictive performance often corresponds to weaker

Pareto fronts, especially under reduced-data and noisy-data conditions, models with similar MSE and R^2 can still produce markedly different optimization outcomes. This occurs because MOO depends not only on global test-set accuracy, but also on how well a surrogate preserves the local smoothness, continuity, ranking fidelity, and realistic trade-off structure of the response surface in the high-performing region explored by the optimizer. In this study, the distinct optimization outcomes were reflected in differences in Pareto-front smoothness, continuity, density of top-ranked solutions, stability of champion regions, contribution to the overall non-dominated front, and robustness under degraded data conditions. More robust models, such as ANN and GPR, preserved smoother and more coherent Pareto structures when sample size was reduced, or noise was introduced, whereas models such as DT and KNN were more strongly affected by reduced sample coverage and noise sensitivity.

Accordingly, model effectiveness for MOO should not be judged solely by global predictive metrics, but also by optimization-oriented indicators such as Pareto-front continuity, density of top-ranked solutions, contribution to the overall non-dominated front, agreement of champion solutions with the original simulator, and computational efficiency. More importantly, such differences may be anticipated before performing full MOO by complementing predictive metrics with local validation in the near-optimal region, uncertainty or stability analysis, and inspection of surrogate-surface behaviour. These additional checks can help identify whether a model is likely to provide a reliable and well-structured optimization landscape, rather than only a good overall statistical fit. Nevertheless, no model is excluded from the overall framework. Rather than discarding models based solely on predictive or MOO performance, the following sections present an integrated approach that leverages the strengths of all trained models to enhance decision-making under uncertainty.

The 2D decision space plots are presented in Figure A.8 to Figure A.13, showing the optimal values of ethylbenzene and steam flow rates, inlet temperature, and inlet pressure corresponding to each model's Pareto front across all three datasets. These plots reflect the constrained optimization framework described earlier, ensuring that all solutions satisfy the steam-to-ethylbenzene ratio and outlet pressure requirements. In the case of the steam-to-ethylbenzene flow rate ratio (Figure A.8 to Figure A.10), most models (particularly ANN, GPR, SVR, and XGBoost) consistently selected the upper bound of 12. Although some models explored intermediate ratios

within the allowed range of 8–12, the majority of solutions were clustered at the upper limit. This indicates that the models have learned implicitly that increasing the steam ratio reduces the partial pressure of ethylbenzene, promotes the forward dehydrogenation reaction, enhances heat transfer, and suppresses coke formation on the catalyst surface, all of which contribute to enhanced conversion and yield.

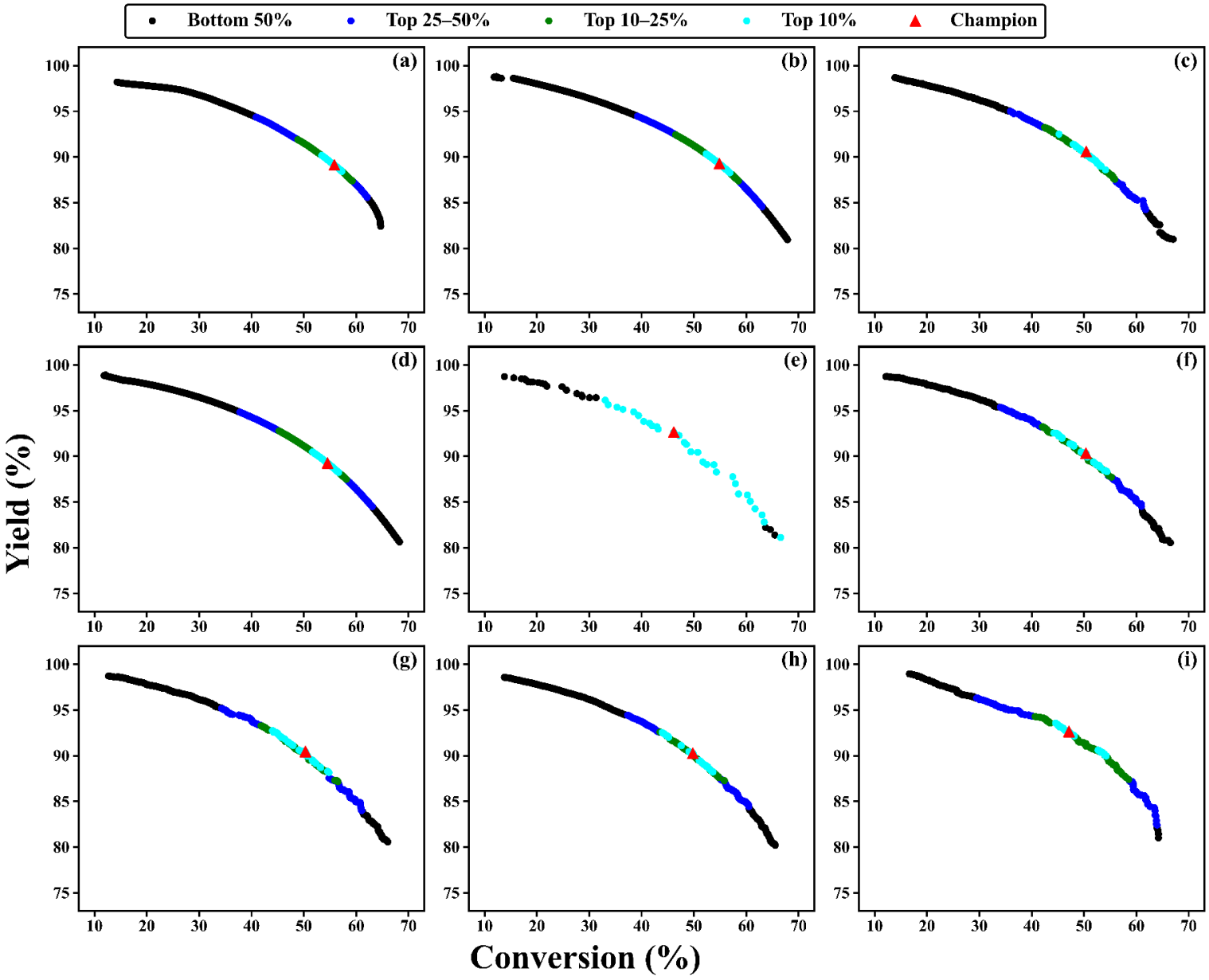


Figure 3.3. Pareto fronts and NFM rankings for models trained on the complete dataset. Each plot corresponds respectively to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost.

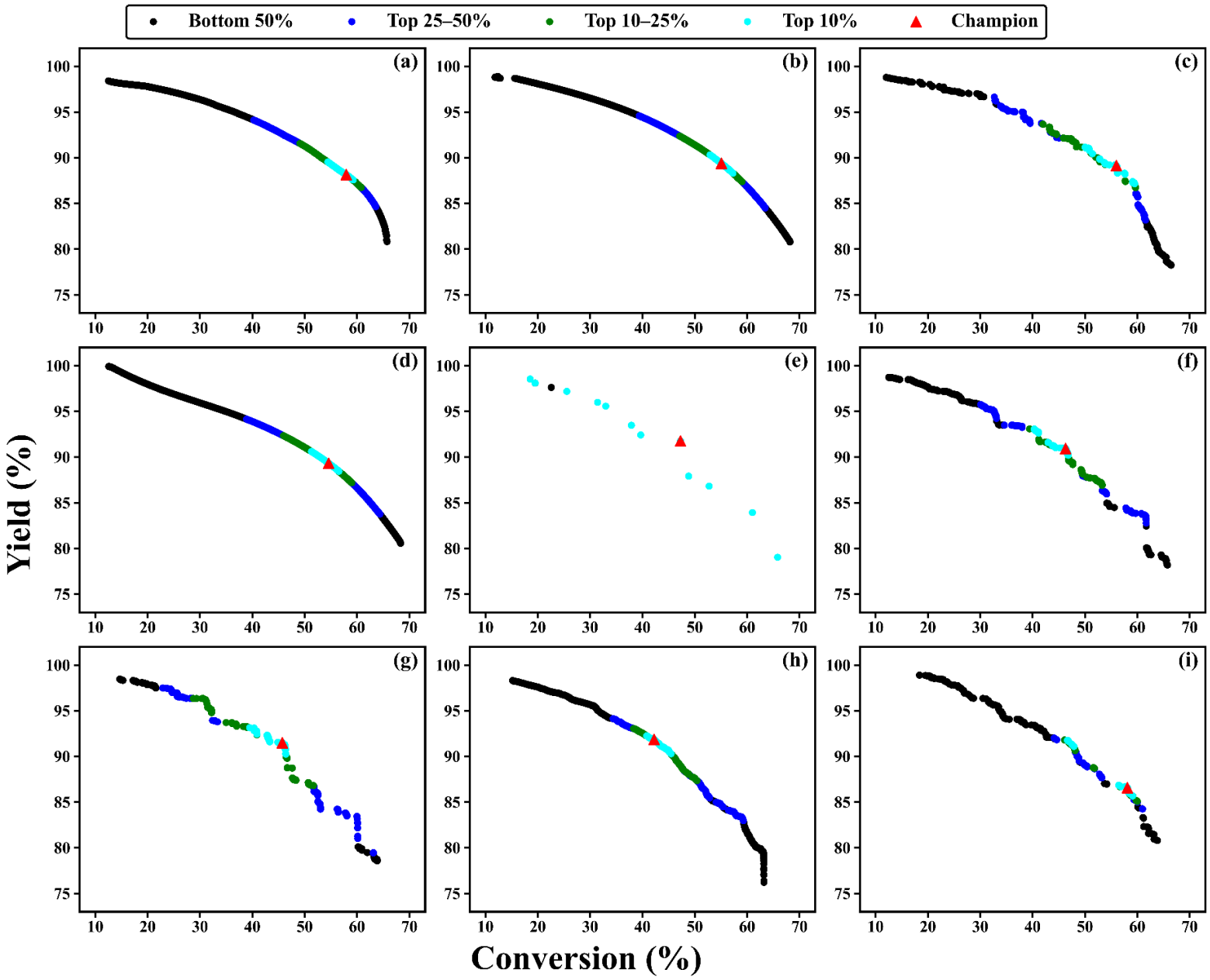


Figure 3.4. Pareto fronts and NFM rankings for models trained on the 1000-record clean dataset. Each plot corresponds respectively to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost.

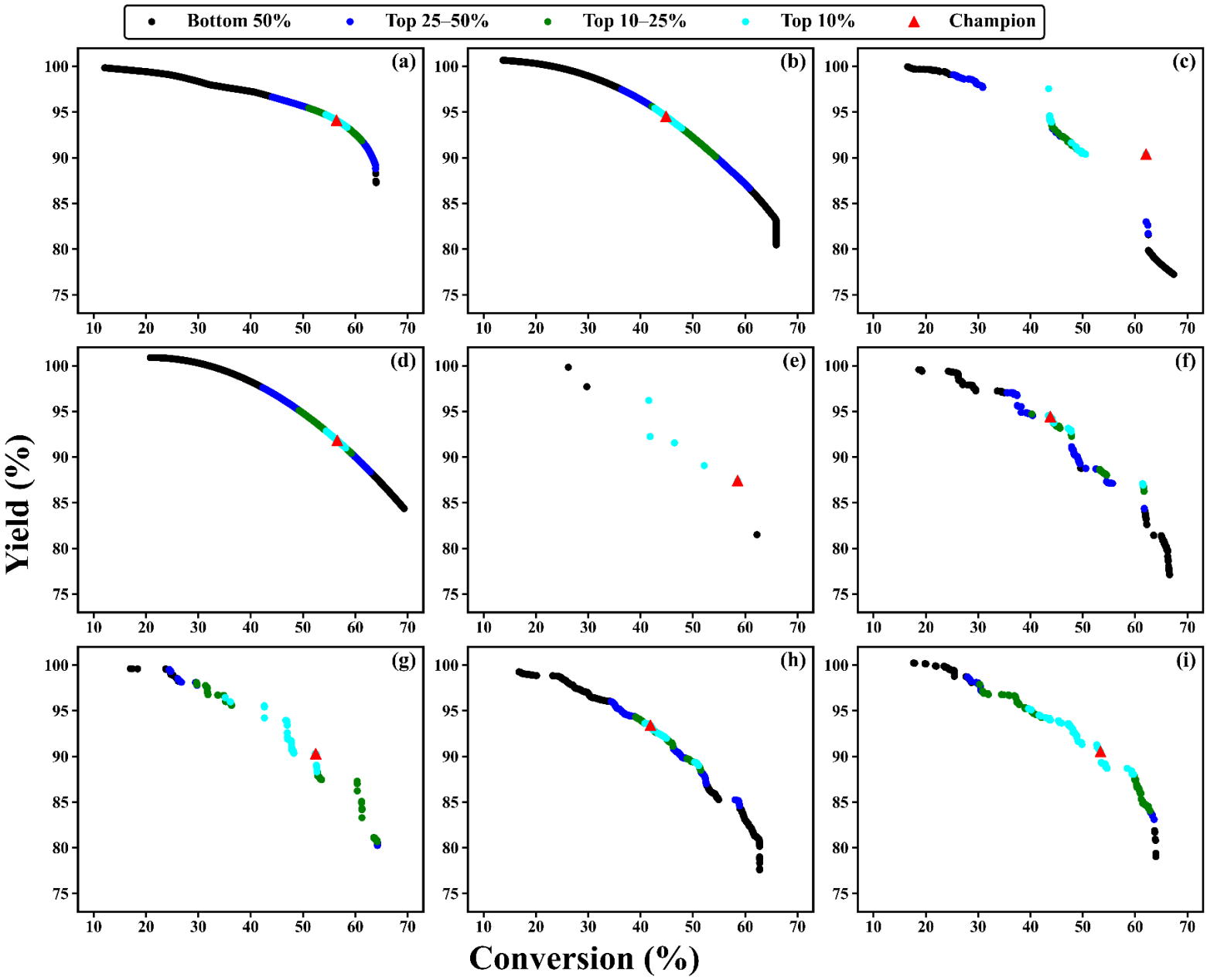


Figure 3.5. Pareto fronts and NFM rankings for models trained on the 1000-record noisy dataset. Each plot corresponds respectively to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost.

Figure A.11 to Figure A.13 depict the optimal inlet pressure and temperature values across all models and datasets. Optimal pressures consistently converged at the lowest allowable limits, satisfying the second constraint and aligning with thermodynamic principles. Since styrene production results in a net increase in the number of moles, lower pressures shift the equilibrium

toward products (Le Chatelier's law) while reducing secondary hydrogenation and cracking pathways. Temperature values showed a broader distribution. Higher temperatures were generally selected in solutions aiming to maximize conversion, consistent with the endothermic nature of ethylbenzene dehydrogenation. Conversely, for high-yield solutions with reduced by-product formation, lower temperatures were selected to limit undesirable secondary reactions. These decision space patterns highlight how, under realistic physical constraints, the models tend to converge toward similar optimal operating strategies while still exhibiting subtle differences in how they explore the feasible operating conditions.

Although each ML model produced distinct optimization outcomes under different data conditions, it is important to note that no model is excluded from the overall framework. Rather than discarding models based solely on predictive or MOO performance, the following sections of this paper present an integrated approach that leverages the strengths of all trained models to enhance decision-making under uncertainty. This approach ensures that every model, regardless of its individual ranking, can contribute meaningfully to robust and comprehensive optimization outcomes.

3.5.2.2 Comparison of dominancy of models

To facilitate direct comparison across models, Figure 3.6 presents the non-dominated set of the combined Pareto-optimal solutions generated by the nine ML models across three distinct training scenarios: (a) complete dataset, (b) 1000-record clean dataset, and (c) 1000-record noisy dataset. Each plot contains only the overall non-dominated solutions, enabling the direct comparison of model contributions under varying data conditions.

Figure 3.6a (complete dataset) shows that the ANN dominates the Pareto front, forming a smooth arc that spans the entire conversion–yield range. XGBoost follows closely, occupying a substantial mid-to-high conversion region, while SVR contributes mainly in the high-conversion region. The remaining models provide only isolated points, exerting minimal impact on the overall front. Figure 3.6b (clean reduced dataset) reveals a shift in dominance with GPR emerging as the leading contributor. It generates a dense cluster of points concentrated in the mid-to-high conversion region. ANN remains competitive, forming a shorter yet continuous arc. XGBoost and KNN also contribute noticeably, particularly in high-yield areas of the Pareto front. SVR's contribution is reduced to a few scattered points. In Figure 3.6c (noisy reduced dataset), SVR

emerges as the dominant model, leading the high-conversion/high-yield segment of the Pareto front. However, the yield values predicted by SVR exceed 100%, and the conversion values surpass the equilibrium limit, both physically unrealistic outcomes resulting from the noise introduced into the reduced dataset. ANN retains coverage of the mid-conversion region, maintaining a smooth front despite the noise. All the other models, including GPR, which was dominant in the clean dataset, are absent from the non-dominated set.

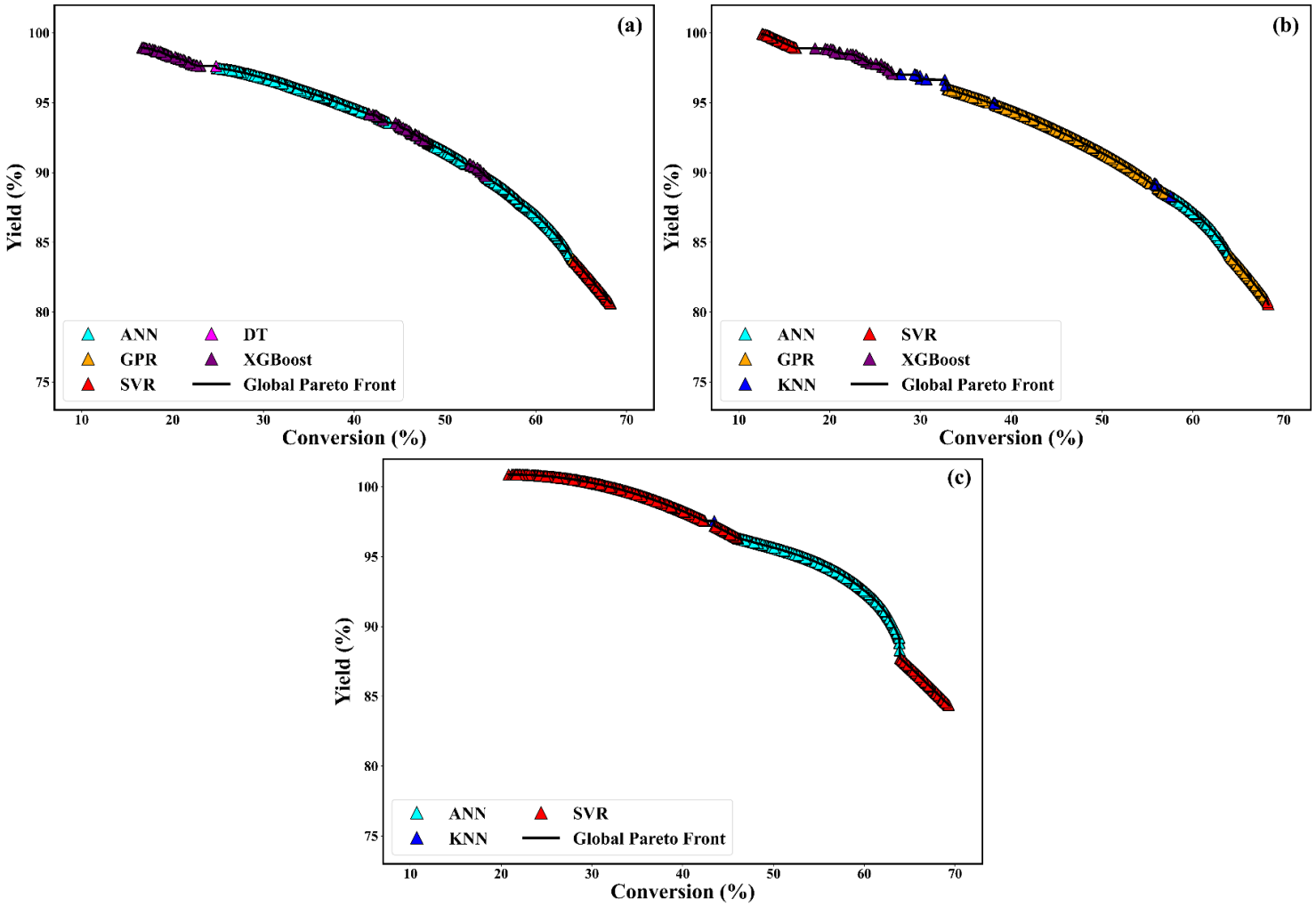


Figure 3.6. Non-dominated Pareto-optimal solutions for nine ML models across three training scenarios: (a) complete dataset, (b) 1000-record clean dataset, and (c) 1000-record noisy dataset.

Figure A.15 in the supporting information presents the full overlay of each model-specific Pareto front with the corresponding global Pareto front, allowing a more direct visual assessment of how closely each model follows the overall best trade-off surface. The comparison shows that the fronts

remain highly consistent for the complete dataset, whereas deviations become more visible in the reduced clean dataset and are most pronounced in the noisy dataset.

Overall, ANN, GPR, and SVR each demonstrate dominance under specific conditions, though their robustness differs. ANN excels with large or noisy data sets, GPR thrives with smaller clean datasets, and SVR exhibits the greatest resilience to noise. Data reduction generally narrows and shifts the Pareto fronts, while the introduction of noise further fragments them. These observations highlight the importance of selecting models capable of maintaining optimality under realistic, imperfect data conditions.

3.5.2.3 Computation time

Beyond dominance performance, the computational cost and efficiency of each ML model were assessed in terms of training time, optimization time, and the number of non-dominated solutions when all Pareto fronts are compared, as shown in Figure 3.7 to Figure 3.9. All modelling and optimization processes were conducted on the Cedar cluster, a national high-performance computing system operated by the Digital Research Alliance of Canada. Cedar provides scalable access to large CPU and GPU resources (Cedar), supporting parallel computing tasks commonly used in data-driven scientific research and optimization.

For the complete dataset, ANN demonstrated the highest optimization performance, producing 649 non-dominated solutions with a training time of 31.1 h and an optimization time of 0.8 h. XGBoost ranked second, generating 450 non-dominated points with minimal computational demand (0.06 h for training and 0.8 h for optimization). SVR yielded 72 non-dominated solutions, requiring 25.9 h of training and 0.9 h of optimization. GPR produced only 32 non-dominated solutions while incurring a significantly higher optimization time (93.2 h); its training time (35.8 h) reflects a single trial, constrained by the model's computational limitations when applied to large datasets. The reported 93.2 h corresponds to the NSGA-II optimization stage using the trained GPR surrogate, not to the initial training stage. During NSGA-II, each candidate solution generated across the population and generations must be evaluated by the trained surrogate model to predict conversion and yield. Therefore, optimization time depends not only on the NSGA-II settings, such as population size and number of generations, but also on the computational cost of each surrogate-model prediction. For GPR, prediction requires kernel-based calculations involving the training data, so the cost of repeated evaluations during NSGA-II increases substantially when

the surrogate is trained on the complete dataset. This explains why the GPR optimization time was much larger for the complete dataset than for the 1000-record clean dataset, whereas models with faster prediction routines, such as ANN, XGBoost, and tree-based ensembles, showed much lower optimization times. When trained on the 1000-record clean dataset, computation times decreased markedly. GPR emerged as the top performer, producing 527 non-dominated solutions with exceptionally low computational requirements (0.02 h for training, 0.9 h for optimization). ANN remained competitive, achieving 201 non-dominated solutions with a training time of 1.0 h and an optimization time of 0.5. SVR generated 66 non-dominated solutions, though with a substantially higher training time (17.5 h). XGBoost and KNN achieved comparable numbers of non-dominated points to SVR, but with significantly lower computation times. Under the 1000-record noisy dataset scenario, SVR and ANN dominated, producing 529 and 446 non-dominated solutions, respectively. ANN sustained strong performance with modest computation time, whereas SVR required 25.3 h of training to achieve its results.

Overall, these findings indicate that ANN consistently offers the most favourable balance between optimization quality and computational efficiency across diverse data conditions. While GPR and SVR can perform competitively with smaller datasets, their sensitivity to data scale is more pronounced. These findings highlight the importance of jointly considering solution quality, runtime, and model robustness when selecting ML models for MOO in engineering applications.

These differences in performance are consistent with the underlying properties of the ML models and their sensitivity to dataset size and quality. ANN maintained strong performance across all cases because its flexible nonlinear approximation capability allowed it to preserve good surrogate quality while remaining computationally efficient during optimization. GPR performed particularly well for the 1000-record clean dataset because its kernel-based formulation is highly effective for modelling smooth nonlinear relationships when the dataset is relatively small and clean; however, its computational burden increased sharply for the complete dataset, making it less practical at larger scale. SVR also remained competitive because of its strong generalization ability in nonlinear spaces, although its training time remained comparatively high. Ensemble and boosting methods such as RF, Bagging, ET, and XGBoost showed moderate robustness because averaging and regularization help stabilize predictions under degraded data conditions. In contrast, models such as KNN and DT were more sensitive to reduced sample coverage and noise, which

limited their ability to preserve smooth and reliable Pareto structures under more challenging data scenarios.

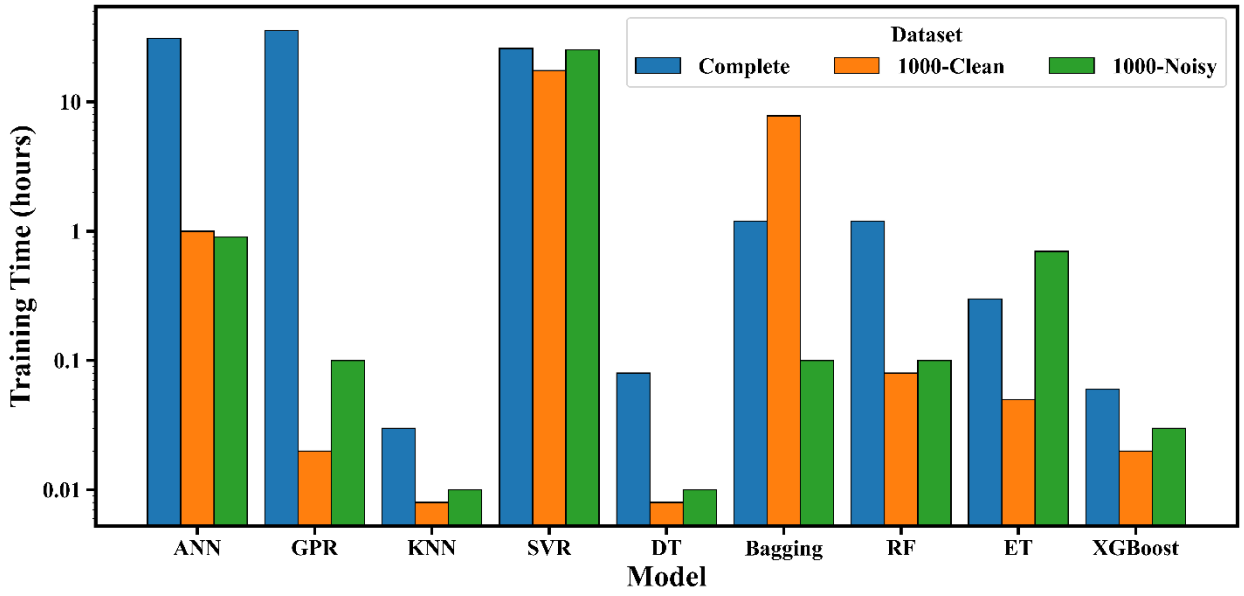


Figure 3.7. Comparison of hyperparameter tuning and training time plotted on a logarithmic scale.

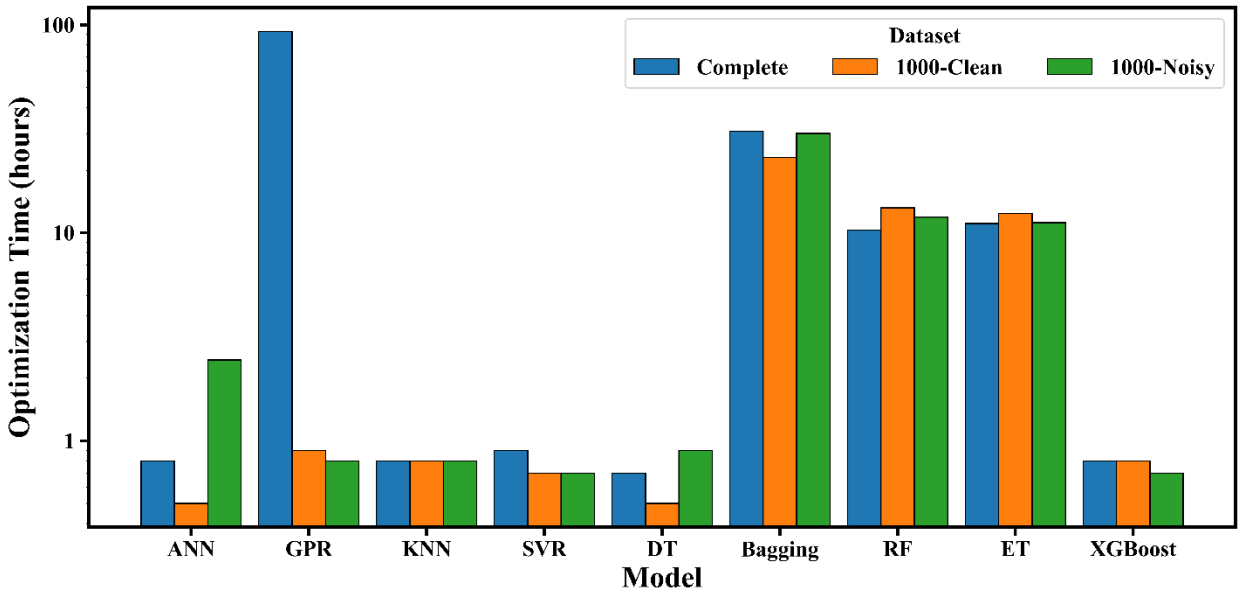


Figure 3.8. Comparison of optimization time plotted on a logarithmic scale.

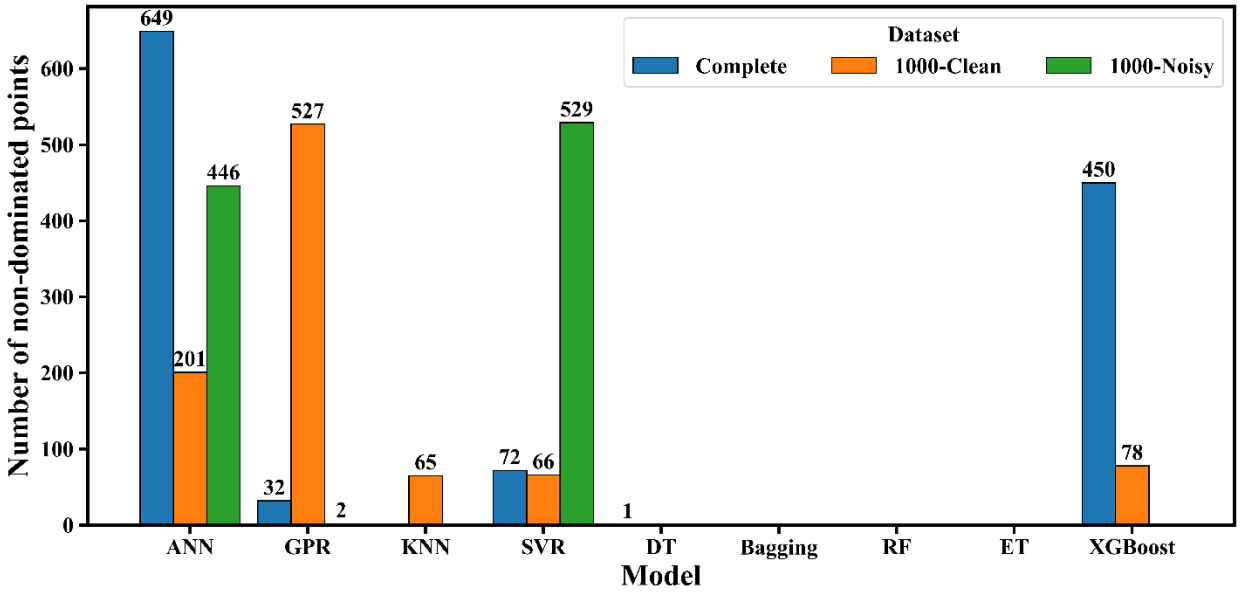


Figure 3.9. Number of Pareto-optimal (non-dominated) solutions generated by each model across the three datasets. Results were obtained using the NSGA-II algorithm with a population size of 1000.

3.5.2.4 Consensus approach to use the information from all the trained models

As outlined earlier, following the validation of the nine ML models, the NSGA-II algorithm was applied with conversion and yield as objectives. Decision variables were constrained within the bounds of the original data, supplemented by feasibility restrictions to ensure realistic operation. Specifically, the steam-to-ethylbenzene ratio was limited to the practical range observed in the simulations, and a minimum outlet pressure was enforced to prevent unrealistic conditions. These measures ensured that all optimized solutions remained physically meaningful and consistent with actual process behaviour.

The champion solutions for each model-dataset combination were obtained using the NFM ranking procedure. To further assess these champions, the corresponding input variable values from the optimal decision space were reapplied in the original simulation program to compute conversion and yield, enabling direct comparison with the model predictions. The outcomes of these simulations, together with the ethylbenzene and steam flow rates, inlet temperature and pressure, and the predicted conversion and yield for each model (champions), are presented in Figure 3.10 to Figure 3.12. The full set of champion solutions associated with these figures is also presented in Table A.12 of Appendix A. It is noteworthy that some models identified noticeably

different operating conditions while still achieving very similar conversion and yield values. This behaviour is not solely due to surrogate-model differences, since the champion points were re-evaluated in the original reactor simulation and remained in close agreement with the ML predictions. Rather, it indicates that the constrained styrene reactor system admits multiple feasible operating strategies that lie on near-equivalent trade-off regions in the objective space. In this context, the different ML models tend to explore and prioritize different parts of this admissible decision space while still converging to comparable reactor performance. While these figures summarize the top-ranked solution for each model, selecting a trial point from a single model may introduce bias toward that model's predictive tendencies.

To mitigate model-specific bias and ensure robust, generalizable recommendations, a consensus-based approach was applied to the champion solutions illustrated in Figure 3.10 to Figure 3.12. For each dataset, models were classified into majority and minority groups according to the similarity of their champion decision variables. The majority group comprised models whose champions clustered closely across all four decision variables, representing the dominant trend, while the minority group reflected an alternative but less frequent set of champions. Within each group, the final value of each decision variable was determined either by selecting the most frequently occurring value or, in the absence of clear repetition, by averaging the values. This method ensured that the resulting consensus points represented genuine cross-model agreement rather than isolated optima from individual champions.

The outcomes of this consensus process are presented in Table 3.4. For the complete dataset, the majority group, comprising GPR, KNN, SVR, DT, Bagging, RF, ET, and XGBoost, converged on the following operating conditions: 57.06 mol/s ethylbenzene, 682.96 mol/s steam, 887.97 K, and 154.07 kPa, achieving 50.44% conversion and 90.68% yield. Validation through simulation under identical input variables produced 50.60% conversion and 90.60% yield, demonstrating excellent agreement with the ML model predictions. In contrast, the minority group, represented solely by ANN, selected 79.75 mol/s ethylbenzene, 957.00 mol/s steam, 910.56 K, and 175.47 kPa, achieving a slightly higher 55.82% conversion and 89.14% yield. A corresponding validation simulation under identical operating conditions resulted in 54.30% conversion and 88.10% yield, showing close agreement with the ANN model predictions.

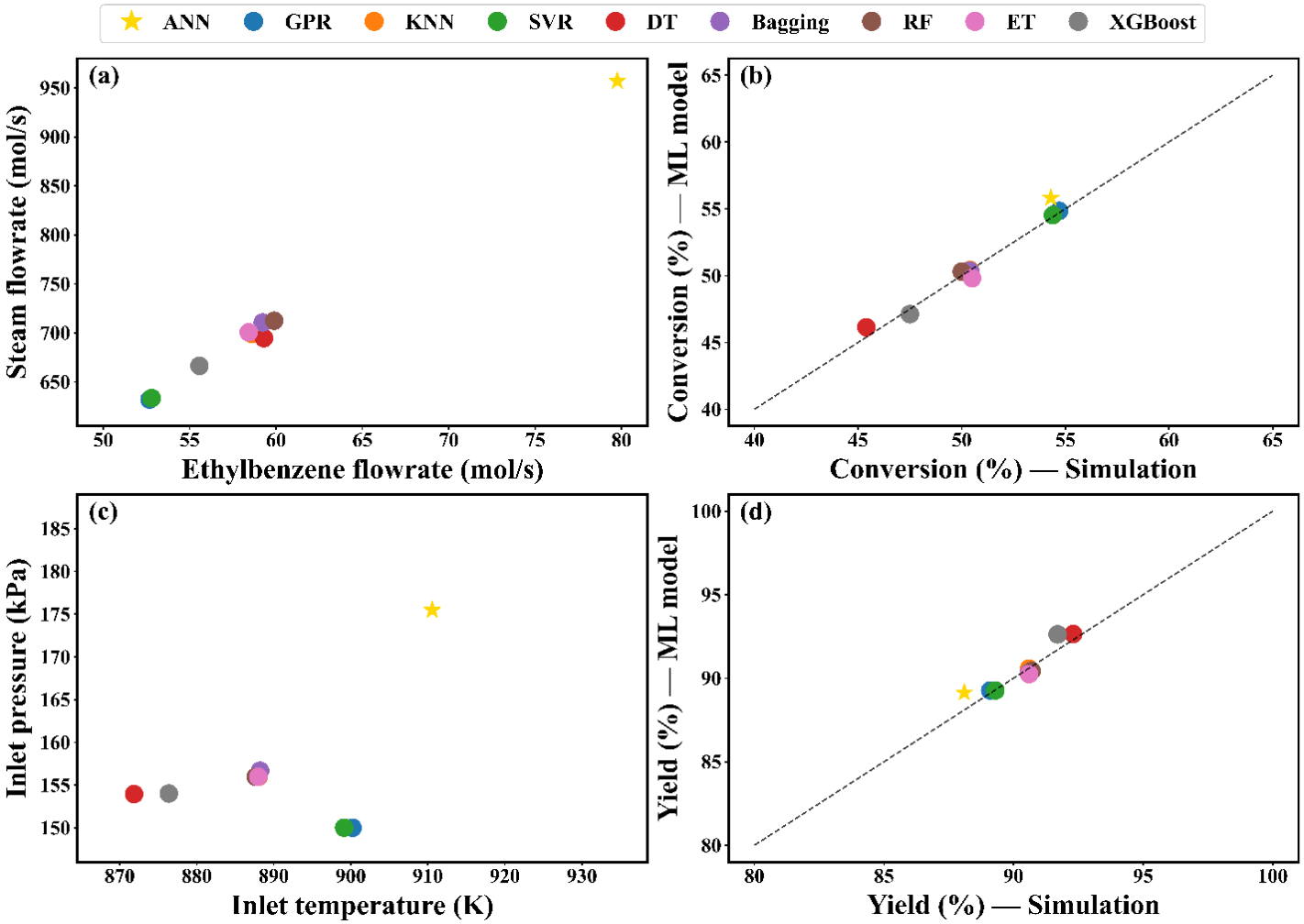


Figure 3.10. Champion points identified by the NFM for each model across both the objective and decision spaces using the complete dataset. Subplots (a) and (c) depict the optimal decision spaces, while (b) and (d) present the corresponding optimal objective values, which have been further validated through simulation.

For the 1000-record clean dataset, the majority group comprising ANN, KNN, DT, Bagging, RF, ET, XGBoost, converged on operating conditions of 72.81 mol/s ethylbenzene, 864.75 mol/s steam, 894.03 K, and 168.87 kPa, resulting in 50.51% conversion and 89.99% yield. Validation through simulation under identical input variables produced 50.30% conversion and 90.10% yield, further confirming the consistency of the ML predictions. The minority (GPR, SVR) selected operating conditions of 52.61 mol/s ethylbenzene, 631.24 mol/s steam, 899.31 K, and 150.00 kPa, achieving 54.83% conversion and 89.36% yield. A corresponding simulation led to a 54.50% conversion and 89.20% yield, reinforcing the reliability of these model predictions.

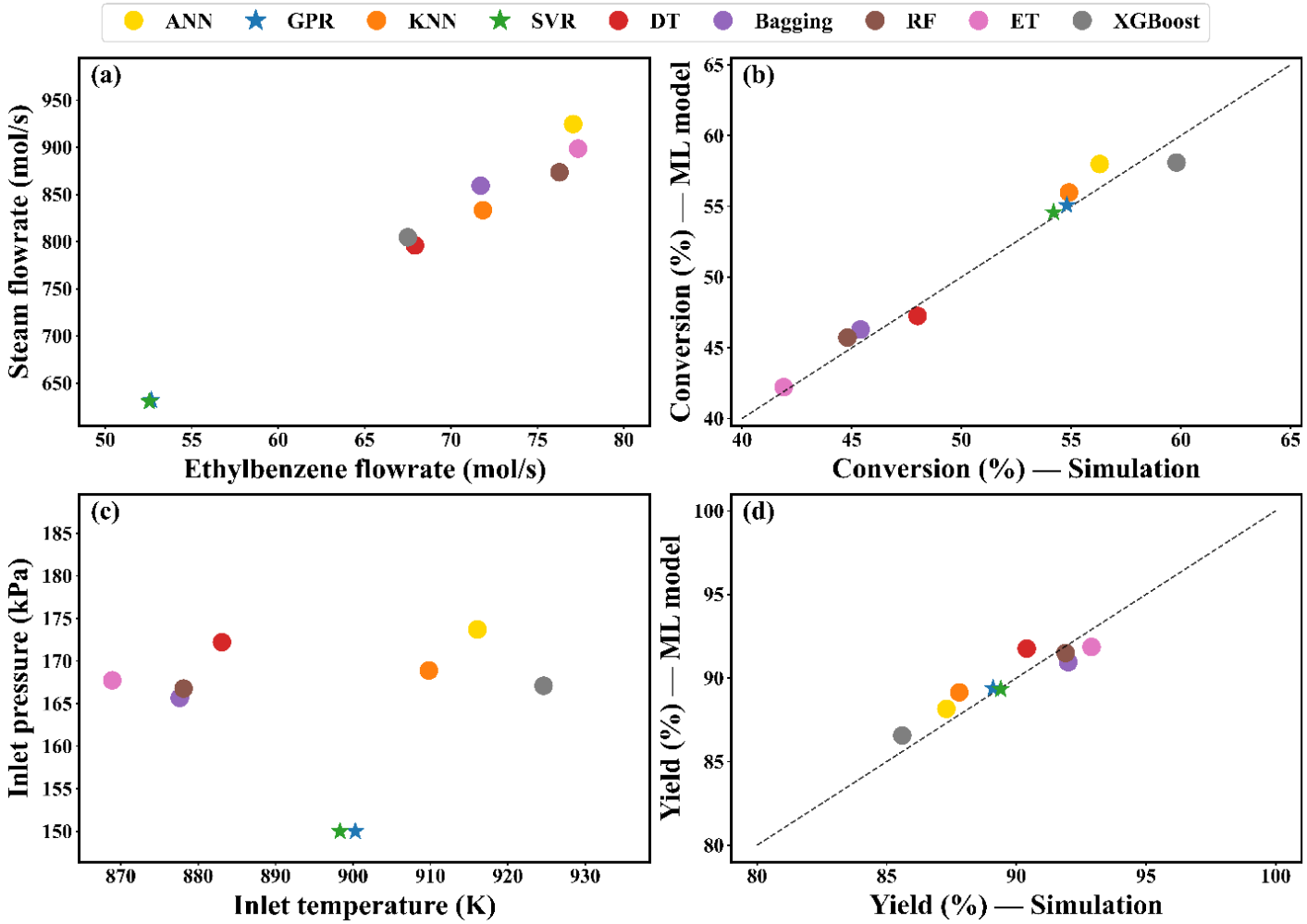


Figure 3.11. Champion points identified by the NFM for each model across both the objective and decision spaces using the 1000-record clean dataset. Subplots (a) and (c) depict the optimal decision spaces, while (b) and (d) present the corresponding optimal objective values, which have been further validated through simulation.

For the 1000-record noisy dataset, the majority (ANN, GPR, SVR, DT, Bagging, RF, ET, XGBoost) converged on operating conditions of 79.07 mol/s ethylbenzene, 948.78 mol/s steam, 895.74 K, and 176.30 kPa, yielding 50.98% conversion and 92.07% yield. A corresponding simulation with the same input conditions led to a 50.20% conversion and 89.80% yield, showing close agreement for conversion but a yield deviation of more than 2%, likely reflecting the sensitivity of yield predictions under noisy data conditions. The minority group, represented by KNN, identified operating conditions of 61.28 mol/s ethylbenzene, 719.91 mol/s steam, 930.98 K, and 167.07 kPa, achieving 62.09% conversion and 90.39% yield. A validation simulation under

the same operating conditions resulted in 62.00% conversion and 83.40% yield, indicating strong agreement for conversion but a yield deviation of about 7%, suggesting potential limitations in yield prediction accuracy in the case where noise is present. These larger discrepancies, particularly for yield in the 1000-record noisy dataset, are consistent with reduced local prediction fidelity under degraded data conditions, even when the surrogate retains reasonable overall predictive performance.

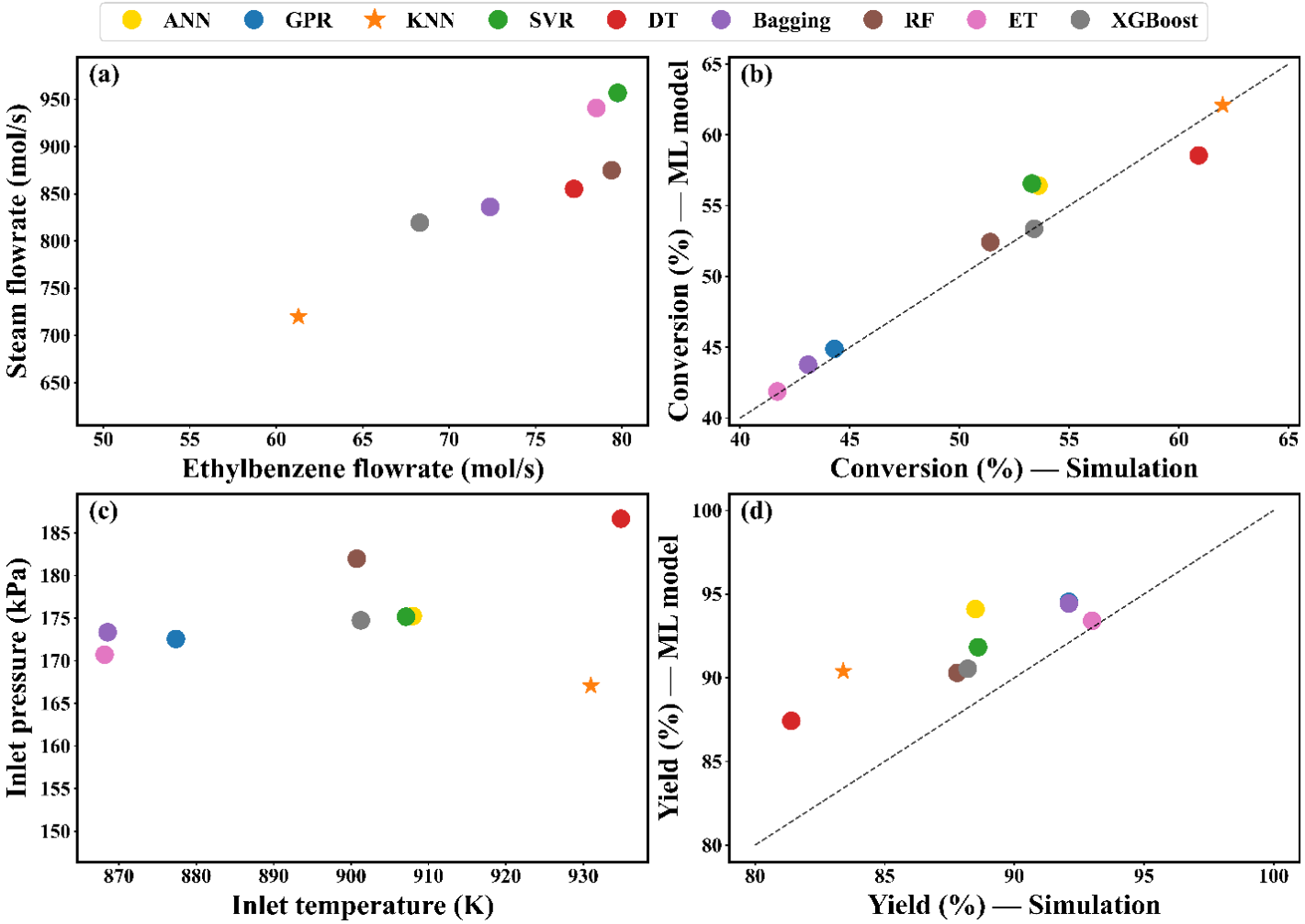


Figure 3.12. Champion points identified by the NFM for each model across both the objective and decision spaces using the 1000-record noisy dataset. Subplots (a) and (c) depict the optimal decision spaces, while (b) and (d) present the corresponding optimal objective values, which have been further validated through simulation.

An evaluation of the six consensus points across the datasets (Table 3.4) revealed a recurring pattern centred on two distinct operational strategies: (i) a high-flow, high-steam configuration and (ii) a low-flow, low-pressure configuration. In the complete dataset, the majority of models favoured the latter strategy, while in both the 1000-record clean and noisy datasets, the majority supported the former strategy. With the exception of the KNN minority case in the noisy dataset, both strategies across all datasets produced predictions that were closely aligned with simulation results, thereby providing reliable candidates for subsequent experimental testing. Although conversion and yield values are comparable under both strategies, their industrial implications differ: the high-flow, high-steam configuration is advantageous for reducing coke deposition and improving catalyst stability, whereas the low-flow, low-pressure approach can lower energy demand and operating costs. Nonetheless, in real industrial applications where only a single dataset is available and simulation cannot be used for validation, it is more rational to rely on the strategy recommended by the majority of models. Accordingly, the three majority-selected points from the different datasets are proposed as the next experimental conditions to investigate.

Table 3.4. Majority and minority consensus operating points for each dataset, derived from the champion solutions using the consensus-based selection approach under the imposed process constraints.

Dataset	Group	Ethylbenzene flow rate (mol/s)	Steam flow rate (mol/s)	Inlet temperature (K)	Inlet pressure (kPa)	Conversion (%) (ML model)	Conversion (%) (simulation)	Yield (%) (ML model)	Yield (%) (simulation)
Complete dataset	Majority	57.06	682.96	887.97	154.07	50.44	50.60	90.68	90.60
	Minority	79.75	957.00	910.56	175.47	55.82	54.30	89.14	88.10
1000-record clean dataset	Majority	72.81	864.75	894.03	168.87	50.51	50.30	89.99	90.10
	Minority	52.61	631.24	899.31	150.00	54.83	54.50	89.36	89.20
1000-record noisy dataset	Majority	79.07	948.78	895.74	176.30	50.98	50.20	92.07	89.80
	Minority	61.28	719.91	930.98	167.07	62.09	62.00	90.39	83.40

To further assess the robustness of the surrogate-based optimization results, a local inter-model uncertainty analysis was performed around the majority and minority consensus points identified in Table 3.4. Here, “local inter-model uncertainty” refers to the variability between different ML models in the neighbourhood of a selected consensus point, rather than to a full probabilistic

uncertainty estimate. For each consensus point and dataset, the nearest Pareto-optimal solution from each ML model was identified in normalized objective space using conversion and yield, and the dispersion across models was quantified. The full results are provided in Appendix A (Table A.13). Overall, the complete dataset showed the tightest agreement across models, particularly around the majority point, indicating a stable and well-supported operating strategy. The 1000-record clean dataset exhibited a moderately larger spread, while the 1000-record noisy dataset showed the greatest inter-model variability, confirming that uncertainty in the recommended operating conditions increases as data quality deteriorates. In general, the majority points tended to be more consistently supported than the minority points, further reinforcing their robustness as preferred operating recommendations. To complement the local inter-model uncertainty summary, an additional predictive uncertainty analysis was conducted for the GPR-based Pareto fronts of the 1000-record clean and 1000-record noisy datasets. The resulting confidence-band visualizations are presented in Figure A.16 of Appendix A. The figure shows that uncertainty remains relatively limited for the 1000-record clean dataset, whereas the 1000-record noisy dataset exhibits substantially wider confidence bands. In particular, the uncertainty associated with yield is generally larger than that of conversion, especially over the mid-to-high conversion region. This trend is consistent with the broader variability observed under degraded data quality and further supports the conclusion that noise reduces confidence in surrogate-based operating recommendations.

A further extension of the present framework would be the use of hybrid or physics-informed surrogate models that combine mechanistic process knowledge with data-driven ML corrections. In this study, the dataset was generated from a mechanistic styrene reactor simulation, whereas the predictive stage relied on purely data-driven surrogate models to enable a broad comparative analysis across nine ML methods. Future work could integrate these two levels more directly by embedding known thermodynamic relationships, process constraints, or mechanistic residual corrections into the surrogate formulation. Such hybrid approaches could improve extrapolation robustness, preserve physical consistency, and reduce sensitivity to noisy data during optimization, thereby further strengthening the reliability of the recommended operating conditions.

3.6 Conclusion

This study implemented a comprehensive MOO framework using NSGA-II algorithm, integrated with nine supervised ML models (ANN, GPR, SVR, KNN, DT, ET, Bagging, RF, and XGBoost) trained on datasets of varying size and quality. The objective was to optimize conversion and yield in an ethylbenzene dehydrogenation process under realistic conditions. These findings demonstrate that dataset quality and size exert significant influence on both the shape and density of Pareto fronts, as well as the robustness of resulting optimal solutions. Using the complete dataset, all models generated smooth and comparable Pareto fronts, indicative of strong generalization and consistent high-performance regions. In contrast, data reduction and noise introduced variability, fragmenting Pareto fronts and disproportionately affecting models more sensitive to data quality, such as KNN and DT. By comparison, models like ANN, GPR, and SVR exhibited greater resilience, each achieving superior performance under at least one scenario. These conclusions should be interpreted as specific to the styrene reactor case study considered here and should not be generalized directly to other chemical processes, datasets, or optimization settings without further validation.

Dominance analysis revealed that ANN excelled with complete and noisy data, GPR thrived with clean reduced datasets, and SVR showed the strongest resistance to noise. In terms of computational performance, ANN provided the most favourable balance between optimization quality and efficiency. By comparison, GPR and SVR yielded competitive outcomes on smaller datasets, albeit at the expense of higher training costs.

To reduce reliance on any single model and enhance the robustness of decision-making, a consensus-based framework is proposed to integrate solutions across models and datasets. This analysis consistently revealed two dominant operational strategies: a high-flow, high-steam configuration and a low-flow, low-pressure configuration, both delivering comparable conversion and yield outcomes. Although their performance metrics are closely aligned, the strategies differ in industrial implications, with the former favouring catalyst stability and the latter offering potential energy savings. Consequently, adopting the majority-recommended points provides a rational and practical basis for guiding future experimental validation.

Overall, the findings underscore the importance of evaluating ML models not only on predictive metrics but also on their optimization behaviour under realistic, imperfect data conditions. By incorporating a consensus strategy, the proposed framework leverages the complementary strengths of multiple models, ensuring robust and comprehensive recommendations for process operation and control.

3.7 References

1. Fischer, L., Ehrlinger, L., Geist, V., Ramler, R., Sobiezky, F., Zellinger, W., Brunner, D., Kumar, M., & Moser, B. (2020). Ai system engineering—Key challenges and lessons learned. *Machine Learning and Knowledge Extraction*, *3*(1), 56–83. https://doi.org/10.3390/make3010004
2. Holzinger, A. (2019). Introduction to machine learning & knowledge extraction (make). *Machine Learning and Knowledge Extraction*, *1*(1), 1–20. https://doi.org/10.3390/make1010001
3. Tapeh, A. T. G., & Naser, M. Z. (2023). Artificial Intelligence, Machine Learning, and Deep Learning in Structural Engineering: A Scientometrics Review of Trends and Best Practices. *Archives of Computational Methods in Engineering*, *30*(1), 115–159. https://doi.org/10.1007/s11831-022-09793-w
4. Yüksel, N., Börklü, H. R., Sezer, H. K., & Canyurt, O. E. (2023). Review of artificial intelligence applications in engineering design perspective. *Engineering Applications of Artificial Intelligence*, *118*, 105697. https://doi.org/10.1016/j.engappai.2022.105697
5. Nti, I. K., Adekoya, A. F., Weyori, B. A., & Nyarko-Boateng, O. (2022). Applications of artificial intelligence in engineering and manufacturing: A systematic review. *Journal of Intelligent Manufacturing*, *33*(6), 1581–1601. https://doi.org/10.1016/j.compchemeng.2019.03.034
6. Baghbani, A., Choudhury, T., Costa, S., & Reiner, J. (2022). Application of artificial intelligence in geotechnical engineering: A state-of-the-art review. *Earth-Science Reviews*, *228*, 103991. https://doi.org/10.1016/j.earscirev.2022.103991
7. Lagaros, N. D., & Plevris, V. (2022). Artificial intelligence (AI) applied in civil engineering. *Applied Sciences*, *12*(15), 7595. https://doi.org/10.3390/app12157595

8. Goswami, L., Deka, M. K., & Roy, M. (2023). Artificial Intelligence in Material Engineering: A Review on Applications of Artificial Intelligence in Material Engineering. *Advanced Engineering Materials*, *25*(13), 2300104. https://doi.org/10.1002/adem.202300104
9. Schweidtmann, A. M., Esche, E., Fischer, A., Kloft, M., Repke, J., Sager, S., & Mitsos, A. (2021). Machine Learning in Chemical Engineering: A Perspective. *Chemie Ingenieur Technik*, *93*(12), 2029–2039. https://doi.org/10.1002/cite.202100083
10. Thebelt, A., Wiebe, J., Kronqvist, J., Tsay, C., & Misener, R. (2022). Maximizing information from chemical engineering data sets: Applications to machine learning. *Chemical Engineering Science*, *252*, 117469. https://doi.org/10.1016/j.ces.2022.117469
11. Yu, J., & Zhang, Y. (2023). Challenges and opportunities of deep learning-based process fault detection and diagnosis: A review. *Neural Computing and Applications*, *35*(1), 211–252. https://doi.org/10.1007/s00521-022-08017-3
12. Amini, M., & Chang, S. (2018). A review of machine learning approaches for high dimensional process monitoring. *Proceedings of the 2018 Industrial and Systems Engineering Research Conference*.
13. Hafner, R., & Riedmiller, M. (2011). Reinforcement learning in feedback control: Challenges and benchmarks from technical process control. *Machine Learning*, *84*(1–2), 137–169. https://doi.org/10.1007/s10994-011-5235-x
14. Goldsmith, B. R., Esterhuizen, J., Liu, J.-X., Bartel, C. J., & Sutton, C. (2018). *Machine learning for heterogeneous catalyst design and discovery*. *64*(7), 2311–2323. https://doi.org/10.1002/aic.16198
15. Yılmaz, B., Ceylan, H. B., Yaz, G., & Yıldırım, R. (2025). Performance Assessment of Catalyst Materials for CO2 Hydrogenation to Methanol Using Explainable Machine Learning. *Energy & Fuels*, *39*, 9956–9967. https://doi.org/10.1021/acs.energyfuels.5c00792
16. Tewari, A., Liu, K.-H., & Papageorgiou, D. (2020). Information-theoretic sensor planning for large-scale production surveillance via deep reinforcement learning. *Computers & Chemical Engineering*, *141*, 106988. https://doi.org/10.1016/j.compchemeng.2020.106988
17. Weichert, D., Link, P., Stoll, A., Rüping, S., Ihlenfeldt, S., & Wrobel, S. (2019). A review of machine learning for the optimization of production processes. *The International Journal of Advanced Manufacturing Technology*, *104*(5–8), 1889–1902. https://doi.org/10.1007/s00170-019-03988-5

18. Tulsyan, A., Garvin, C., & Ündey, C. (2018). Advances in industrial biopharmaceutical batch process monitoring: Machine-learning methods for small data problems. *Biotechnology and Bioengineering*, *115*(8), 1915–1924. https://doi.org/10.1002/bit.26605
19. Ning, C., & You, F. (2019). Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming. *Computers & Chemical Engineering*, *125*, 434–448. https://doi.org/10.1016/j.compchemeng.2019.03.034
20. Bhosekar, A., & Ierapetritou, M. (2018). Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Computers & Chemical Engineering*, *108*, 250–267. https://doi.org/10.1016/j.compchemeng.2017.09.017
21. McBride, K., & Sundmacher, K. (2019). Overview of surrogate modeling in chemical process engineering. *Chemie Ingenieur Technik*, *91*(3), 228–239. https://doi.org/10.1002/cite.201800091
22. Mosavi, A. (2010). Optimization and Decision Making in Chemical Engineering Problems. *Global Journal of Researches in Engineering*, *10*, 8–13.
23. Rangaiah, G. P. (2016). *Multi-objective optimization: Techniques and applications in chemical engineering* (Vol. 5). world scientific.
24. Rangaiah, G. P., & Bonilla-Petriciolet, A. (2013). *Multi-objective optimization in chemical engineering: Developments and applications*. John Wiley & Sons.
25. Deb, K., Sindhya, K., & Hakanen, J. (2016). Multi-objective optimization. In *Decision sciences* (pp. 161–200). CRC Press.
26. Gunantara, N. (2018). A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, *5*(1), 1502242. https://doi.org/10.1080/23311916.2018.1502242
27. Triantaphyllou, E. (2000). Multi-Criteria Decision Making Methods. In E. Triantaphyllou, *Multi-criteria Decision Making Methods: A Comparative Study* (Vol. 44, pp. 5–21). Springer. https://doi.org/10.1007/978-1-4757-3157-6_2
28. Ceballos, B., Lamata, M. T., & Pelta, D. A. (2016). A comparative analysis of multi-criteria decision-making methods. *Progress in Artificial Intelligence*, *5*, 315–322. https://doi.org/10.1007/s13748-016-0093-1
29. Wang, Z., Nabavi, S. R., & Rangaiah, G. P. (2024). Multi-Criteria Decision Making in Chemical and Process Engineering: Methods, Progress, and Potential. *Processes*, *12*(11), 2532. https://doi.org/10.3390/pr12112532

30. Nabavi, S. R., Wang, Z., & Rangaiah, G. P. (2024). Sensitivity Assessment of Multi-Criteria Decision-Making Methods in Chemical Engineering Optimization Applications. In F. Gómez-Castro & V. Rico-Ramírez, *Optimization in chemical engineering: Deterministic, meta-heuristic and data-driven techniques*. De Gruyter.
31. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. https://doi.org/10.1109/4235.996017
32. Thibault, J. (2009). Net flow and rough sets: Two methods for ranking the pareto domain. In *Multi-objective optimization: Techniques and applications in chemical engineering* (pp. 199–246). World Scientific. https://doi.org/10.1142/10240
33. Fettaka, S., Gupta, Y. P., & Thibault, J. (2012). Multiobjective optimization of an industrial styrene reactor using the dual population evolutionary algorithm (DPEA). *International Journal of Chemical Reactor Engineering*, *10*(1). https://doi.org/10.1515/1542-6580.2828
34. Leite, B., da Costa, A. O. S., & da Costa Junior, E. F. (2023). Multi-objective optimization of adiabatic styrene reactors using Generalized Differential Evolution 3 (GDE3). *Chemical Engineering Science*, *265*, 118196. https://doi.org/10.1016/j.ces.2021.116805
35. Leite, B., da Costa, A. O. S., & da Costa Junior, E. F. (2021). Simulation and optimization of axial-flow and radial-flow reactors for dehydrogenation of ethylbenzene into styrene based on a heterogeneous kinetic model. *Chemical Engineering Science*, *244*, 116805. https://doi.org/10.1016/j.ces.2021.116805
36. Tiso, T., Winter, B., Wei, R., Hee, J., de Witt, J., Wierckx, N., Quicker, P., Bornscheuer, U. T., Bardow, A., & Nogales, J. (2022). The metabolic potential of plastics as biotechnological carbon sources–review and targets for the future. *Metabolic Engineering*, *71*, 77–98. https://doi.org/10.1016/j.ymben.2021.12.006
37. Precedence Research. (2025, January 16). *Styrene Market Size, Share, Trends, Growth, Report 2024–2033*. https://www.precedenceresearch.com/styrene-market
38. Sheel, J. G. P., & Crowe, C. M. (1969). Simulation and optimization of an existing ethylbenzene dehydrogenation reactor. *The Canadian Journal of Chemical Engineering*, *47*(2), 183–187. https://doi.org/10.1002/cjce.5450470215

39. Yee, A. K., Ray, A. K., & Rangaiah, G. P. (2003). Multiobjective optimization of an industrial styrene reactor. *Computers & Chemical Engineering*, *27*(1), 111–130. https://doi.org/10.1016/S0098-1354(02)00163-1
40. García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: Methods and prospects. *Big Data Analytics*, *1*(1), 1–22. https://doi.org/10.1186/s41044-016-0014-0
41. Jia, W., Sun, M., Lian, J., & Hou, S. (2022). Feature dimensionality reduction: A review. *Complex & Intelligent Systems*, *8*(3), 2663–2693. https://doi.org/10.1007/s40747-021-00637-x
42. Plevris, V., Solorzano, G., Bakas, N. P., & Ben Seghier, M. E. A. (2022). Investigation of performance metrics in regression analysis and machine learning-based prediction models. *8th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2022)*. https://doi.org/10.23967/eccomas.2022.155
43. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*, 115–133. https://doi.org/10.1007/BF02478259
44. Ojha, V. K., Abraham, A., & Snášel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, *60*, 97–116. https://doi.org/10.1016/j.engappai.2017.01.013
45. Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, *11*(2), 431–441. https://doi.org/10.1137/0111030
46. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv Preprint arXiv:1412.6980*. https://doi.org/10.48550/arXiv.1412.6980
47. Awad, M., & Khanna, R. (2015). Support Vector Regression. In M. Awad & R. Khanna, *Efficient Learning Machines* (pp. 67–80). Apress. https://doi.org/10.1007/978-1-4302-5990-9_4
48. Drucker, H., Burges, C. J., Kaufman, L., Smola, A., & Vapnik, V. (1996). Support vector regression machines. *Advances in Neural Information Processing Systems*, *9*, 155–161.
49. Zhang, F., & O'Donnell, L. J. (2020). Support vector regression. In *Machine learning* (pp. 123–140). Academic Press. https://doi.org/10.1016/B978-0-12-815739-8.00007-9

50. Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*, 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88
51. Gibbs, M. N. (1998). *Bayesian Gaussian processes for regression and classification* [PhD Thesis, University of Cambridge].
52. Williams, C., & Rasmussen, C. (1995). Gaussian processes for regression. *Advances in Neural Information Processing Systems*, *8*.
53. Schulz, E., Speekenbrink, M., & Krause, A. (2018). A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, *85*, 1–16. https://doi.org/10.1016/j.jmp.2018.03.001
54. Deringer, V. L., Bartók, A. P., Bernstein, N., Wilkins, D. M., Ceriotti, M., & Csányi, G. (2021). Gaussian Process Regression for Materials and Molecules. *Chemical Reviews*, *121*(16), 10073–10141. https://doi.org/10.1021/acs.chemrev.1c00022
55. Gunnell, L. L., Manwaring, K., Lu, X., Reynolds, J., Vienna, J., & Hedengren, J. (2022). Machine Learning with Gradient-Based Optimization of Nuclear Waste Vitrification with Uncertainties and Constraints. *Process.*, *10*(11). https://doi.org/10.3390/pr10112365
56. Ortiz-Villaseñor, D., Trujillo-Hernández, G., Real-Moreno, O., Castro-Toscano, M. J., Medina-Madrazo, L. D., & Barrera-Román, D. (2025). K-Nearest Neighbors Regression and Applications. In *Exploring Psychology, Social Innovation and Advanced Applications of Machine Learning* (pp. 295–316). IGI Global Scientific Publishing.
57. Halder, R. K., Uddin, M. N., Uddin, Md. A., Aryal, S., & Khraisat, A. (2024). Enhancing K-nearest neighbor algorithm: A comprehensive review and performance analysis of modifications. *Journal of Big Data*, *11*(1), 113. https://doi.org/10.1186/s40537-024-00973-y
58. Wekalao, J., & Mandela, N. (2024). Graphene metasurface-based biosensor for direct dopamine detection utilizing surface Plasmon resonance in the terahertz regime with machine learning optimization via K-nearest neighbors regression. *Plasmonics*, 1–29. https://doi.org/10.1007/s11468-024-02570-4
59. Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees* (First). Chapman and Hall/CRC.
60. Hastie, T., Friedman, J., & Tibshirani, R. (2001). *The Elements of Statistical Learning*. Springer. https://doi.org/10.1007/978-0-387-21606-5

61. Rathore, S. S., & Kumar, S. (2016). A Decision Tree Regression based Approach for the Number of Software Faults Prediction. *ACM SIGSOFT Software Engineering Notes*, *41*(1), 1–6. https://doi.org/10.1145/2853073.2853083
62. Xu, M., Watanachaturaporn, P., Varshney, P. K., & Arora, M. K. (2005). Decision tree regression for soft classification of remote sensing data. *Remote Sensing of Environment*, *97*(3), 322–336. https://doi.org/10.1016/j.rse.2005.05.008
63. Balcan, M.-F., & Sharma, D. (2024). Learning accurate and interpretable decision trees. *arXiv Preprint arXiv:2405.15911*. https://doi.org/10.48550/arXiv.2405.15911
64. Mienye, I. D., & Jere, N. (2024). A survey of decision trees: Concepts, algorithms, and applications. *IEEE Access*, *12*, 86716–86727. https://doi.org/10.1109/ACCESS.2024.3416838
65. Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, *24*(6), 2350–2383. https://doi.org/10.1214/aos/1032181158
66. Bakır, R., Orak, C., & Yüksel, A. (2024). Optimizing hydrogen evolution prediction: A unified approach using random forests, lightGBM, and Bagging Regressor ensemble model. *International Journal of Hydrogen Energy*, *67*, 101–110. https://doi.org/10.1016/j.ijhydene.2024.04.173.
67. Sutton, C. D. (2005). Classification and regression trees, bagging, and boosting. *Handbook of Statistics*, *24*, 303–329. https://doi.org/10.1016/S0169-7161(04)24011-1
68. Prasad, A. M., Iverson, L. R., & Liaw, A. (2006). Newer Classification and Regression Tree Techniques: Bagging and Random Forests for Ecological Prediction. *Ecosystems*, *9*(2), 181–199. https://doi.org/10.1007/s10021-005-0054-1
69. Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random Forests. In C. Zhang & Y. Ma (Eds.), *Ensemble Machine Learning* (pp. 157–175). Springer. https://doi.org/10.1007/978-1-4419-9326-7_5
70. Hastie, T., Tibshirani, R., & Friedman, J. (2009). Random Forests. In T. Hastie, R. Tibshirani, & J. Friedman, *The Elements of Statistical Learning* (pp. 587–604). Springer. https://doi.org/10.1007/978-0-387-84858-7_15
71. Segal, M. R. (2004). Machine learning benchmarks and random forest regression. *UCSF: Center for Bioinformatics and Molecular Biostatistics*.

72. Genuer, R., & Poggi, J.-M. (2020). Random Forests. In R. Genuer & J.-M. Poggi, *Random Forests with R* (pp. 33–55). Springer International Publishing. https://doi.org/10.1007/978-3-030-56485-8_3
73. Ahmad, M. W., Reynolds, J., & Rezgui, Y. (2018). Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. *Journal of Cleaner Production*, *203*, 810–821. https://doi.org/10.1016/j.jclepro.2018.08.207
74. John, V., Liu, Z., Guo, C., Mita, S., & Kidono, K. (2016). Real-Time Lane Estimation Using Deep Features and Extra Trees Regression. In T. Bräunl, B. McCane, M. Rivera, & X. Yu (Eds.), *Image and Video Technology* (Vol. 9431, pp. 721–733). Springer. https://doi.org/10.1007/978-3-319-29451-3_57
75. Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1
76. Geurts, P., & Louppe, G. (2011). Learning to rank with extremely randomized trees. *Proceedings of the Learning to Rank Challenge*, 49–61. http://proceedings.mlr.press/v14/geurts11a.html
77. Louppe, G., & Geurts, P. (2012). Ensembles on Random Patches. In P. A. Flach, T. De Bie, & N. Cristianini (Eds.), *Machine Learning and Knowledge Discovery in Databases* (Vol. 7523, pp. 346–361). Springer. https://doi.org/10.1007/978-3-642-33460-3_28
78. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785
79. Huber, F., Yushchenko, A., Stratmann, B., & Steinhage, V. (2022). Extreme Gradient Boosting for yield estimation compared with Deep Learning approaches. *Computers and Electronics in Agriculture*, *202*, 107346. https://doi.org/10.1016/j.compag.2022.107346
80. Peng, J., Sun, W., Zhou, G., Xie, L., Han, H., & Xiao, Y. (2022). The accurate prediction and analysis of bed expansion characteristics in liquid–solid fluidized bed based on machine learning methods. *Chemical Engineering Science*, *260*, 117841. https://doi.org/10.1016/j.ces.2022.117841

81. Sakheta, A., Raj, T., Nayak, R., O'Hara, I., & Ramirez, J. A. (2025). Development and assessment of hybrid machine learning model of biomass pyrolysis process. *Chemical Engineering Science*, *310*, 121552. https://doi.org/10.1016/j.ces.2025.121552
82. Yu, T., & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. *arXiv Preprint arXiv:2003.05689*. https://doi.org/https://doi.org/10.48550/arXiv.2003.05689
83. Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A., Deng, D., & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery*, *13*(2), e1484. https://doi.org/10.1002/widm.1484
84. Andonie, R. (2019). Hyperparameter optimization in learning systems. *Journal of Membrane Computing*, *1*(4), 279–291. https://doi.org/10.1007/s41965-019-00023-0
85. Rangaiah, G. P., Feng, Z., & Hoadley, A. F. (2020). Multi-objective optimization applications in chemical process engineering: Tutorial and review. *Processes*, *8*(5), 508. https://doi.org/10.3390/pr8050508
86. Miettinen, K. (1999). *Nonlinear multiobjective optimization* (Vol. 12). Springer.
87. Deb, K. (2011). Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction. In L. Wang, A. H. C. Ng, & K. Deb (Eds.), *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing* (pp. 3–34). Springer. https://doi.org/10.1007/978-0-85729-652-8_1
88. Brans, J.-P., Vincke, P., & Mareschal, B. (1986). How to select and how to rank projects: The PROMETHEE method. *European Journal of Operational Research*, *24*(2), 228–238. https://doi.org/10.1016/0377-2217(86)90044-5
89. Mareschal, B., Brans, J. P., & Vincke, P. (1984). *PROMETHEE: A new family of outranking methods in multicriteria analysis*. ULB–Université Libre de Bruxelles.

Chapter 4

Autoassociative Neural Network for Missing Data Imputation: A Case Study via the Styrene Production Process

Farough Agin, Jules Thibault, and Clémence Fauteux-Lefebvre*
Department of Chemical and Biological Engineering
University of Ottawa, Ottawa, Ontario, K1N 6N5, Canada
*Corresponding Author: Clémence Fauteux-Lefebvre

This chapter is a manuscript published in The Canadian Journal of Chemical Engineering (2025, 103 (1), 339-358).

4.1 Abstract

A neural network-based model is proposed to estimate missing values of incomplete datasets to augment their size. An autoassociative neural network (AANN), for which the output vector is identical to the input vector, was built for a styrene production process dataset. The proposed model was used to investigate the ability of an AANN to estimate one to three missing variables, evaluating the impact of the size of the datasets used and the level of correlation of the missing values with other process variables. Results show that the proposed AANN model can predict the process data even when the number of records used is relatively small. Moreover, the AANN method is suitable for estimating missing variables with an accuracy that depends on the correlation coefficient of the missing values with other process variables, keeping acceptable estimation for weakly-correlated variables. Moreover, the model was tested on noisy data, and it is shown that the model trained on noisy data can also predict missing values in an acceptable estimation range.

Keywords: Missing value, Autoassociative neural network, Styrene production process, Process data, Data imputation, Noisy data

4.2 Introduction

Missing data is referred to the absence of observations or values, a phenomenon that is common in a variety of fields [1]. Missing values are frequently ascribed to human mistakes while recording and handling data, machine error as a result of faulty machinery, combining unrelated data, etc. [1,2]. The challenge of missing data is pervasive in all spheres of human activities including chemical processes, traffic monitoring, speech recognition, surveys, medical research, business analysis, computer networks, data reconciliation, and industrial processes [3–10]. The prevalence of missing values across all data-related fields leads to a variety of challenges, including performance degradation, issues with data processing, and biased results brought on by disparities between missing and complete values [1].

In chemical processes, ML algorithms are powerful and robust techniques for estimating underlying non-linear relationships between process variables and require a sufficiently large representative databank to achieve the desired performance. Unlike other domains such as finance, recognition, and speech, chemical engineering is not classified as a big data field [11]. Therefore, the size of any dataset for ML in chemical engineering is of great importance. However, process data may be missing due to several reasons such as sensor failure, power blackout, malfunction in the computer system of recording data, and incomplete data entry to name a few [3,12–14]. Moreover, when data are gathered from various literature sources and reported by individuals, not all needed process variables are necessarily available or complete in each of the sources. Therefore, when developing a model from reported data, a frequent significant challenge is to address the lack of uniformity in data obtained from different sources. Some variables and/or data must then be ignored to build a uniform usable dataset. Some of the ignored variables could have a significant importance on the output prediction, potentially leading to a loss of information. For example, Boucheikhchoukh et al.[14] used ML to propose catalyst formulations in order to improve the conversion of SO_2 to SO_3. Experimental data were collected from 106 articles/theses, but because of missing data, only 31 of them, published between 1929 and 2019, were used. As a result, considering the lack of complete data in some sources, techniques to estimate missing values could be considered to replace these missing values and therefore, augment the size of the databank.

Various techniques for handling missing data have been investigated [15–19]. Removing data records containing missing values is a frequent default option in statistical methodologies. Indeed,

removing all variables with missing values is the easiest way of acquiring a complete, albeit reduced database, with an inevitable loss of information [20–22]. However, several imputation procedures to replace missing values with their best estimates have been proposed [23]. The global imputation approach, which is based on the missing attribute, fills in the missing values based on existing values. The statistical average, median, and mode are examples of central tendency measures used in this approach [24,25]. Additionally, procedures using regression [24] or association rule [26] are other global imputation strategies. Local imputation approaches are modifications of global imputation procedures, which impute missing values based on their classes, historical data, or finding the nearest neighbour. Multiple imputation procedures [27] were designed to provide a flexible alternative to similarity approaches (selection based on the distance or belonging to a specific class). Multiple imputation is a three-stage technique that attempts to decrease the uncertainty in imputed data. Although imputation approaches are simple to implement and help to keep the information, they add uncertainties to the imputed values. Some methods such as multiple imputation might be able to tackle the uncertainty, however, integrating the uncertainty of information is challenging. Another disadvantage of such techniques is that they are not suitable for all types of missingness [28,29]. Missing values also can be treated without the necessity for explicit imputation using direct manipulation of missing data approaches, for instance, using decision trees. Decision trees are able to implicitly handle missing data without the need for a separate imputation procedure or pre-defined model structure. They are nonparametric methods that do not make assumptions about data distribution. However, the performance of the algorithm and its ability to estimate missing values can be negatively affected by poor correlation among process variables [30]. Recently, data modelling entailed statistical, probabilistic, and learning approaches to create a model that can describe the data's features in a general way. In the imputation of appropriate values, such models can be employed as an inference process. The maximum likelihood, Bayesian, selection models, pattern-mixture, and neural networks fall in this category [31].

Recently, the application of neural networks in treating missing values due to their underlying flexible structure and generalization ability has been studied by many researchers [32–41]. A neural network with a nonlinear activation function can theoretically approximate complex non-linear relationships [32]. The neural network-based imputation approach can exploit complex underlying relationships within missing information in the data. The neural network imputation approach

typically utilizes all available records to train the network, then prefills partial records into the network and imputes missing values using the network's output [33]. Sharpe and Solly [34] built a multilayer perceptron (MLP) for each missingness pattern, which is utilized to fit the regression relationship between missing attributes and available attributes. Nonetheless, in the case of many missingness patterns, the number of generated models could become excessively large and complex to manage in addition of requiring more computation time. Ankaiah and Ravi [35] introduced an improved MLP imputation approach that constructs a single objective prediction network using each missing feature as output and the remaining variables as inputs. This approach generates the same number of models as the number of missing attributes and usually applies to a single missing value.

The autoassociative neural network (AANN) is a type of neural network in which the input and output layers are identical and have an equal number of nodes. To impute incomplete data for all missingness patterns, only one model is then required [36]. Marwala et al. [37] presented and tested an imputation approach that combined AANN with a genetic algorithm (GA) on two real datasets [38,39]. When the network has been trained with the complete dataset, the cost function of AANN is used as the fitness function of the genetic algorithm to impute missing values. High accuracy was reported for one and two missing variables. Ravi and Krishna [42] proposed four improved imputation models based on AANN: general regression AANN, particle swarm optimization based on AANN, autoassociative wavelet neural network, and radial basis function AANN. Autoassociative extreme learning machine [43] and counter propagation AANN are two other imputation models based on AANN suggested by Gautam and Ravi [44]. The findings of the experiments suggest that combining local learning with global approximation can improve imputation outcomes. These studies have shown the potential of AANN for the accurate prediction of missing values in complex datasets, especially in chemical processes which typically deal with a large number of variables [3,45–47]. However, in chemical processes, the size of a complete dataset, the correlation between process variables, and the number of missing variables in each data point play a vital role in the treatment of missing process values.

In this study, we propose a methodology using AANN with MLP along with the global imputation technique, to augment the size of a chemical process dataset with missing data. To test the ability of AANN for the imputation of missing values for a process with a large number of

variables, the production of styrene, with 22 process variables, is considered. Two major aspects are investigated: 1) the impact of the size of the dataset on the prediction ability of AANN models for missing values, and 2) the effect of one, two and three missing variables on their accurate estimation and their impact on the prediction of other non-missing variables that can be used for data reconciliation. In the end, the ability of AANN to predict missing values in a noisy dataset has been tested and compared with the AANN models obtained by clean and simulated data.

4.3 Methodology

4.3.1 Autoassociative neural network (AANN)

AANNs have been used for many applications: nonlinear principal component analysis to reduce the problem dimensionality [48,49], data reconciliation [50], signal analysis [51] and fault diagnostic and reconstruction [46]. In this investigation, an AANN is used to estimate missing values in order to have a dataset that is as complete as possible. An AANN is a feedforward neural network with an input layer, a number of hidden layers, and an output layer [52]. Figure 4.1 presents the architecture of a simple AANN. The input and the output vectors of the AANN are identical, meaning that the AANN is attempting to reconstruct an exact copy, albeit filtered, of the input vector. During training, the interrelationships between the process variables are encapsulated in the neural network connection weights. An AANN typically uses at least three hidden layers: a mapping layer, a bottleneck layer and a de-mapping layer. For more complex processes, an AANN may contain additional hidden layers. The nodes of the mapping and de-mapping layers must have nonlinear transfer functions such as sigmoid functions. The bottleneck layer has the lowest dimensionality amongst all hidden layers and produces internal encoding and compression of the inputs, which serves to removing all the unnecessary existing data correlations. The transfer function of the nodes of the bottleneck layer can be linear or nonlinear without impacting the network's generality. When used for data imputation, the network is trained by considering only complete data records and then the AANN is used to predict the inputs of missing values of incomplete records.

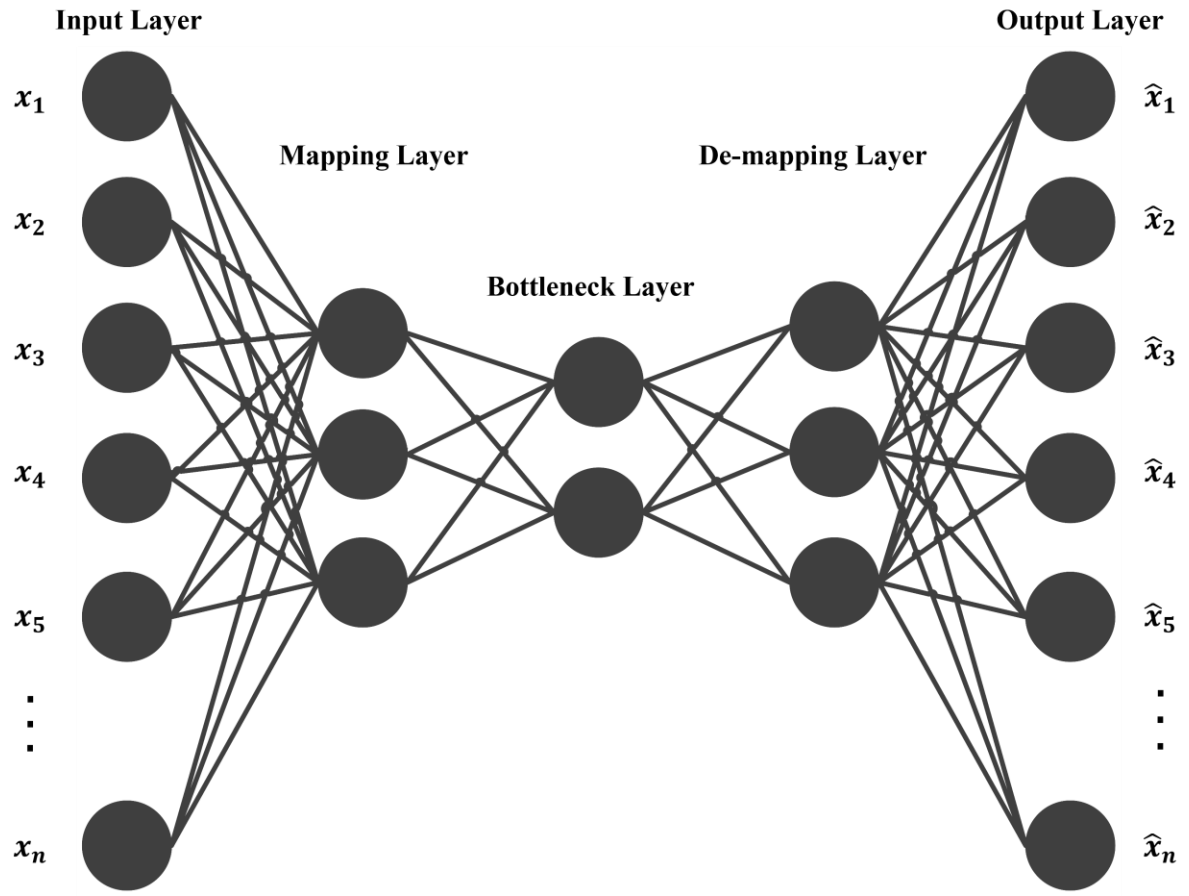


Figure 4.1. Simple example of an AANN model.

The number of neurons in the input and output layers is equal to the number of variables in the process being considered. The number of neurons in the hidden layers depends on the complexity of the problem and must be obtained by trial and error or hyperparameter tuning techniques. If the number of neurons is too low, the representational capacity of the network might be impaired whereas an excessive number of neurons may lead to overfitting. To reconstruct the input variable vector, the process information is encapsulated in the network architecture and the weights of the neuron connections. The training process adjusts the neural network weights such that the reconstructed measurement vector at the output layer represents the input as closely as possible in a least-squares sense. This ensures that the network's internal representation keeps as much information from the original dataset as possible for the given degree of dimensional compression indicated by the bottleneck layer [52,53]. This reconstruction process also serves to filter out some of the inherent noise associated to the process variables.

4.3.2 Styrene production and dataset generation

To evaluate the performance of AANNs for the imputation of missing values, a styrene production process was considered. Styrene is manufactured by dehydrogenating ethylbenzene in the presence of steam over an iron oxide-based catalyst [54,55]. Superheated steam which is a substantial operational cost element [56] is utilized to provide heat for this endothermic reaction, to prevent coke formation, and to lower the partial pressure of the reaction mixture, thereby increasing the equilibrium conversion to favour reaction products [57]. Figure 4.2 depicts a flowsheet diagram of the styrene production process.

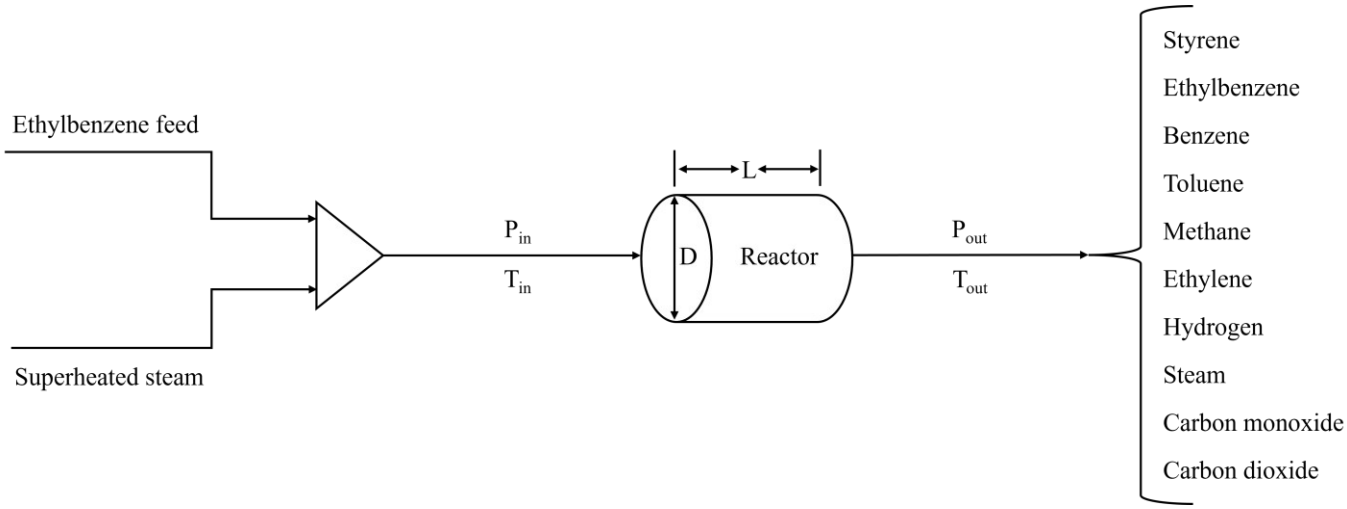


Figure 4.2. Styrene production process diagram.

In the styrene production process, there is one desired reaction (Equation (4.1)) and five secondary and undesired reactions (Equations (4.2)-(4.6)) as follows [58,59]:

$$C_6H_5CH_2CH_3 \rightleftharpoons C_6H_5CHCH_2 + H_2 \quad (4.1)$$

$$C_6H_5CH_2CH_3 \rightarrow C_6H_6 + C_2H_4 \quad (4.2)$$

$$C_6H_5CH_2CH_3 + H_2 \rightarrow C_6H_5CH_3 + CH_4 \quad (4.3)$$

$$2H_2O + C_2H_4 \rightarrow 2CO + 4H_2 \quad (4.4)$$

$$H_2O + CH_4 \rightarrow CO + 3H_2 \quad (4.5)$$

$$H_2O + CO \rightarrow CO_2 + H_2 \quad (4.6)$$

Under typical operating conditions, in which the principal reaction usually achieves equilibrium, the conversion of ethylbenzene can reach roughly 80%. Based on the Le Chatelier's principle, low pressure and high temperature favour the forward dehydrogenation of ethylbenzene to styrene

(Equation (4.1)) because it is an endothermic reversible reaction that yields two moles of product from one mole of reactant. Ethylbenzene is also converted to benzene and toluene by the two main parallel reactions (Equations (4.2) and (4.3)). The operating conditions are then adjusted to favour the conversion of ethylbenzene to styrene and to limit the formation of by-products along with a selective catalyst [60,61].

To generate the dataset used in this investigation, the pseudo-homogeneous kinetic model proposed by Sheel and Crowe [62] was used. They obtained the reaction kinetics and heat of reaction for an industrial adiabatic styrene reactor with Shell 105 catalyst, a promoted iron-oxide catalyst (62% Fe_2O_3, 36% K_2CO_3, and 2% Cr_2O_3). This model was developed using industrial process data with the assumptions that the styrene packed bed behaves as an ideal plug flow reactor and that the kinetic rate expressions account for the heat and mass transfer limitations.

To generate a dataset for the styrene production in an adiabatic reactor, six input variables were varied randomly within pre-defined ranges: the inlet reactor temperature, the inlet pressure, the inlet flow rate of ethylbenzene, the inlet flow rate of steam, and the diameter and the length of the reactor. The reactor model was solved at steady state by finite differences to obtain the flow rates of the ten chemical species in the outlet stream: styrene, ethylbenzene, benzene, toluene, methane, ethylene, hydrogen, steam, carbon monoxide and carbon dioxide, along with the outlet temperature and pressure. The ethylbenzene conversion, as well as the yield and the selectivity of styrene, were calculated knowing the inlet and outlet flow rates of all chemical species. Another variable is the steam to ethylbenzene molar ratio, known as the steam-to-reactant ratio (SOR). The range of SOR is determined based on preventing the formation of coke and minimizing the cost of energy. The obtained dataset therefore contains a total of 22 variables. A total of 30000 data records were generated for this investigation. The list of the 22 elements of each data record is presented in Table 4.1 along with their mean, standard deviation, and minimum and maximum values.

Analyzing the correlation matrix between the above variables and their statistical significance is highly important and is the basis of the decision for predicting missing values. In this work, the Pearson coefficient between two variables above 0.5 or below -0.5 was labeled as “highly-correlated” and the rest was considered as “weakly-correlated”. Figure 4.3 shows the Pearson correlation matrix between all of the variables.

Table 4.1. Description of the generated dataset.

Variable	Min	Max	Mean	Standard Deviation
Steam-to-reactant ratio (SOR*)	8.00	12.00	9.97	1.16
E-Benzene inlet molar flow [mol/s] (FLOW-EB (in))	50.00	80.00	64.78	8.64
Water inlet molar flow [mol/s] (FLOW-H_2O (in))	400.00	960.00	645.96	113.90
Inlet temperature [°C] (T (in))	750.00	950.00	849.25	57.64
Inlet pressure [kPa] (P (in))	150.00	300.00	226.41	42.85
Reactor length [m] (LENGTH)	3.00	9.00	5.80	1.72
Reactor diameter [m] (DCOL)	4.00	10.00	7.07	1.70
Styrene outlet molar flow [mol/s] (FLOW-STY)	0.63	41.89	17.68	6.94
E-Benzene outlet molar flow [mol/s] (FLOW-EB (out))	6.79	76.00	41.35	11.48
Benzene outlet molar flow [mol/s] (FLOW-BZ)	0.01	7.71	1.24	1.38
Toluene outlet molar flow [mol/s] (FLOW-TL)	0.01	50.01	4.51	5.96
Methane outlet molar flow [mol/s] (FLOW-ME)	0.01	3.20	0.86	0.59
Ethylene outlet molar flow [mol/s] (FLOW-ET)	0.00	0.77	0.02	0.05
H_2 outlet molar flow [mol/s] (FLOW-H_2)	2.62	181.67	35.03	24.20
H_2O outlet molar flow [mol/s] (FLOW-H_2O (out))	405.74	1000.10	653.21	113.40
CO outlet molar flow [mol/s] (FLOW-CO)	0.00	0.28	0.04	0.03
CO_2 outlet molar flow [mol/s] (FLOW-CO_2)	0.01	60.45	6.05	7.73
Outlet temperature [°C] (T (out))	713.76	872.74	784.81	35.09
Outlet pressure [kPa] (P (out))	99.98	296.92	199.37	48.56
Conversion of E-Benzene (CONV)	0.04	0.87	0.36	0.15
Yield of Styrene (YIELD)	0.01	0.99	0.80	0.18
Selectivity of Styrene (SELEC)	0.01	91.14	8.11	8.95

* Each variable is represented by a phrase in the parenthesis and used as their symbols in the graphs of the paper.

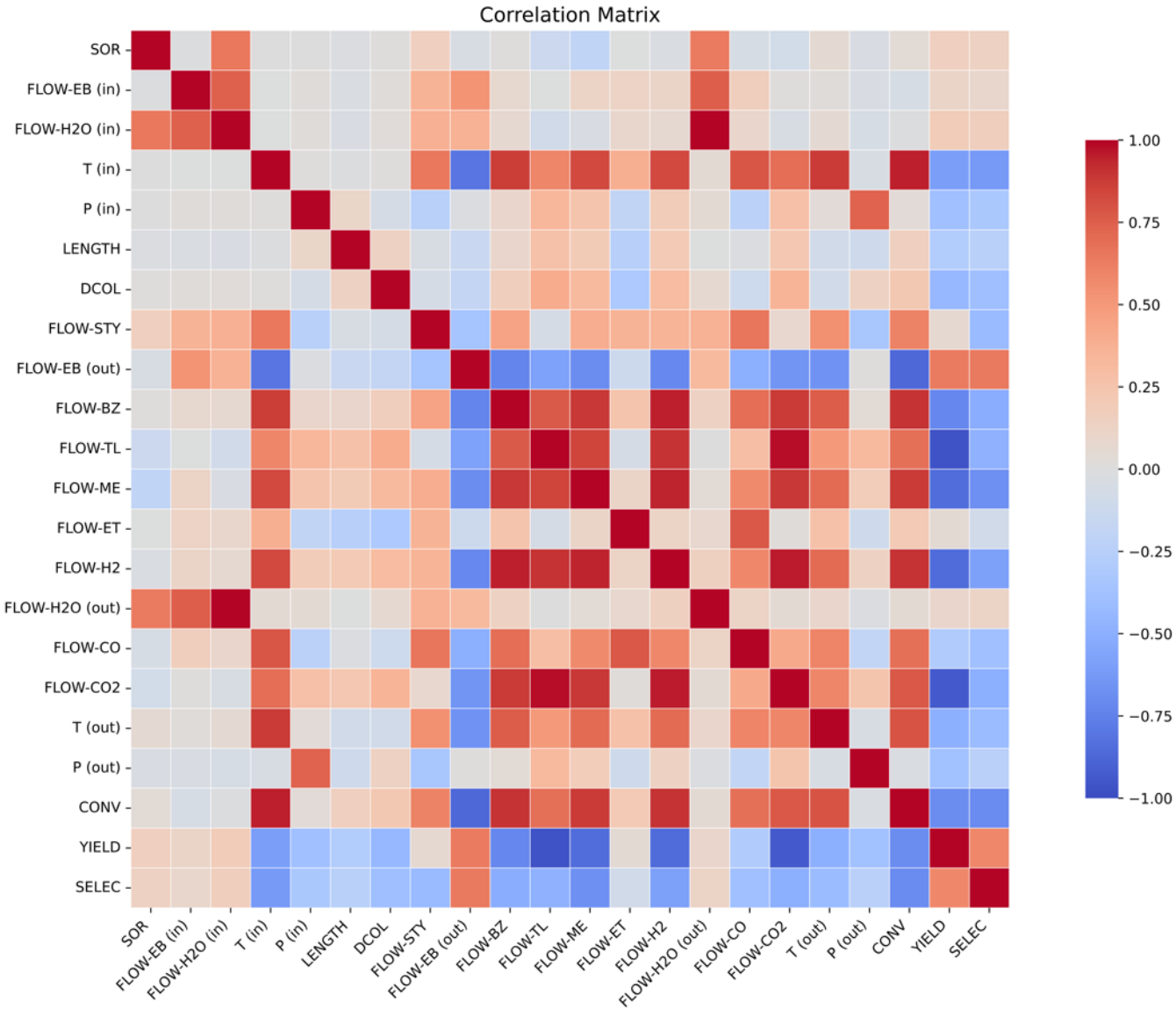


Figure 4.3. Pearson correlation matrix between all of the variables.

4.3.3 Noisy dataset

Real industrial measurements are rarely considered exempt from noise. Recorded data are subject to noise from different sources, such as sensor inherent variability [63,64]. In this regard, the applicability of AANN to predict missing values in a noisy dataset has also been investigated. In this investigation, Gaussian noise ($N(0,\sigma^2)$) was added to the simulated noise-free datasets. Gaussian noise represents the typical measurement normal distribution [65,66].

In this work, the standard deviation of each variable was separately calculated, and 10% of its value was used as the standard deviation of Gaussian noise added to each variable. Therefore,

random noise is generated, and noise is assigned to each variable. Table 4.2 represents the statistical features of the noisy dataset.

Table 4.2. Description of the noisy dataset.

Variable	Min	Max	Mean	Standard Deviation
Steam-to-reactant ratio (SOR)	7.66	12.36	9.97	1.16
E-Benzene inlet molar flow [mol/s] (FLOW-EB (in))	47.67	83.11	64.78	8.68
Water inlet molar flow [mol/s] (FLOW-H_2O (in))	389.12	977.13	645.96	114.5
Inlet temperature [°C] (T (in))	734.76	965.56	849.25	57.87
Inlet pressure [kPa] (P (in))	139.02	314.75	226.41	43.04
Reactor length [m] (LENGTH)	2.55	9.41	5.80	1.73
Reactor diameter [m] (DCOL)	3.41	10.45	7.07	1.7
Styrene outlet molar flow [mol/s] (FLOW-STY)	0.39	42.49	17.68	6.98
E-Benzene outlet molar flow [mol/s] (FLOW-EB (out))	6.31	76.63	41.35	11.53
Benzene outlet molar flow [mol/s] (FLOW-BZ)	0	7.71	1.24	1.38
Toluene outlet molar flow [mol/s] (FLOW-TL)	0	50.2	4.51	5.95
Methane outlet molar flow [mol/s] (FLOW-ME)	0	3.22	0.86	0.59
Ethylene outlet molar flow [mol/s] (FLOW-ET)	0	0.77	0.02	0.05
H_2 outlet molar flow [mol/s] (FLOW-H_2)	0.01	178.53	35.03	24.31
H_2O outlet molar flow [mol/s] (FLOW-H_2O (out))	381.83	992.24	653.21	113.93
CO outlet molar flow [mol/s] (FLOW-CO)	0	0.29	0.04	0.03
CO_2 outlet molar flow [mol/s] (FLOW-CO_2)	0	60.51	6.05	7.7
Outlet temperature [°C] (T (out))	708.58	874.05	784.81	35.26
Outlet pressure [kPa] (P (out))	88.55	305.42	199.37	48.78
Conversion of E-Benzene (CONV)	0.02	0.85	0.36	0.15
Yield of Styrene (YIELD)	0.01	1.04	0.80	0.19
Selectivity of Styrene (SELEC)	0	91.43	8.11	8.96

4.3.4 Data preprocessing

The 22-variable dataset of the styrene manufacturing process is used to evaluate the ability of an AANN to estimate missing values for data imputation. Because the predictive ability of a model depends on the quality and amount of information used to train the AANN, various sizes of the

dataset were used to develop the AANN models, and then each model was used and tested for the imputation of missing values. From the entire dataset consisting of 30000 randomly-generated data records with each record being comprised of 22 process variables, the last 1000 records were extracted and set aside for evaluating the performance of the predictive model for the estimation of missing values. This selection does not add a bias to the results since the entire dataset was generated randomly. These 1000 records were not used in the training, validation, and testing of the neural models. Seven AANNs using 29000, 25000, 15000, 5000, 1000, 500, and 100 data records, respectively, were obtained and used to study the effect of the size of the dataset on the predictive quality of the model. Each dataset for the training of the seven neural networks was partitioned into training, validation, and testing datasets.

As depicted in Table 4.1 and Table 4.2, the values and the ranges of process variables are drastically different. For developing an AANN in this investigation, the values of all variables were then normalized between 0 and 1. Normalization brings the different data features on a similar scale such that all process variables are treated as equally important. Normalization also helps the minimization algorithm to converge more rapidly. In order to be able to compare the effectiveness of the AANN models, the minimum of the mean square error (MSE) was used as a comparison criterion, which is calculated as follows (Equation (4.7)):

$$MSE = \sum_{i}^{N} (y_i - \hat{y}_i)^2 / N \tag{4.7}$$

where N is the number of samples, and y_i and $\hat{y}_i$ are the real and predicted outputs, respectively. The results were derived using the Google Colab platform alongside various libraries, including TensorFlow, Keras, and Scikit-learn.

4.4 Results and discussion

4.4.1 Final AANN architecture

Regarding the development of the AANN, meticulous attention was paid to its structural design and parameter selection. Various models were assessed by adjusting the network parameters to minimize model error and enhance prediction accuracy. Determining the precise architecture of the AANN, including the number of layers and neurons within each layer, involved an iterative approach. The aim was to achieve optimal predictive performance while preventing overfitting

and structure complexity. To this end, AANN structures with three, five, and seven hidden layers, each with differing neuron numbers alongside various optimization algorithms, learning rates, batch sizes, and epoch numbers were examined. The selection of the optimal AANN structure was based on the performance of each model. The parameters utilized in constructing the AANN are detailed in Table 4.3. MSE served as the loss function, employing the Adam algorithm as the backpropagation optimizer with a learning rate set at 0.01. The learning process spanned 2000 epochs, utilizing a batch size of 200 for each iteration. Subsequently, an AANN structure featuring five hidden layers was identified as the optimal model and was thus adopted for further investigation.

Table 4.3. Final neural network parameters.

Number of layers	Number of neurons in each layer	Batch Size	Learning rate	Loss Function	Optimizer	Epochs
7	Input layer: 22 1st hidden layer: 22 2nd hidden layer: 14 3rd hidden layer: 7 4th hidden layer: 14 5th hidden layer: 22 Output layer: 22	200	0.01	MSE	Adam	2000

The architecture of the AANN used in this investigation is presented in Figure 4.4. It consists of an input layer, two mapping layers, a bottleneck layer, two de-mapping layers and an output layer. The input and the output layers have 22 neurons, each neuron corresponding to one of the 22 process variables. Indeed, the intrinsic characteristic of an AANN is for the output to replicate the input layer. The first mapping layer and the second de-mapping layer have 22 neurons, and the second mapping layer and the first de-mapping layer have 14 neurons each. The bottleneck layer, which is the smallest layer in the architecture of an AANN, has 7 neurons. The lower number of neurons in the bottleneck layer is used to obtain a representation of the input with reduced dimensionality to extract salient features and typically leads to faster training, while maintaining comparable classification performance [67].

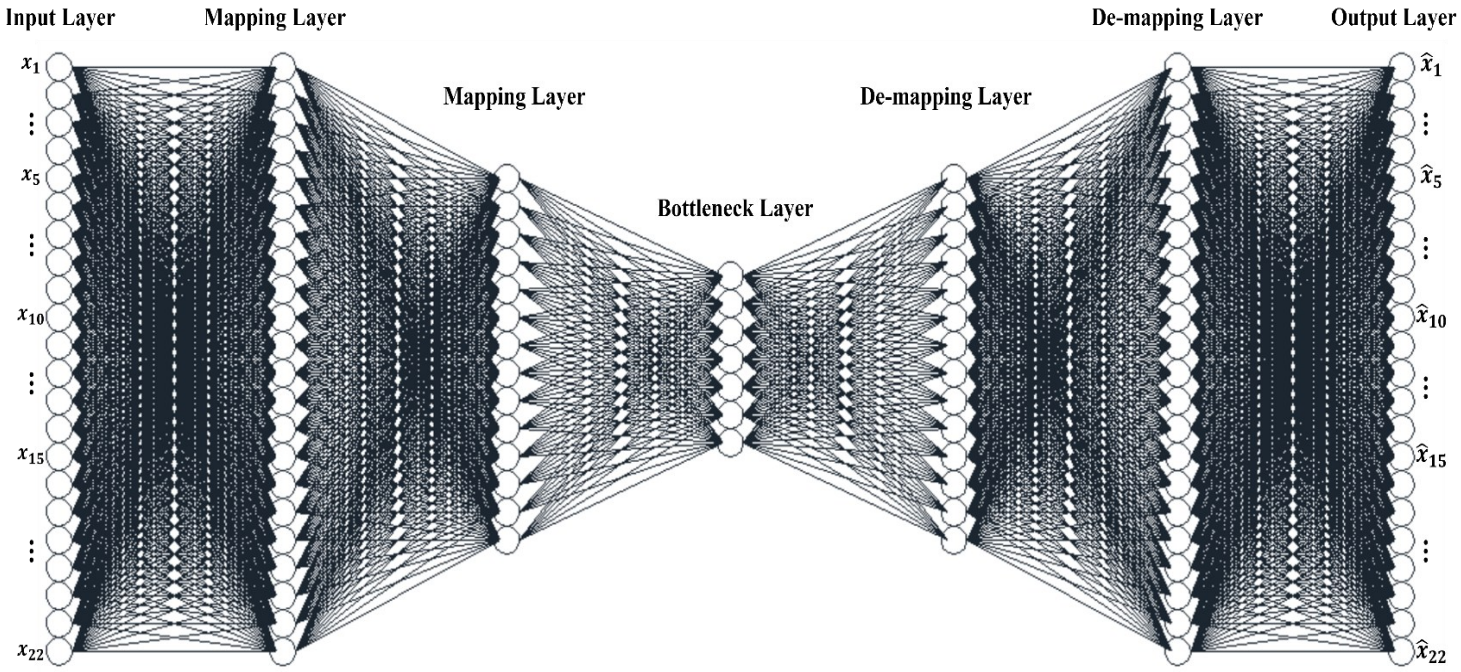


Figure 4.4. AANN architecture used in this investigation.

4.4.2 Performance of AANN models for the seven dataset sizes

With the datasets of various sizes available and the neural network parameters selected, AANN models are developed by determining the values of the weights to minimize the MSE (Equation (4.7)). In order to simulate the existence of missing values in a dataset, the 1000-record evaluation dataset from the 30000-record dataset, set aside and never used for developing the models, was used to observe the prediction error when one, two or three variables are missing. In the building phase of the models, different subsets of the remaining 29000 record were used. Each dataset was split into three subsets: 70% of the data was used to train the models, 15% to validate the models during training, and the remaining 15% to test the models at the end of the learning process.

The first series of tests were conducted to determine the impact of the size of the dataset on the ability of the neural network to reproduce the input vector of the process variables in the absence of missing data. The size varied from 29000 to 100 record datasets. The MSE of models with different sizes of clean datasets are presented in Figure 4.5. The MSE values reported in Figure 4.5 are the averages of the MSE of the 22 process variables in the dataset that were scaled between 0 and 1. Results of Figure 4.5 show that models predict the values of the 22 process variables with MSE around 0.01 and below (lower than 1%) for all sizes of the datasets. The MSE

of training for datasets containing 29000, 25000, and 15000 records were 0.000053, 0.000075, and 0.000064, respectively (0.001% error for training, validation, and test). For the dataset of 5000 records, it was 0.00015 (0.01% error for training, validation, and test). For the datasets of 1000, 500, and 100 records, the MSE for training data was 0.00098, 0.0018, and 0.0022 (error between 0.01% to 1% for training, validation, and test), respectively, showing an exponential increase when reducing the size of the dataset. Nevertheless, even with a dataset of 100 records, the average error hovers around 1% for the test dataset. It has been observed that decreasing the size to 1000 will have little effect on the performance of the model, however, reduces the runtime and storage capacity. Moreover, it can prevent the occurrence of overfitting in complex and biased datasets.

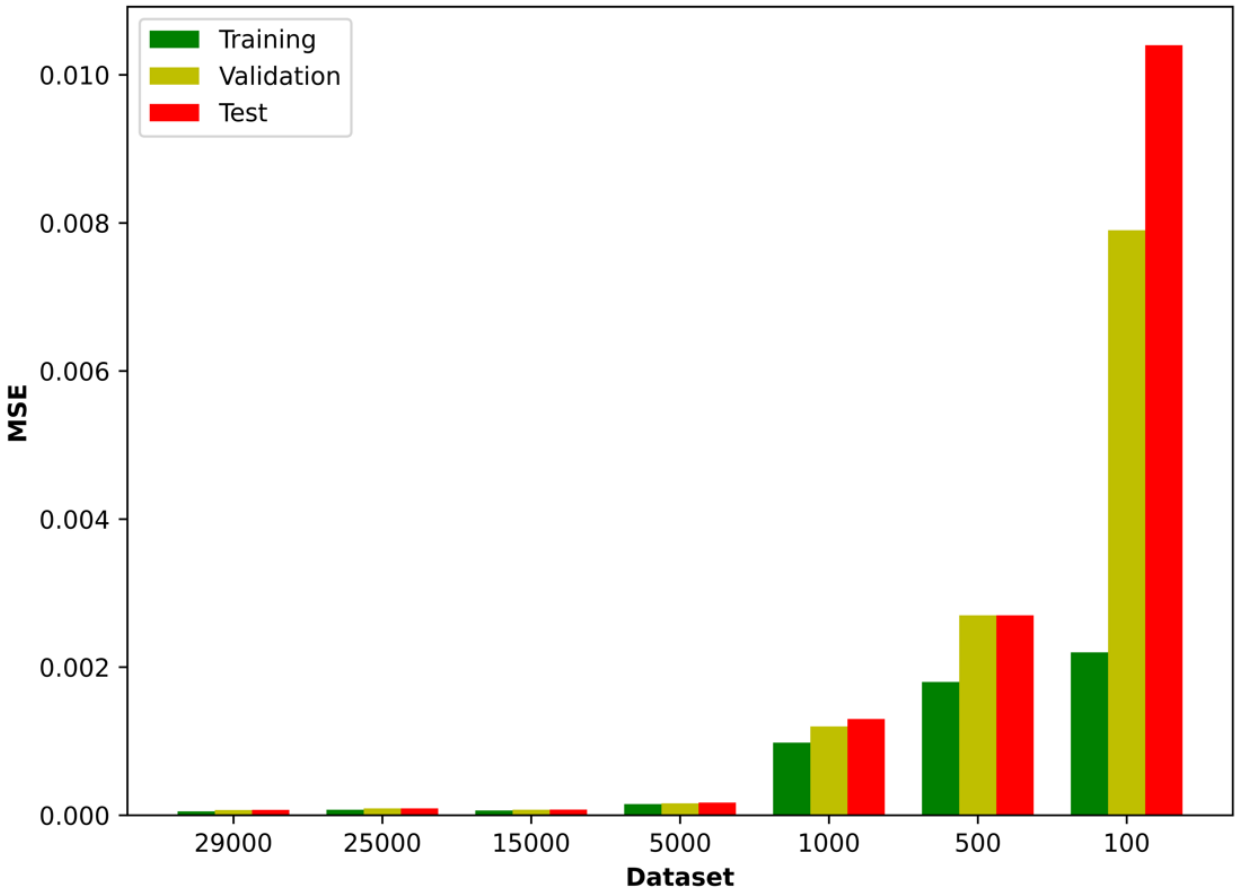


Figure 4.5. MSE values for the training, validation, and test data for seven datasets.

4.4.3 Estimation of one missing value for different sizes of the datasets

After building the AANN models from noise-free dataset of various sizes, each AANN model was used for predicting missing values for the 1000-record evaluation dataset. In addition to the

estimation of the missing values, the impact of the missing variables on the prediction of other non-missing variables was also assessed. When one variable is missing, the values of one column of the 1000-record evaluation dataset were initially replaced by the mean value calculated from the complete dataset. Then, the whole dataset of 1000 records with the new column was imputed to the AANN to predict all input values, including the replaced missing values. Next, the estimated values of the missing variable are fed back into the input vector, and the estimated outputs are calculated once more in order to provide new estimates of the missing variables. This feedback process is repeated for 10 iterations. By performing this iteration procedure, even if the initial estimated missing input was set to zero instead of the mean value, the difference between the final output values of the missing variables and the real values was dramatically decreased and achieved a constant value.

Two process variables were chosen from the clean dataset to illustrate the model performance in predicting missing values based on their correlation with other variables. The correlation between variables was examined by determining the correlation matrix (Figure 4.3) on the complete 30000-record dataset. It is intuitive to think that if a missing variable is strongly correlated with other variables, it is highly probable that the estimated missing value will be closer to the true value. Similarly, if a missing value has a weak correlation with other variables, the estimated value may not be as satisfactory.

The first representative process variable, having a weak correlation with other variables, is the reactor length. Figure 4.6 shows the histogram of the errors of the estimated length value for the models obtained with the seven different sizes of the datasets. The standard deviation and the mean value of the errors are given for each model used. The average errors are centred close to zero, and the distributions are symmetric. In general, the standard deviation of the estimation errors is high and increases when the size of the dataset decreases. This was expected as the length of the packed bed reactor has a very weak correlation with most of the other variables, such that the AANN is not able to infer adequately this missing variable based on the other process variables.

The second representative process variable is the styrene flow rate in the outlet stream of the reactor. This variable is highly correlated with many other variables based on the correlation matrix. Figure 4.7 illustrates the histogram of the estimation errors for the AANN obtained for the models obtained with the seven different sizes of the datasets. The average errors are centred close

to zero and the distribution is symmetric. The standard deviation of the estimation errors is low compared to the reactor length and increases when the size of the dataset decreases, going from 0.005 to 0.064. Similar results were obtained for variables that had a strong correlation coefficient with other process variables.

Figure 4.8 provides some insight into the iterative procedure used to predict missing values of the styrene flow rate from the evaluation dataset with 1000 records. The predictive capabilities of the two models, trained respectively with datasets comprised of 29000 and 100 records, are depicted in Figure 4.8A. Initially, the styrene flow rate column values are replaced by the mean of the entire dataset, represented by the yellow line at 0.41 in Figure 4.8A. Subsequently, employing the AANN models, the missing values are iteratively predicted, with each iteration utilizing the predicted values as inputs for the subsequent iteration. This iterative process is repeated 10 times. Figure 4.8A illustrates a randomly selected sample value from the styrene flow rate column in the 1000-record evaluation dataset, which has an actual value of 0.75. The blue line portrays the predicted value for each iteration by the AANN model developed using the dataset with 29000 records. Notably, the predicted value stabilizes at 0.78 after two iterations, indicating its closeness to the actual value. Conversely, the red line represents the predicted value derived from the AANN model trained on the 100-record dataset. Despite stabilizing after 3 iterations at 0.60, the predicted value deviates from the actual value, albeit closer than the mean value. Notably, rapid convergence towards actual values is observed for the styrene flow rate and the other highly correlated variables, contingent upon the dataset size utilized for AANN training. However, weakly correlated variables exhibited slower convergence rates and lower predictive accuracy, necessitating more iterations for convergence. Figure 4.8B complements this analysis by presenting the MSE values between the predicted and actual styrene flow rate values within the 1000-record evaluation dataset. Initially, the MSE reflects the disparity between the imputed mean value and actual values, yielding 0.028. Subsequent iterations of prediction with AANN models markedly reduce MSE for both AANN models built with the 29000-record (blue line) and 100-record (red line) datasets. The MSE for the 29000-record dataset stabilizes at 0.00046, while for the 100-record dataset, it stabilizes at 0.0045. This discrepancy in MSE underscores the superior performance of the AANN model trained with the larger dataset, indicating its enhanced predictive capabilities.

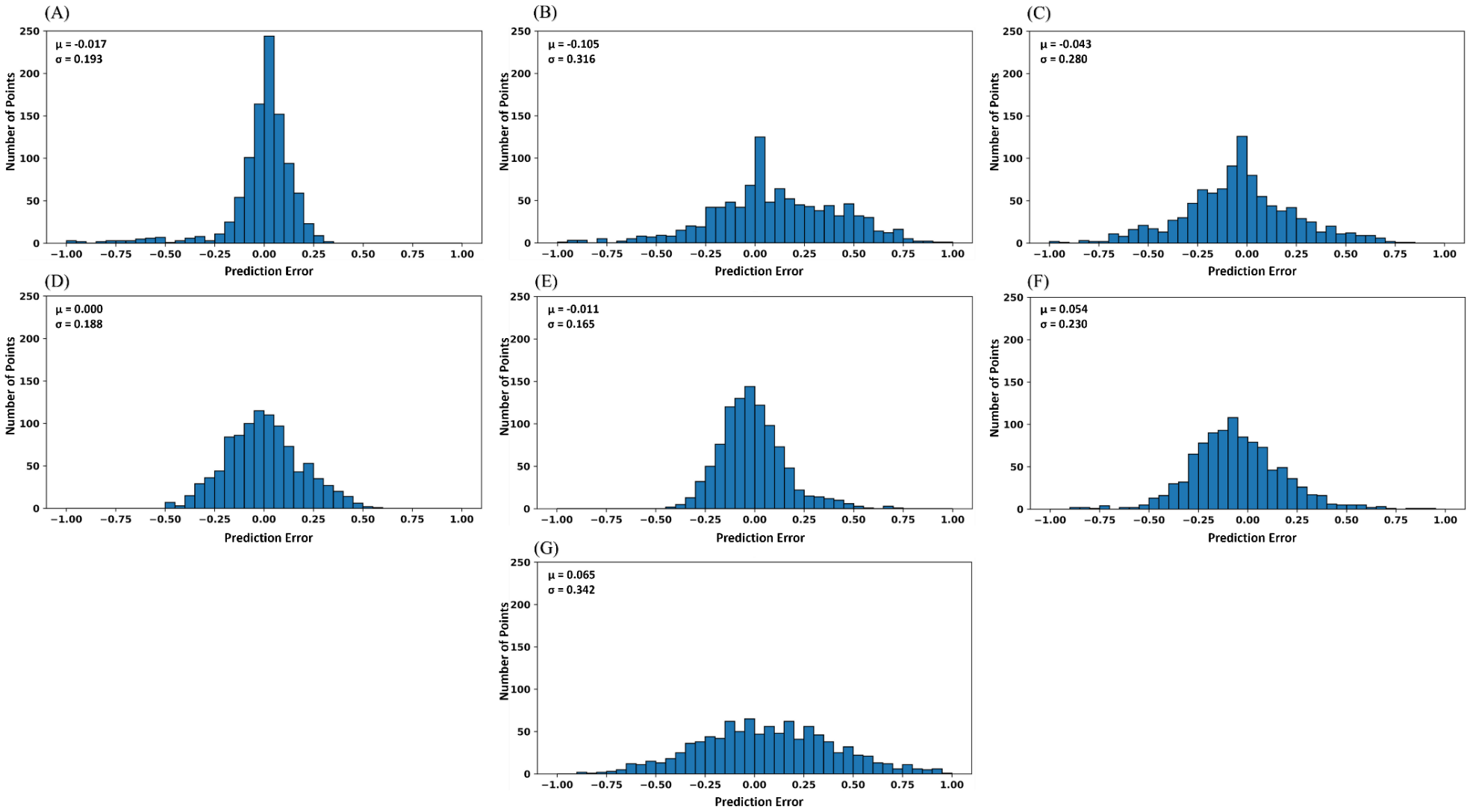


Figure 4.6. Histogram of the estimated errors of the reactor length as one missing process variable for the AANN models obtained with different sizes of the datasets. (A) 29000; (B) 25000; (C) 15000; (D) 5000; (E) 1000; (F) 500; (G) 100**.**

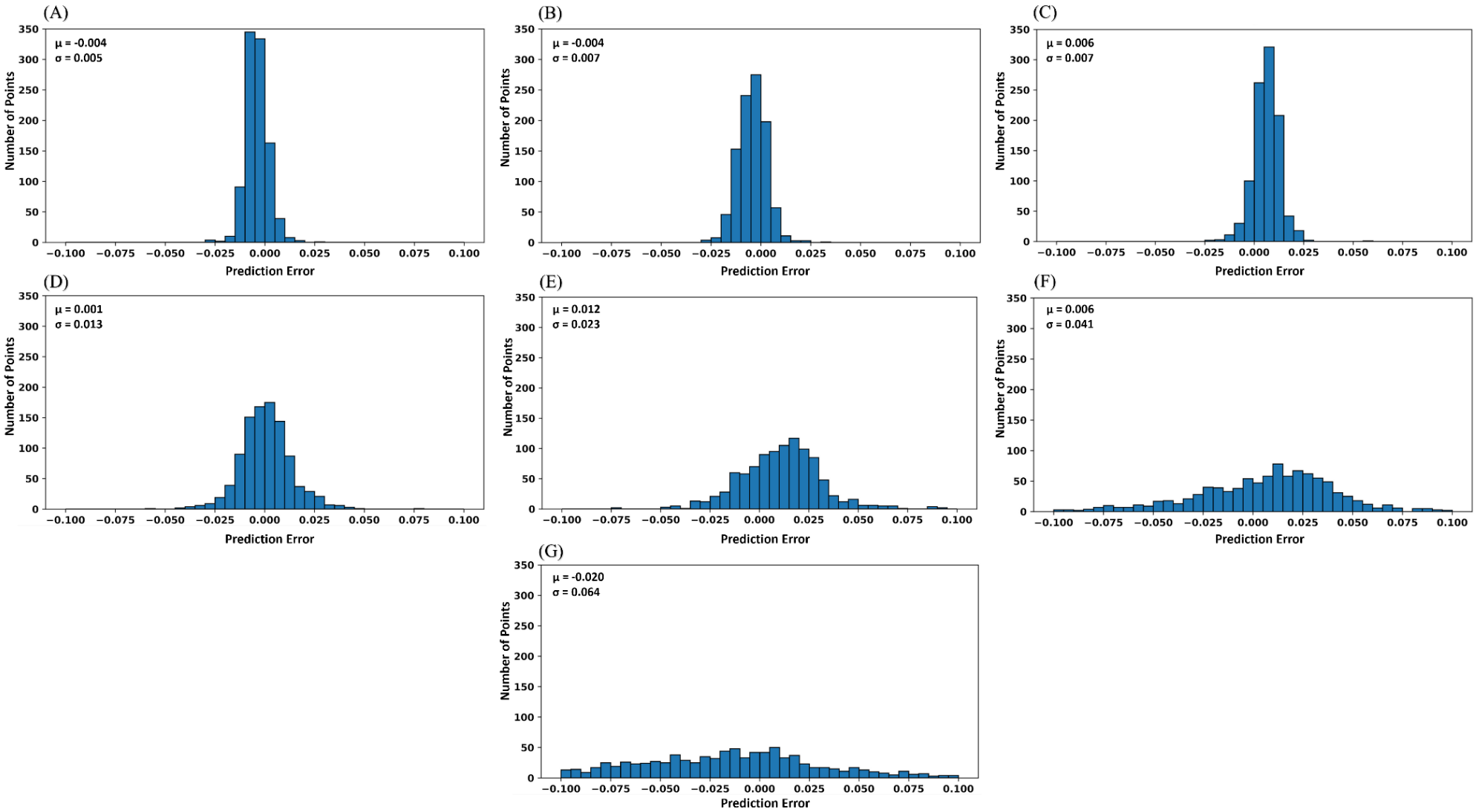


Figure 4.7. Histogram of the estimated errors of the outlet styrene flow rate as one missing process variable for the AANN models obtained with different sizes of the datasets. (A) 29000; (B) 25000; (C) 15000; (D) 5000; (E) 1000; (F) 500; (G) 100.

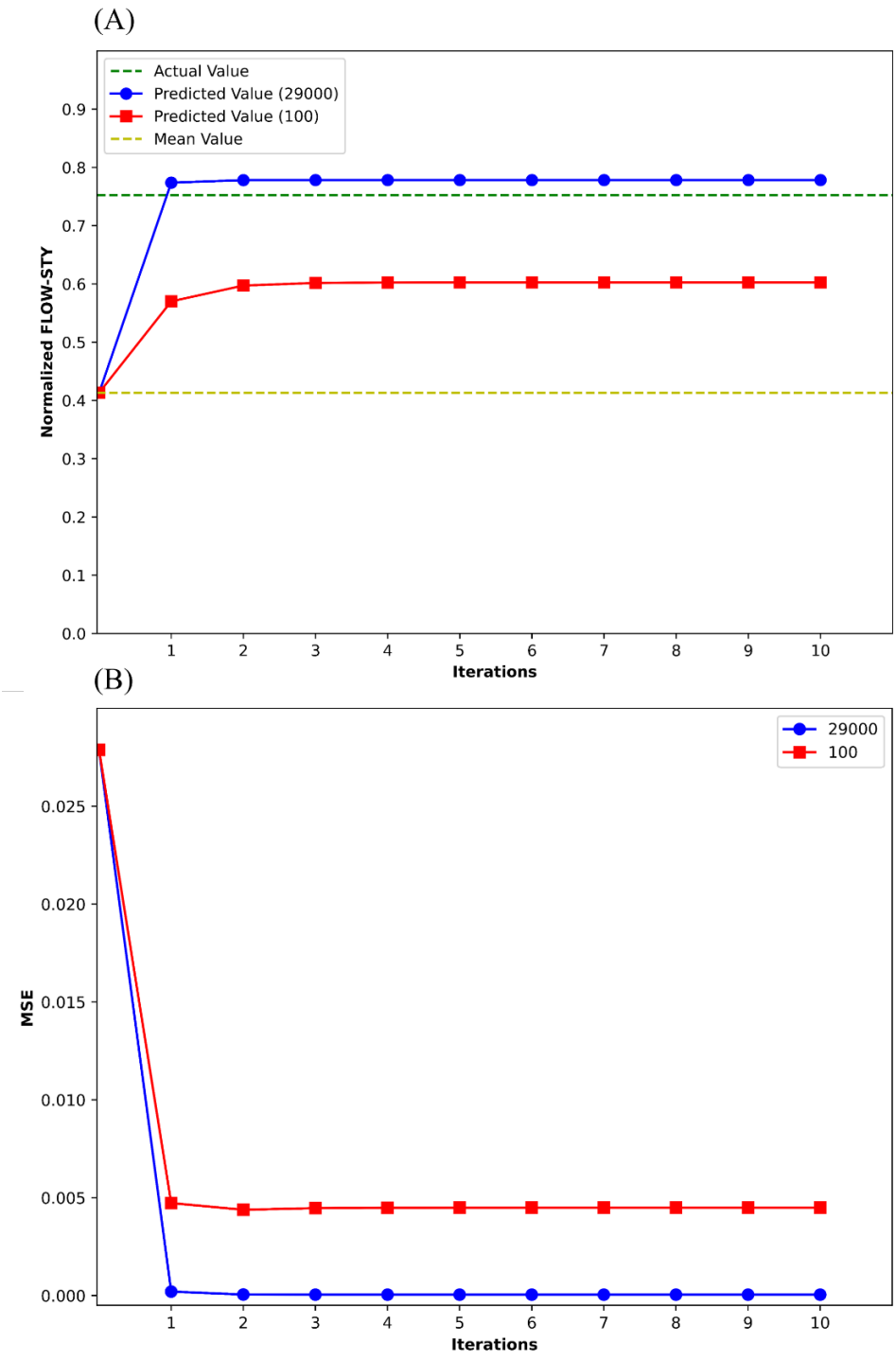


Figure 4.8. (A) Predicted values of a sample of the normalized styrene flow rate obtained from two models trained with 29000 and 100 records; (B) MSE of predicted styrene flow rates of the 1000-record evaluation dataset for two models trained with 29000 and 100 records at each iteration.

The effect of the missing values on the prediction of the other non-missing variables was also investigated. There are two ways a databank can be used to build a representative model of a process. In the first option, the actual unfiltered process data, complemented when possible with the imputation of missing data, are used. The second option is to use the data predicted by the AANN, including the estimation of the missing data. The latter option has the advantage that some noise filtering and data reconciliation are performed via an AANN. When the second option is favoured and one process variable is missing, it is possible that the estimation procedure using the AANN may also affect the estimation of the other process variables. This is not the case for the first option.

The difference between the output values of the AANN models obtained for the complete 1000-record evaluation dataset and the predicted outputs of the AANN models when a missing value has to be estimated was calculated. As there are 22 variables, a table with 22 rows and 22 columns of the average difference or the standard deviation of the differences can be generated by considering that, in turn, each process variable was missing. Figure 4.9 provides the heat map of the standard deviation of the differences between the predictions with and without a missing value for the model that was developed with the 29000-record clean dataset. As the results of Figure 4.9 show, the procedure for the estimation of one missing variable for most variables has a weak impact on the predictions of the other variables. However, for some process variables, which are weakly correlated with other variables, results show a greater variability in the standard deviations of the difference. For weakly-correlated variables, a greater impact was observed on themselves but also on some of the other process variables. For example, the absence of variables such as the length and the diameter of the reactor, the outlet temperature, and the outlet pressure resulted in an error above 5% for their own evaluation. The absence of the length and the diameter of the reactor has affected the prediction of the flow rates of some output chemical species, such as the flow rates of styrene, toluene, and CO_2, as well as the yield of styrene.

The same trends were observed with the model developed using the 100 clean data records, as shown in Figure 4.10. The effect of missing values on the other variables, especially for weakly-correlated variables, is higher than for the highly-correlated variables. However, due to the weaker performance of the built AANN model for the 100-record dataset, some highly-correlated

variables, such as SOR and the inlet flow rate of ethylbenzene, were more impacted as expressed by their standard deviations of the differences.

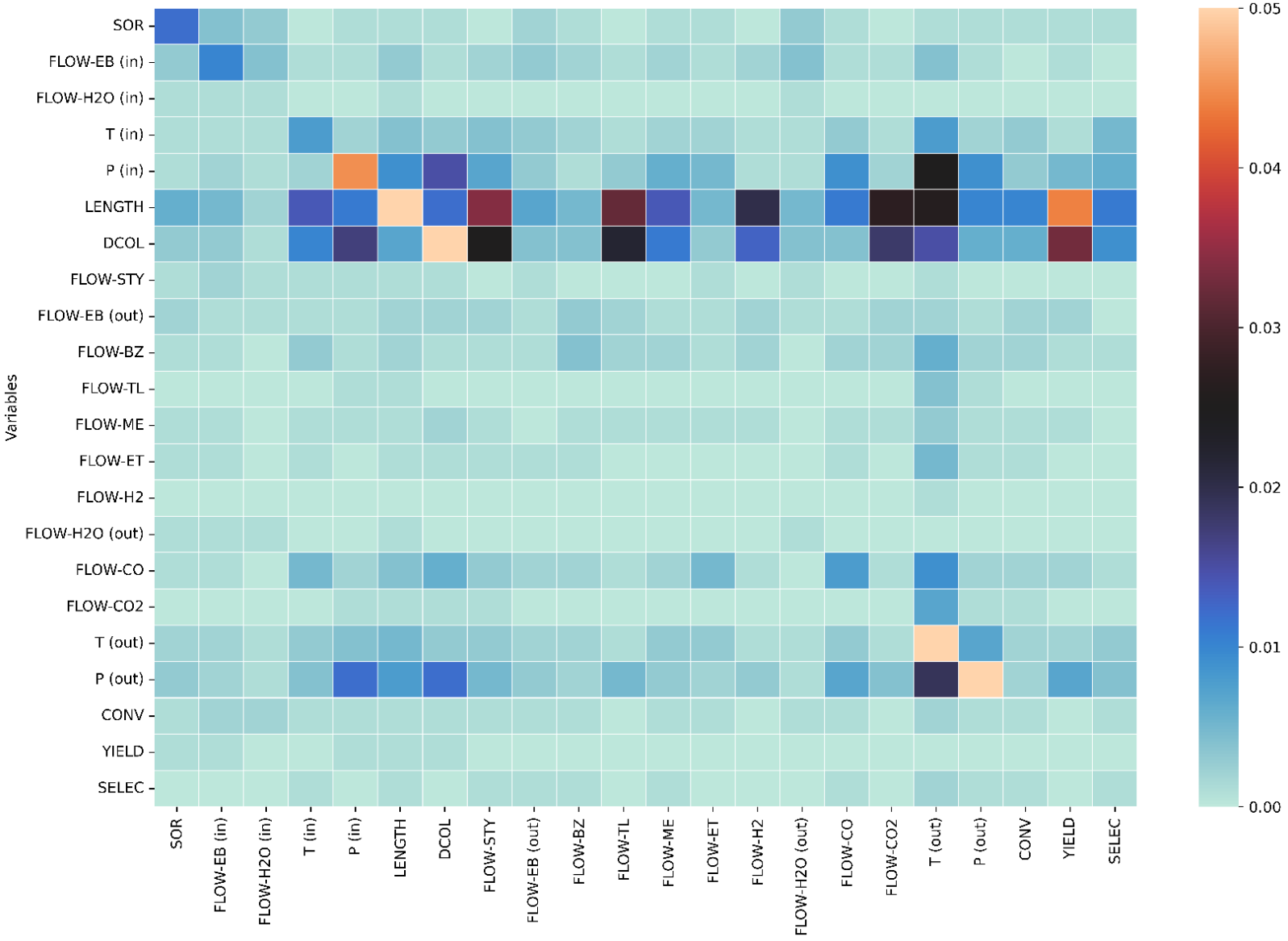


Figure 4.9. Heat map of the standard deviations of the effect of one missing variable on itself and on the other 21 variables for the model developed with the 29000-record dataset.

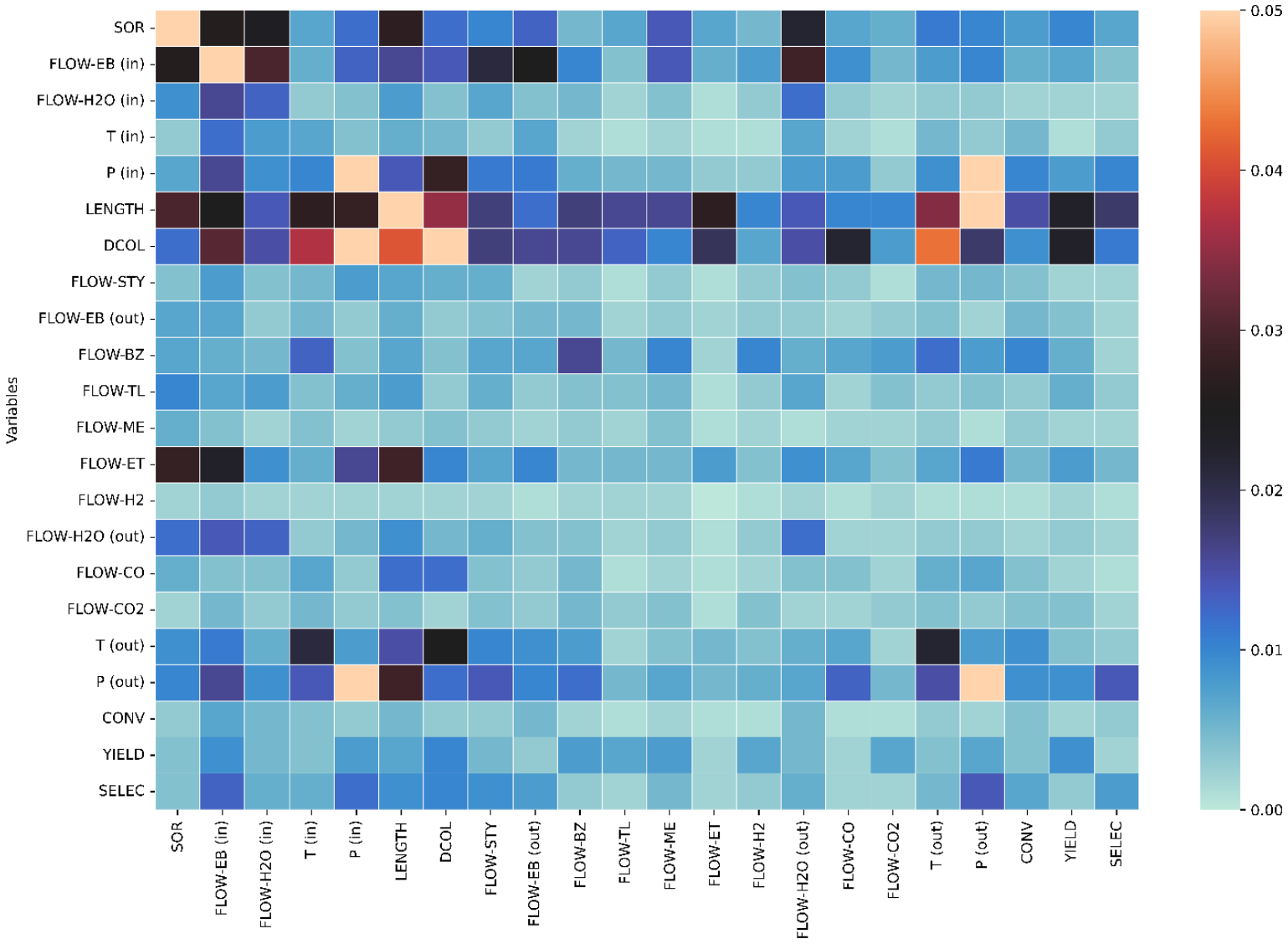


Figure 4.10. Heat map of the standard deviations of the effect of one missing variable on itself and on the other 21 variables for the model developed with the 100-record dataset.

4.4.4 Performance of AANNs for two missing process variables

In the previous section, the performance of AANNs for the estimation of one missing variable has been presented. In this section, the evaluation of the performance of AANNs for the imputation of two missing variables is examined. Since there are 22 variables in the vector of process variables, choosing randomly two variables among these 22 variables and reporting all the results is obviously impossible. Instead, ten pairs of variables were selected, based on the correlation matrix (Figure 4.3). In this analysis, five pairs were chosen with highly-correlated variables, and the other

five pairs were selected with weakly-correlated variables. The procedure for evaluating the performance of AANNs for the estimation of the two missing variables and their impact on the estimation of the other 20 variables is identical to the one described in the previous section. The two selected columns of missing values of the 1000-record evaluation clean dataset were initially replaced with their mean values and the estimations of the two missing values by the AANN models were fed back to the input vector for a total of 10 iterations.

The average values of the mean and the standard deviation of the differences between the estimated outputs obtained for the 1000-record evaluation clean dataset using the AANN models developed using different sizes of datasets were calculated with and without the missing values for all selected pairs of two missing variables. The rationale for choosing highly-correlated variables is to assess the performance of the AANN when fewer variables, which could mutually assist in estimating the missing variables, are missing from the dataset. For instance, the inlet molar flow rate of ethylbenzene and the conversion of ethylbenzene are two process variables that are highly correlated with each other. For this pair of variables, Figure 4.11A presents the plots of the mean and the standard deviation of the differences, based on the 1000-record evaluation dataset for both variables and for all AANNs developed using different sizes of the datasets. Results show that the standard deviation increases when the size of datasets is reduced from 29000 to 100, from 0.008 up to 0.072. The mean values are located close to zero with slightly lower means with AANNs associated with larger datasets. On the other hand, the trend is different for weakly-correlated variables. Figure 4.11B presents the mean and the standard deviation of a second set of two missing variables with a mutually weak correlation. These two variables are the length of the reactor and the outlet flow rate of water. For the length of the reactor, which is weakly correlated with most process variables, its mean value shows a greater variation from the expected nil value, and the standard deviations of the prediction differences are high. On the other hand, the output flow rate of water is weakly correlated with the length of the reactor but has a relatively good correlation with other process variables, and it was possible to recover the value of this missing variable more successfully. Indeed, the mean value and the standard deviation of the difference for the outlet flow rate of water is very low and becomes more important when a smaller dataset size is used to build the model. The performance is therefore not attributed to the correlation between the two missing variables, but to their individual degree of correlation with other non-missing variables.

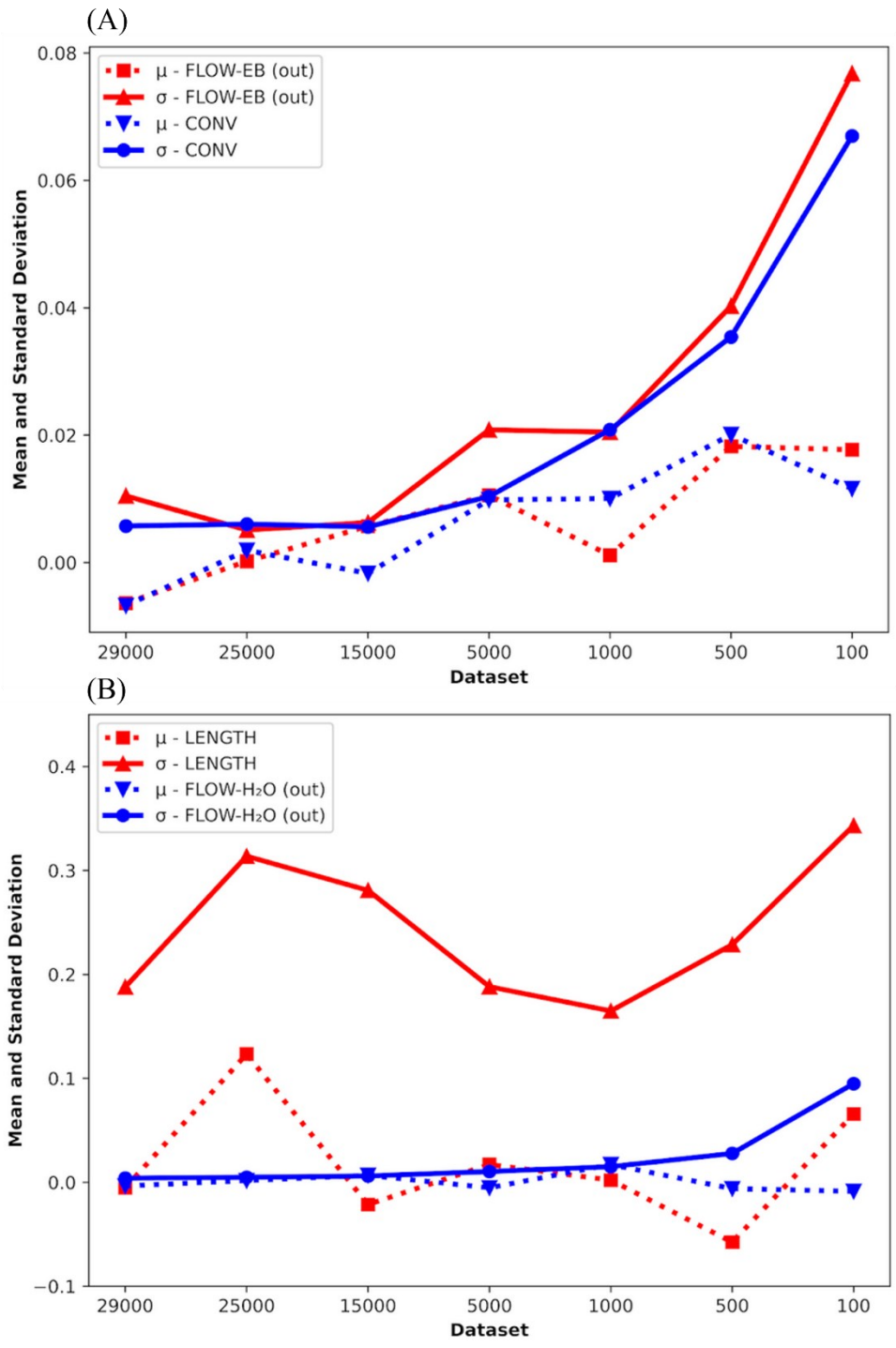


Figure 4.11. Mean and standard deviation of the prediction errors from the 1000-record dataset for two missing variables as a function of the size of the dataset used to train AANNs. (A) outlet flow rate of ethylbenzene and conversion; (B) length of the reactor and outlet flow rate of water.

As described in the previous section on the effect of one missing variable, the impact of two missing variables on predicting other variables was analyzed. Figure 4.12A shows the heat map of the standard deviation of the differences between the predicted output of the complete dataset of 1000-record evaluation clean dataset and the predicted output of this dataset with two missing variables using the AANN model developed with the dataset of 29000 records. The first five pairs correspond to missing process variables having high correlation coefficients, whereas the last five pairs have lower correlation coefficients. For the first five pairs of variables, the standard deviation of the differences is very low except for the case where both the inlet and outlet stream temperatures are missing. With the absence of these two temperatures, it is difficult for the AANN to recover these two temperatures with high precision, whereas the impact on the other process variables is relatively weak. For the last five pairs of weakly-correlated missing variables, the impact is more important. For example, the pair of missing variables comprised of the length and outlet flow rate of water leads to considerable standard deviations of the differences for several process variables, including the length, the flow rate of styrene, the flow rate of toluene, the flow rate of CO_2, the outlet temperature, and the yield. The highest impact was seen in the length itself, in which the standard deviation is more than 5%. The second greatest impact was on the yield with a standard deviation of approximately 4.5%.

A larger effect was observed in Figure 4.12B when the AANN model was developed with the 100-record evaluation clean dataset. This was expected since the AANN reconstructed the input vector of the process variables with reduced accuracy and the impact of missing variables is much more important. It is important to restate that the standard deviations shown in Figure 4.12 are the standard deviations of the differences between the predictions without the missing values and the predictions with the missing values, such that they clearly measure only the impact of the missing variable on their own. In general, these results show that to recover a missing value, it is important to have more measurements that have a high degree of correlation with other process variables, with the AANN model performing some data reconciliation. This was the case for the chemical species flow rates, which are related through reaction stoichiometry. However, the estimation of some other process variables, not intimately linked with other variables, is not as easy even though it is still possible to obtain a descent replacement value. One example of the latter is the input and output pressures where a missing pressure value is harder to estimate, in addition to being more affected by other missing variables.

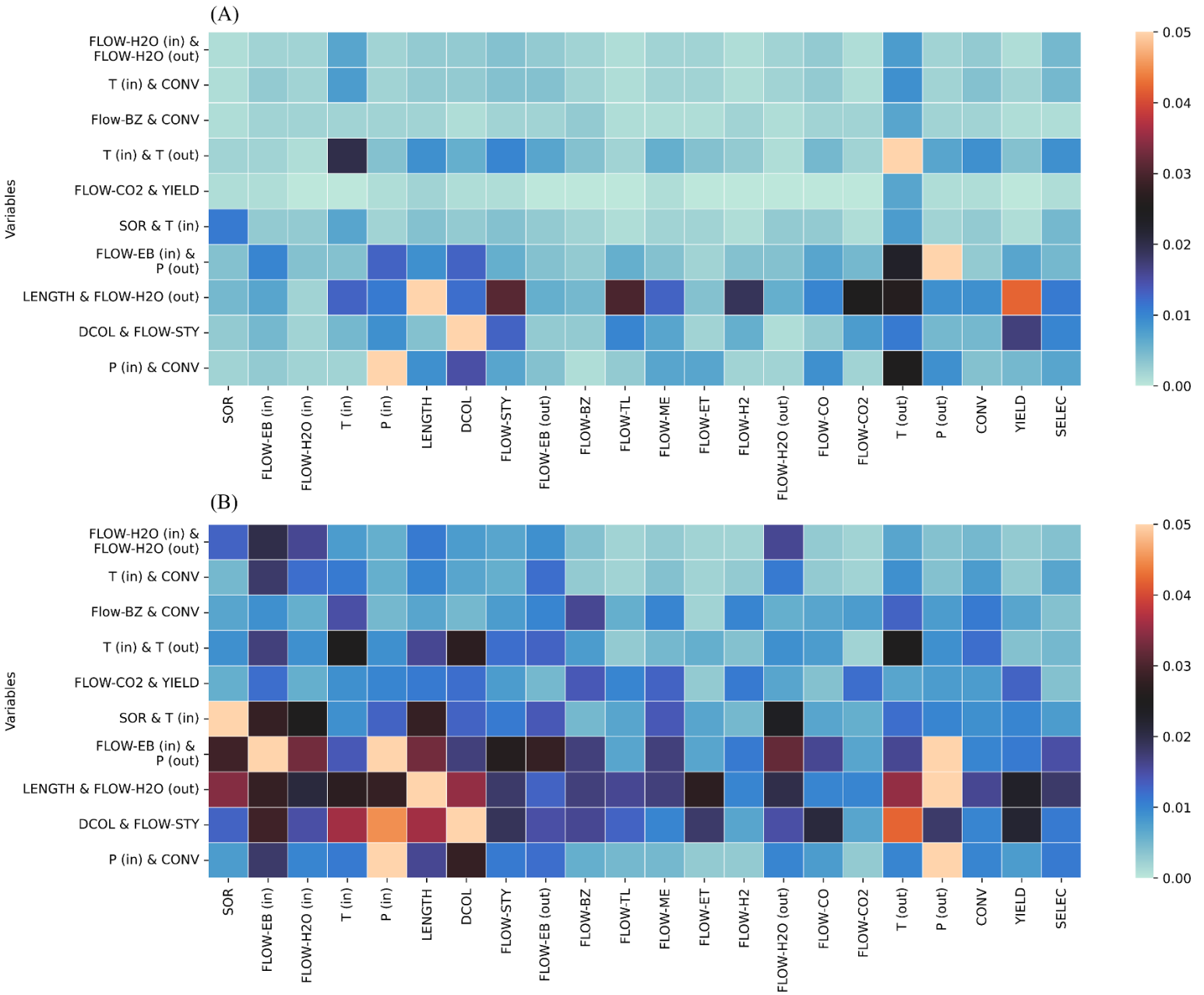


Figure 4.12. Heat map of the standard deviations of the differences between the estimations with and without the missing values, showing the effect of two missing variables on themselves and other variables. (A) 29000 record AANN; (B) 100 record AANN.

4.4.5 Performance of AANNs for three missing process variables

The performance of an AANN for recovering missing data is now explored for the case where three missing process variables in the styrene production clean dataset. As for previous sections, given the large number of combinations of three process variables selected from 22 variables, only

four sets of three variables were selected based on the correlation matrix (Figure 4.3) for illustration. Two sets of variables classified as highly correlated and two other sets of weakly-correlated variables were selected. The prediction method and the iteration procedure were as described in the previous sections.

For each case, the average values of the mean and the standard deviation of the differences between the estimated outputs with and without the missing values were calculated for all selected sets of three missing variables. Figure 4.13A presents the results for one set of three missing and highly-correlated variables for the 1000-record evaluation clean dataset. For instance, the SOR, the inlet flow rate of ethylbenzene and the inlet flow rate of water are highly correlated to each other. On average, the mean values for the estimation of the three variables are relatively small. On the other hand, the standard deviations of the differences between the estimated outputs with and without the missing values are more pronounced, especially for the SOR and the inlet flow rate of ethylbenzene. The standard deviations generally increase when the size of the dataset to train the AANN decreases. The standard deviation of the flow rate of water is significantly less, mainly because of its high correlation with the output water flow rate. Figure 4.13B presents average values of the means and the standard deviations of the differences between the estimated outputs with and without the missing values for one set of weakly-correlated variables, namely the length of the reactor, the outlet flow rate of water and the outlet flow rate of CO. As expected, the mean value and the standard deviation of the length of the reactor are relatively high since this process variable is weakly correlated with all the other process variables. On the other hand, the missing outlet water flow rate and the outlet flow rate of CO are very well estimated with very low values of the mean and the standard deviation. A small increase in the standard deviation is observed for the AANN obtained using smaller datasets.

As in previous sections where the effect of one or two missing variables on the prediction of other variables was assessed, the impact of three missing variables was examined. Figure 4.14 shows the heat map of the standard deviation of the differences between the predicted outputs of the completely separate clean evaluation dataset of 1000 records. Figure 4.14A and Figure 4.14B depict the variation of the standard deviation for four sets of three missing variables of highly-correlated and weakly-correlated variables, respectively.

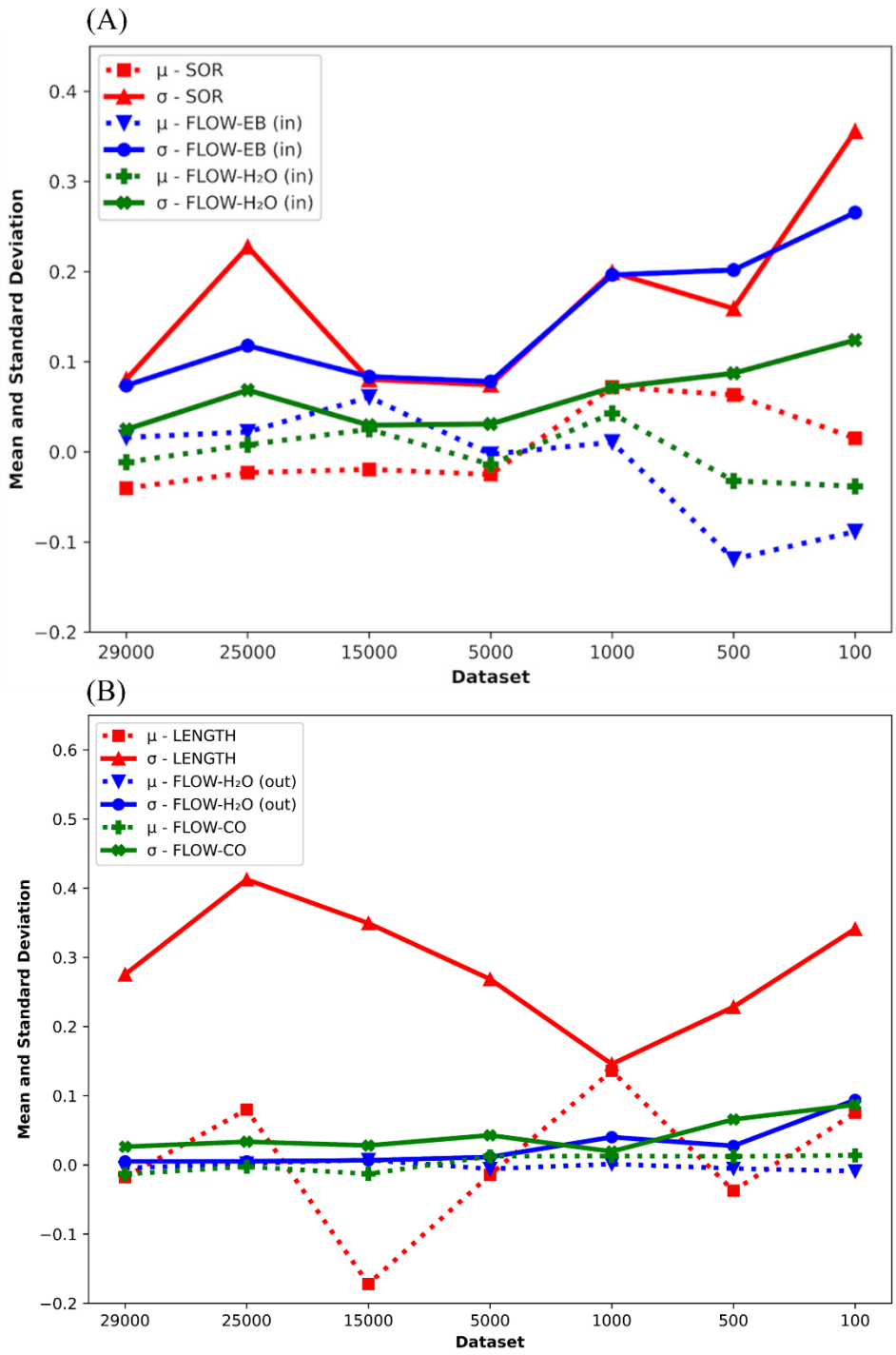


Figure 4.13. Mean and standard deviation of the prediction errors for three missing variables as a function of the size of the dataset used to train AANNs. (A) SOR, inlet flow rate of ethylbenzene, and inlet flow rate of water; (B) length of reactor, outlet flow rate of water, and outlet flow rate of CO.

Figure 4.14A shows that the combination of critical missing reactor inlet information (SOR, the inlet flow rate of ethylbenzene and the inlet flow rate of water) has significantly affected the prediction of SOR and the inlet flow rate of ethylbenzene. On the other hand, when the missing process variables were the conversion, yield and selectivity, the missing variables were estimated very well for the AANN model fitted with the 29000-record dataset. For the AANN obtained with the 100-record dataset (Figure 4.14B), the predictive performances are slightly deteriorated, but still quite acceptable, especially for process variables that have a good level of correlation in a way to constrain process variables closer to their measured values. Nevertheless, by comparing Figure 4.12A and Figure 4.14A, it can be concluded that the presence of an additional missing variable, such as the outlet flow rate of CO to the combination of the reactor length and the outlet flow rate of water, not only increased the magnitude of the errors for other non-missing variables but also extend the number of variables affected by the number of missing values in the dataset. It is expected that an increase in the number of missing variables will lead to a poorer estimation of the missing variables, associated with a deterioration in the quality of the information in the dataset.

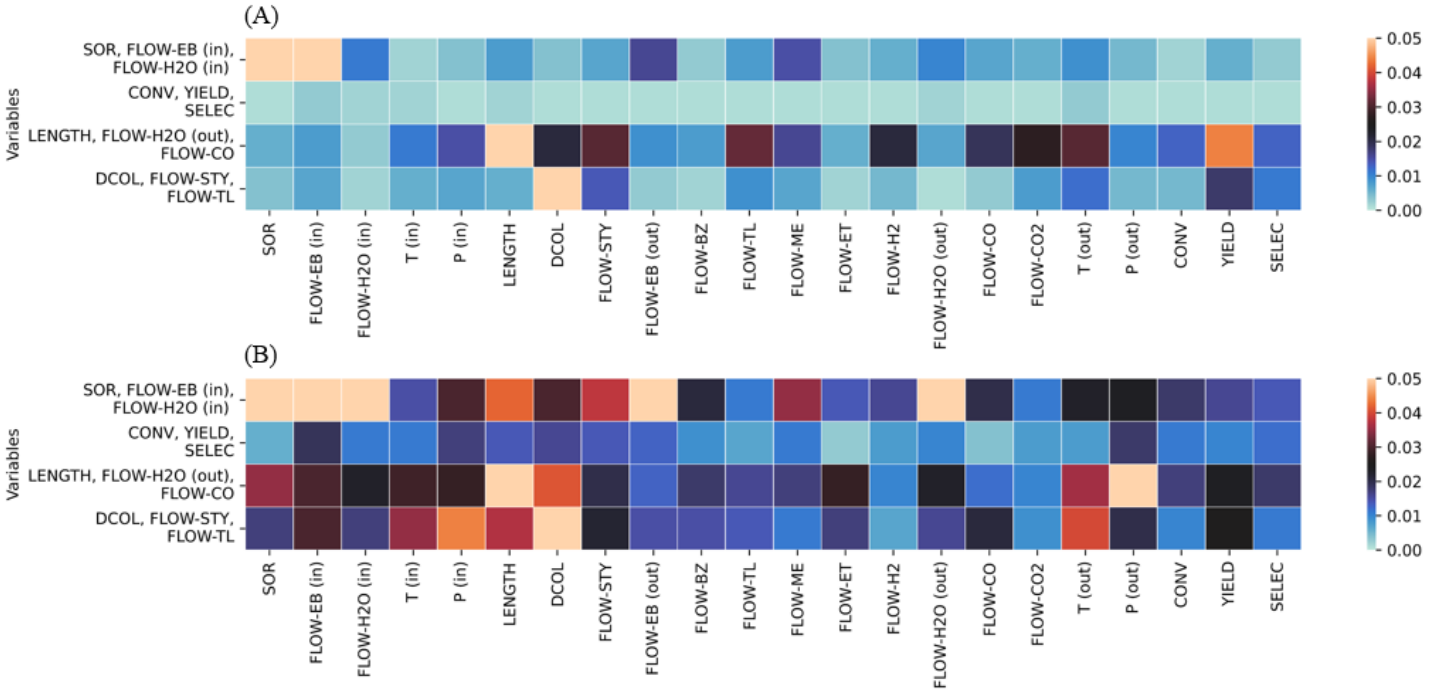


Figure 4.14. Heat map of the standard deviations of the differences between the estimations with and without the missing values showing the effect of three missing variables on themselves and other variables. (A) 29000 record AANN; (B) 100 record AANN.

4.4.6 Building AANN models on noisy data

Results of Sections 3.1 to 3.4 were presented for the case of clean data, that is without the presence of noise. In this section, the same procedure is followed to build different AANN models with the same architecture, hyperparameters, training/validation/test splitting sizes, and performance criteria for all seven datasets. However, AANN models are built and trained on noisy data as explained in Section 2.4. The MSE for each AANN model of noisy data, as well as clean data (same as Figure 4.5), is presented in Table 4.4. The comparison between the performance of AANN models from clean and noisy data indicates the overall lower error of AANN model for each clean dataset. However, the AANN results for noisy data still show quite low errors which emphasize the fact that AANN is capable of capturing the necessary information even if the data is contaminated with noise.

Table 4.4. MSE of AANN models for two cases: clean data and noisy data.

	Clean Data			**Noisy Data**		
Datasets	Training	Validation	Test	Training	Validation	Test
29000	0.000053	0.00007	0.000073	0.00025	0.00025	0.00025
25000	0.000075	0.000093	0.000093	0.00027	0.00028	0.00028
15000	0.000064	0.000075	0.000077	0.00029	0.00030	0.00031
5000	0.00015	0.00016	0.00017	0.00040	0.00047	0.00050
1000	0.00098	0.0012	0.0013	0.0011	0.0012	0.0015
500	0.0018	0.0027	0.0027	0.0037	0.0050	0.0046
100	0.0022	0.0079	0.0104	0.0020	0.0052	0.0064

To compare the ability of built AANN models on noisy data to predict missing values, the same evaluation dataset of 1000 records (including noise) has been considered, and the same procedure is repeated for the case of having one missing variable. Again, the values of a process variable were replaced by the mean value and the whole dataset of 1000 records was presented to the AANN model to predict the missing values. The new estimated outputs of the missing variable were again fed back to the AANN model to obtain new estimates of the missing variable. This process is repeated for 10 iterations. This process was carried out for all 22 process variables.

Table 4.5. MAE between denormalized observed data and predicted missing values of different AANN models of clean and noisy datasets.

	Datasets					
	29000		100		Average of All Datasets	
Variables	Clean Data	Noisy Data	Clean Data	Noisy Data	Clean Data	Noisy Data
SOR	0.13	0.15	1.10	0.80	0.42	0.35
FLOW-EB (in)	0.37	0.90	6.06	4.63	1.76	1.98
FLOW-H_2O (in)	2.46	6.84	39.58	34.84	11.28	13.19
T (in)	3.83	3.24	8.94	10.49	4.95	5.56
P (in)	13.56	10.26	18.62	21.89	12.92	13.00
LENGTH	0.66	0.89	1.68	1.59	1.13	1.06
DCOL	0.48	0.32	1.85	1.45	0.77	0.71
FLOW-STY	0.23	0.45	1.77	1.86	0.70	0.83
FLOW-EB (out)	0.26	0.58	3.67	3.35	1.12	1.32
FLOW-BZ	0.06	0.06	0.34	0.29	0.32	0.13
FLOW-TL	0.10	0.24	1.46	1.09	0.46	0.48
FLOW-ME	0.01	0.03	0.10	0.13	0.27	0.06
FLOW-ET	0.00	0.00	0.02	0.02	0.01	0.01
FLOW-H_2	0.38	0.80	5.04	4.05	1.60	1.67
FLOW-H_2O (out)	2.54	7.01	41.78	33.66	11.13	13.62
FLOW-CO	0.00	0.00	0.01	0.01	0.01	0.00
FLOW-CO_2	0.11	0.30	1.88	1.53	0.55	0.57
T (out)	8.46	7.98	12.44	13.66	8.68	12.96
P (out)	9.67	13.93	22.47	23.94	13.76	20.09
CONV	0.01	0.01	0.03	0.03	0.01	0.01
YIELD	0.00	0.01	0.05	0.04	0.02	0.02
SELEC	0.36	0.67	2.38	2.23	1.11	1.17

The predicted values of AANN models of clean and noisy data are denormalized to their original values and the mean absolute error (MAE) between the observed data and predicted values from AANN models of clean and noisy data was calculated, separately. The MAE between the observed data and the predicted missing data for both the clean dataset and the noisy dataset for 1000 records are presented in Table 4.5 for each process variable. The results are provided for two AANN

models (models with 29000-records and 100-records). Furthermore, the average of MAE for all AANN models of the observed data and the predicted missing data for both the clean dataset and the noisy dataset of 1000-records was calculated and presented in Table 4.5. From these results, it can be concluded that AANN models built using clean and noisy data are able to predict missing values quite satisfactorily and especially the AANN models trained with larger datasets. Even the AANN model fitted with the 100-records dataset was able to predict missing values within an acceptable range. Comparing the MAE of models with the clean and noisy data, it was observed that AANN is capable of predicting missing values and can also capture very well the non-linear relationships between the process variables. The success of AANN models for the estimation of missing variables, even built with noisy data, is due to the noise filtering ability of AANN and its ability to reproduce adequately the input process variables.

4.5 Conclusion

This work investigated the ability of AANN to predict the missing values of process variables in a chemical engineering process. The styrene production process was selected as the target process and a large steady-state dataset containing 30000 records was generated with 22 process variables. Of the main dataset with 30000 records, 1000 records were set aside as an evaluation dataset for investigating the prediction of missing values. The remaining dataset was divided into seven separate datasets, namely 29000, 25000, 15000, 5000, 1000, 500, and 100 record datasets, to develop AANN models to investigate the impact of the size of a dataset on their predictive ability. The AANN models developed with different sizes were able to reproduce the input variables with good performance. It was determined that the prediction error decreases with an increase in the size of the dataset used to develop the AANN with a maximum error of about 1% for the AANN model based on the dataset with 100 records.

The 1000-record evaluation dataset was used to assess the ability of the seven AANNs, developed with datasets of different sizes, to investigate data imputation when one, two and three variables are missing. A set of variables was selected based on their correlation coefficients with other process variables to study various scenarios. Akin to the precision of the AANN with different sizes of the training dataset, results showed that the estimation error of missing values decreased with an increase in the size of the dataset used to develop the AANN. Results clearly show that the estimation error and the impact of missing variables on the prediction of non-missing

variables increased with an increase in the number of missing variables. In addition, the correlation coefficient between process variables played a significant role in the efficiency of AANN models to estimate the missing values. Indeed, the weakly-correlated variables exhibited the poorest estimation with higher standard deviations and yet providing in most cases acceptable values. This procedure was repeated on noisy data, and it was observed that AANN was able to predict missing values with high accuracy even though it was trained on noisy and corrupted data. This methodology can be adapted for various chemical engineering problems, especially when dealing with data obtained for different sources displaying inherent discrepancies in the categories of data reported.

4.6 References

1. Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., & Tabona, O. (2021). A survey on missing data in machine learning. *Journal of Big Data*, *8*(1), 1–37. https://doi.org/10.1186/s40537-021-00516-9
2. Suthar, B., Patel, H., & Goswami, A. (2012). A survey: Classification of imputation methods in data mining. *International Journal of Emerging Technology and Advanced Engineering*, *2*(1), 309–312.
3. Imtiaz, S. A., & Shah, S. L. (2008). Treatment of missing values in process data analysis. *The Canadian Journal of Chemical Engineering*, *86*(5), 838–858. https://doi.org/10.1002/cjce.20099
4. Li, L., Zhang, J., Wang, Y., & Ran, B. (2018). Missing value imputation for traffic-related time series data based on a multi-view learning method. *IEEE Transactions on Intelligent Transportation Systems*, *20*(8), 2933–2943. https://doi.org/10.1109/TITS.2018.2869768
5. Raj, B., & Stern, R. M. (2005). Missing-feature approaches in speech recognition. *IEEE Signal Processing Magazine*, *22*(5), 101–116. https://doi.org/10.1109/MSP.2005.1511828
6. De Leeuw, E. D. (2001). Reducing missing data in surveys: An overview of methods. *Quality and Quantity*, *35*, 147–160. https://doi.org/10.1023/A:1010395805406
7. Luo, Y. (2022). Evaluating the state of the art in missing data imputation for clinical data. *Briefings in Bioinformatics*, *23*(1), bbab489. https://doi.org/10.1093/bib/bbab489

8. Zhang, Y., & Thorburn, P. J. (2022). Handling missing data in near real-time environmental monitoring: A system and a review of selected methods. *Future Generation Computer Systems*, *128*, 63–72. https://doi.org/10.1016/j.future.2021.09.033
9. Crowe, C. M. (1996). Data reconciliation—Progress and challenges. *Journal of Process Control*, *6*(2–3), 89–98. https://doi.org/10.1016/0959-1524(96)00012-1
10. Eckert, F., Fort, T. C., Schott, P. K., & Yang, N. J. (2020). *Imputing Missing Values in the US Census Bureau's County Business Patterns* (Working Paper No. 26632). National Bureau of Economic Research. https://doi.org/10.3386/w26632
11. Venkatasubramanian, V. (2019). The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal*, *65*(2), 466–478. https://doi.org/10.1002/aic.16489
12. Lakshminarayan, K., Harp, S. A., & Samad, T. (1999). Imputation of missing data in industrial databases. *Applied Intelligence*, *11*(3), 259–275. https://doi.org/10.1023/A:1008334909089
13. Lin, W.-C., & Tsai, C.-F. (2020). Missing value imputation: A review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, *53*(2), 1487–1509. https://doi.org/10.1007/s10462-019-09709-4
14. Boucheikhchoukh, A., Thibault, J., & Fauteux-Lefebvre, C. (2020). Catalyst design using artificial intelligence: SO2 to SO3 case study. *The Canadian Journal of Chemical Engineering*, *98*(9), 2016–2031. https://doi.org/https://doi.org/10.1002/cjce.23756
15. Musil, C. M., Warner, C. B., Yobas, P. K., & Jones, S. L. (2002). A comparison of imputation techniques for handling missing data. *Western Journal of Nursing Research*, *24*(7), 815–829. https://doi.org/10.1177/0193945027624770
16. Saar-Tsechansky, M., & Provost, F. (2007). Handling missing values when applying classification models. *Journal of Machine Learning Research*, *8*, 1623–1657. https://doi.org/10.5555/1314498.1314553
17. Kang, H. (2013). The prevention and handling of the missing data. *Korean Journal of Anesthesiology*, *64*(5), 402–406. https://doi.org/10.4097/kjae.2013.64.5.402
18. Donders, A. R. T., Van Der Heijden, G. J., Stijnen, T., & Moons, K. G. (2006). A gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology*, *59*(10), 1087–1091. https://doi.org/10.1016/j.jclinepi.2006.01.014

19. Wafaa Mustafa Hameed, N. A. A. (2022). Comparison of Seventeen Missing Value Imputation Techniques. *Journal of Hunan University Natural Sciences*, *49*(7). https://doi.org/10.55463/issn.1674-2974.49.7.4
20. Bell, B. A., Kromrey, J. D., & Ferron, J. M. (2009). Missing data and complex samples: The impact of listwise deletion vs. Subpopulation analysis on statistical bias and hypothesis test results when data are MCAR and MAR. *Proceedings of the Joint Statistical Meetings, Survey Research Methods Section*, *26*, 759–4770.
21. Marsh, H. W. (1998). Pairwise deletion for missing data in structural equation models: Nonpositive definite matrices, parameter estimates, goodness of fit, and adjusted sample sizes. *Structural Equation Modeling: A Multidisciplinary Journal*, *5*(1), 22–36. https://doi.org/10.1080/10705519809540087
22. Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American Statistical Association*, *91*(434), 473–489. https://doi.org/10.1080/01621459.1996.10476908
23. Noor, N. M., Yahaya, A. S., Ramli, N. A., & Abdullah, M. M. A. (2006). The replacement of missing values of continuous air pollution monitoring data using mean top bottom imputation technique. *Journal of Engineering Research & Education*, *3*, 96–105.
24. Little, R. J. (1992). Regression with missing X's: A review. *Journal of the American Statistical Association*, *87*(420), 1227–1237. https://doi.org/10.1080/01621459.1992.10476282
25. Tahmasbi, H., Amoozgar, M., & Adine, H. (2015). Replacement of missing values and its effect on the classification accuracy in medical data mining. *Journal of Health and Biomedical Informatics*, *2*(1), 24–32.
26. Magnani, M. (2004). *Techniques for dealing with missing data in knowledge discovery tasks*. *15*(01), 2007.
27. Fletcher Mercaldo, S., & Blume, J. D. (2020). Missing data and prediction: The pattern submodel. *Biostatistics*, *21*(2), 236–252. https://doi.org/10.1093/biostatistics/kxy040
28. Jäger, S., Allhorn, A., & Bießmann, F. (2021). A benchmark for data imputation methods. *Frontiers in Big Data*, *4*, 693674. https://doi.org/10.3389/fdata.2021.693674
29. Curley, C., Krause, R. M., Feiock, R., & Hawkins, C. V. (2019). Dealing with Missing Data: A Comparative Exploration of Approaches Using the Integrated City Sustainability Database. *Urban Affairs Review*, *55*(2), 591–615. https://doi.org/10.1177/1078087417726394

30. Silva, L. O., & Zárate, L. E. (2014). A brief review of the main approaches for treatment of missing data. *Intelligent Data Analysis*, *18*(6), 1177–1198. https://doi.org/10.3233/IDA-140690
31. Papageorgiou, G., Grant, S. W., Takkenberg, J. J., & Mokhles, M. M. (2018). Statistical primer: How to deal with missing data in scientific research? *Interactive Cardiovascular and Thoracic Surgery*, *27*(2), 153–158. https://doi.org/10.1093/icvts/ivy102
32. Tealab, A. (2018). Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, *3*(2), 334–340. https://doi.org/10.1016/j.fcij.2018.10.003
33. Gheyas, I. A., & Smith, L. S. (2010). A neural network-based framework for the reconstruction of incomplete data sets. *Neurocomputing*, *73*(16–18), 3039–3065. https://doi.org/10.1016/j.neucom.2010.06.021
34. Sharpe, P. K., & Solly, R. J. (1995). Dealing with missing values in neural network-based diagnostic systems. *Neural Computing & Applications*, *3*(2), 73–77. https://doi.org/doi.org/10.1007/BF01421959
35. Ankaiah, N., & Ravi, V. (2011). A novel soft computing hybrid for data imputation. *Proceedings of the International Conference on Data Science (ICDATA)*, 1.
36. Gondara, L., & Wang, K. (2018). Mida: Multiple imputation using denoising autoencoders. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 260–272. https://doi.org/10.1007/978-3-319-93040-4_21
37. Abdella, M., & Marwala, T. (2005). The use of genetic algorithms and neural networks to approximate missing data in database. *IEEE 3rd International Conference on Computational Cybernetics, 2005. ICCC 2005.*, 207–212. https://doi.org/10.1109/ICCCYB.2005.1511574
38. Betechuoh, B. L., Marwala, T., & Tettey, T. (2006). Autoencoder networks for HIV classification. *Current Science*, *91*(11), 1467–1473.
39. Marwala, T., & Chakraverty, S. (2006). Fault classification in structures with incomplete measured data using autoassociative neural networks and genetic algorithm. *Current Science*, *90*(4), 542–548.
40. Mistry, F. J., Nelwamondo, F. V., & Marwala, T. (2009). Missing data estimation using principle component analysis and autoassociative neural networks. *Journal of Systemics, Cybernatics and Informatics*, *7*(3), 72–79.

41. Mohamed, A. K., Nelwamondo, F. V., & Marwala, T. (2007). Estimating missing data using neural network techniques, principal component analysis and genetic algorithms. *Proceedings of the Eighteenth Annual Symposium of the Pattern Recognition Association of South Africa*.
42. Ravi, V., & Krishna, M. (2014). A new online data imputation method based on general regression auto associative neural network. *Neurocomputing*, *138*, 106–113. https://doi.org/10.1016/j.neucom.2014.02.037
43. Gautam, C., & Ravi, V. (2015). Data imputation via evolutionary computation, clustering and a neural network. *Neurocomputing*, *156*, 134–142. https://doi.org/10.1016/j.neucom.2014.12.073
44. Gautam, C., & Ravi, V. (2015). Counter propagation auto-associative neural network based data imputation. *Information Sciences*, *325*, 288–299. https://doi.org/10.1016/j.ins.2015.07.016
45. Lu, Y., Yang, D., Li, Z., Peng, X., & Zhong, W. (2022). Neural networks with upper and lower bound constraints and its application on industrial soft sensing modeling with missing values. *Knowledge-Based Systems*, *243*, 108510. https://doi.org/10.1016/j.knosys.2022.108510
46. Xiao, H., Huang, D., Pan, Y., Liu, Y., & Song, K. (2017). Fault diagnosis and prognosis of wastewater processes with incomplete data by the auto-associative neural networks and ARMA model. *Chemometrics and Intelligent Laboratory Systems*, *161*, 96–107. https://doi.org/10.1016/j.chemolab.2016.12.009
47. Pan, Z., Wang, Y., Wang, K., Chen, H., Yang, C., & Gui, W. (2022). Imputation of missing values in time series using an adaptive-learned median-filled deep autoencoder. *IEEE Transactions on Cybernetics*. https://doi.org/10.1109/TCYB.2022.3167995
48. Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, *37*(2), 233–243. https://doi.org/10.1002/aic.690370209
49. Heo, S., & Lee, J. H. (2019). Parallel neural networks for improved nonlinear principal component analysis. *Computers & Chemical Engineering*, *127*, 1–10. https://doi.org/10.1016/j.compchemeng.2019.05.011
50. Du, Y.-G., Hodouin, D., & Thibault, J. (1997). Use of a novel autoassociative neural network for nonlinear steady-state data reconciliation. *AIChE Journal*, *43*(7), 1785–1796. https://doi.org/10.1002/aic.690430714

51. Marseguerra, M., & Zoia, A. (2005). The autoassociative neural network in signal analysis: I. The data dimensionality reduction and its geometric interpretation. *Annals of Nuclear Energy*, *32*(11), 1191–1206. https://doi.org/10.1016/j.anucene.2005.03.006
52. Kramer, M. A. (1992). Autoassociative neural networks. *Computers & Chemical Engineering*, *16*(4), 313–328. https://doi.org/10.1016/0098-1354(92)80051-A
53. Thibault, J. (1991). Feedforward neural networks for the identification of dynamic processes. *Chemical Engineering Communications*, *105*(1), 109–128. https://doi.org/10.1080/00986449108911521
54. Muhler, M., Schlögl, R., & Ertl, G. (1992). The nature of the iron oxide-based catalyst for dehydrogenation of ethylbenzene to styrene 2. Surface chemistry of the active phase. *Journal of Catalysis*, *138*(2), 413–444. https://doi.org/10.1016/0021-9517(92)90295-S
55. Shekhah, O., Ranke, W., Schüle, A., Kolios, G., & Schlögl, R. (2003). Styrene synthesis: High conversion over unpromoted iron oxide catalysts under practical working conditions. *Angewandte Chemie International Edition*, *42*(46), 5760–5763. https://doi.org/10.1002/anie.200352135
56. Rase, H. F. (2000). *Handbook of commercial catalysts: Heterogeneous catalysts*. CRC press.
57. Sheppard, C. M., Maier, E. E., & Caram, H. S. (1986). Ethylbenzene dehydrogenation reactor model. *Industrial & Engineering Chemistry Process Design and Development*, *25*(1), 207–210. https://doi.org/10.1021/i200032a033
58. Leite, B., da Costa, A. O. S., & da Costa Junior, E. F. (2021). Simulation and optimization of axial-flow and radial-flow reactors for dehydrogenation of ethylbenzene into styrene based on a heterogeneous kinetic model. *Chemical Engineering Science*, *244*, 116805. https://doi.org/10.1016/j.ces.2021.116805
59. Fettaka, S., Gupta, Y. P., & Thibault, J. (2012). Multiobjective optimization of an industrial styrene reactor using the dual population evolutionary algorithm (DPEA). *International Journal of Chemical Reactor Engineering*, *10*(1). https://doi.org/https://doi-org/10.1515/1542-6580.2828
60. Li, Y., Rangaiah, G. P., & Ray, A. K. (2003). Optimization of styrene reactor design for two objectives using a genetic algorithm. *International Journal of Chemical Reactor Engineering*, *1*(1), A13.

61. Sancheti, S. V., & Yadav, G. D. (2021). Highly selective production of styrene by non-oxidative dehydrogenation of ethylbenzene over molybdenum-zirconium mixed oxide catalyst in fixed bed reactor: Activity, stability and kinetics. *Catalysis Communications*, *154*, 106307. https://doi.org/10.1016/j.catcom.2021.106307
62. Sheel, J. G. P., & Crowe, C. M. (1969). Simulation and optimization of an existing ethylbenzene dehydrogenation reactor. *The Canadian Journal of Chemical Engineering*, *47*(2), 183–187. https://doi.org/10.1002/cjce.5450470215
63. Thebelt, A., Wiebe, J., Kronqvist, J., Tsay, C., & Misener, R. (2022). Maximizing information from chemical engineering data sets: Applications to machine learning. *Chemical Engineering Science*, *252*, 117469. https://doi.org/https://doi.org/10.1016/j.ces.2022.117469
64. Wu, Z., Luo, J., Rincon, D., & Christofides, P. D. (2021). Machine learning-based predictive control using noisy data: Evaluating performance and robustness via a large-scale process simulator. *Chemical Engineering Research and Design*, *168*, 275–287. https://doi.org/10.1016/j.cherd.2021.02.011
65. Arslan, M., Guzel, M., Demirci, M., & Ozdemir, S. (2019). SMOTE and gaussian noise based sensor data augmentation. *2019 4th International Conference on Computer Science and Engineering (UBMK)*, 1–5. https://doi.org/10.1109/UBMK.2019.8907003
66. Wu, Z., Rincon, D., Luo, J., & Christofides, P. D. (2021). Machine learning modeling and predictive control of nonlinear processes using noisy data. *AIChE Journal*, *67*(4), e17164. https://doi.org/10.1002/aic.17164
67. Wu, Y., & Lee, T. (2018). *Reducing model complexity for DNN based large-scale audio classification*. 331–335. https://doi.org/10.1109/ICASSP.2018.8462168

Chapter 5

Optimal Design of Catalytic Conversion of SO_2 to SO_3 via Machine Learning Modelling

Farough Agin, Clémence Fauteux-Lefebvre, and Jules Thibault*
Department of Chemical and Biological Engineering
University of Ottawa, Ottawa, Ontario, K1N 6N5, Canada
*Corresponding Author: Jules Thibault

This chapter is a manuscript published in Journal of Machine Intelligence and Data Science (2025, 6 (1), 1-22).

5.1 Abstract

This study explores the use of machine learning and multi-objective optimization to improve the catalytic conversion of sulfur dioxide (SO_2) to sulfur trioxide (SO_3), a key process in sulfuric acid production and environmental mitigation. A feedforward neural network model is employed to predict the SO_2 conversion based on the catalyst composition and the prevailing operating conditions. Subsequently, multi-objective optimization methodologies are employed to identify optimal solutions that concurrently maximize conversion and productivity while minimizing the associated catalyst costs. Two case studies are conducted to determine the optimal catalyst/promoter composition and operating conditions for sulfuric acid production. The first candidate involves a combination of vanadium and potassium, while the second focuses on platinum. The study highlights the potential of these methodologies to enhance sulfuric acid production efficiency and address pollution, contributing to industrial productivity and environmental sustainability.

Keywords: SO_2 conversion, sulfuric acid, pollution mitigation, machine learning, multi-objective optimization

5.2 Introduction

In the realm of chemical engineering, the core of processes often resides in their reactions. Certain reactions have garnered heightened focus owing to their practicality, significance, and economic viability. Among these pivotal reactions within chemical processes lies the catalytic conversion of sulfur dioxide (SO_2) to sulfur trioxide (SO_3). The conversion of SO_2 has two domains of applications: the mitigation of pollution and the production of sulfuric acid (H_2SO_4). This reaction is based on the following stoichiometric equilibrium expressed by Equation (5.1):

$$SO_2(g) + \frac{1}{2}O_2(g) \rightleftarrows SO_3(g) \tag{5.1}$$

For the first application, it is uncontested that urban population growth has sparked an industrial activity boom that has significantly increased the amount of air contaminants such as SO_2 [1–3]. To remove SO_2 emanating from coal-fired power plants and mitigate pollution, industries use a variety of flue gas desulfurization (FGD) methods. FGD processes are essential for reducing air pollution caused by sulfur dioxide emissions. FGD processes can broadly be classified into two main categories: wet processes and dry processes [4–7]. Wet processes, the most used technologies, involve the utilization of limestone or slaked lime slurry as sorbents in spray towers, enabling chemical absorption to eliminate SO_2. Wet processes can eliminate up to 99% of the pollutant, producing a significant amount of hazardous material. On the other hand, dry processes involve the physical or chemical sorption of SO_2 on materials such as activated carbon or other sorbents, typically implemented in a fixed bed configuration [8–10]. Through the various technologies to remove SO_2, some also rely on its oxidation [8,11,12].

For the latter application, the SO_2 oxidation is the most important step in manufacturing H_2SO_4, the most-produced chemical product worldwide [13–15]. It is used in several applications, such as the manufacturing of phosphate fertilizers and a variety of other chemicals. It plays a role in metal processing, it serves as an electrolyte in lead-acid car batteries, and it is employed in the petroleum refining industry to remove impurities from fuel as well as other refinery products [16–18]. Using a catalyst bed, SO_2 is oxidized to produce SO_3 (Equation (5.1)). The oxidation of SO_2 is an exothermic reversible reaction, influenced by temperature and pressure, and is typically performed industrially in an adiabatic packed bed reactor [19–21]. At temperatures exceeding 400°C, the reverse reaction becomes progressively favoured, leading to a decrease in the conversion of

SO_2 as the temperature increases. However, it is worth noting that the minimum operating temperature of common SO_2 oxidation catalysts, often referred to as the strike temperature, which is also around 400°C. Balancing these conflicting requirements is achieved through the implementation of multistage catalytic reactors and inter-stage cooling. In the following process step, slightly diluted H_2SO_4 (usually as 97-98% H_2SO_4 containing 2-3% water) and SO_3 are reacted to produce more concentrated H_2SO_4, as given by Equation (5.2) [22,23].

$$SO_3(g) + H_2O(l) \rightarrow H_2SO_4(l) \quad (5.2)$$

The conversion of SO_2 to SO_3 relies heavily on the catalyst performance. While vanadium, platinum, and iron-based catalysts have been the primary focus of numerous studies, other metals have also been considered [24–27]. According to the available literature, catalysts such as sodium-vanadium, platinum-palladium, platinum, and platinum-tin are widely recognized for their predominant use in FGD. These catalysts have demonstrated their effectiveness in facilitating the desired chemical reactions involved in the removal of SO_2 from flue gases [1,28,29]. The rate of SO_2 conversion can be significantly influenced by the combination of different metals in varying quantities, particularly in promoted or bimetallic catalysts. For instance, research has demonstrated that SO_2 oxidation promoters like alkali metals, namely potassium, sodium, and cesium, strengthen the structure and bonding of the catalyst. As a result, vanadium oxide-based catalysts with promoters exhibit improved conversion [30–32]. Therefore, in the context of H_2SO_4 production, commonly employed catalysts include potassium-vanadium, cesium-vanadium, platinum, and mixtures containing vanadium combined with other compounds and/or promoters such as calcium, copper, iron, barium, manganese, and magnesium [33–35]. Ongoing research and development efforts aim to explore new catalyst formulations and modifications, enhancing their efficiency and durability in both applications.

The selection and optimization of catalysts play a critical role in ensuring effective SO_2 oxidation for sulfur removal and H_2SO_4 production [30]. However, due to a large number of parameters in any catalytic reaction, including catalyst properties and operating variables, establishing analytical correlations between all these variables and the resultant steady-state conversion is challenging. Furthermore, the proliferation of diverse catalyst types encompassing a wide array of compositions exacerbates the difficulty in identifying the optimal catalyst type and composition. In addressing this challenge, the integration of machine learning (ML) models is

promising in elucidating the underlying nonlinear relationships between variables and in the pursuit of optimal operating conditions, and aids in selecting the most suitable catalyst type and composition.

Numerous researchers have employed diverse experimental catalytic strategies, varying in operating parameters such as temperature, pressure, and concentrations of inlet feed gases [27,36,37]. Nevertheless, there is frequently a paucity of comprehensive information regarding the optimal catalyst(s) and their corresponding compositions for the desired operating conditions [30]. Consequently, robust methodologies for handling nonlinear relationships and optimization techniques have garnered recent attention. ML methodologies, such as artificial neural networks (ANNs), present a promising avenue for constructing nonlinear models incorporating all relevant variables to ascertain the optimal catalyst and its composition, thereby enhancing both efficiency and conversion. Despite the growing utilization of ML methodologies across various disciplines, their application within the catalysis domain remains nascent [38]. The major reason is the lack of universal datasets in catalytic activities [39]. Catalyst development often relies on empirical trial-and-error methods based on physicochemical properties, necessitating significant time and effort to identify optimal solutions. Automated ML approaches have demonstrated efficacy in refining models, elucidating catalytic mechanisms, and generating innovative concepts for catalytic design [38,40–43].

This study relies on a comprehensive dataset on the catalytic conversion of SO_2 [30], updated to 2023, with a diverse array of catalysts and operating parameters. The dataset, which includes two ranges of SO_2 mole fractions (below 1% for FGD and above 7% for H_2SO_4 production), is refined for accuracy and consistency. An optimal ANN model is identified through meticulous hyperparameter tuning (to optimize the ANN model's architecture and parameters) and used within a multi-objective optimization (MOO) framework to address trade-offs in the conversion process and generate the Pareto domain. The study uses the non-dominated sorting genetic algorithm (NSGA-II) and the net flow method (NFM) to identify and rank Pareto-optimal solutions. This approach integrates advanced computational methodologies with chemical engineering principles, contributing to efficiency enhancement and cost reduction in catalytic processes.

5.3 Methodology

This study comprises three primary sections. The initial section focuses on the development of an ANN model. Within this section, data collection is conducted, followed by rigorous preprocessing techniques to ensure data integrity. Hyperparameter tuning is subsequently employed to optimize the ANN's performance, involving the construction and evaluation of multiple ANN models. The final model selection is based on the comparative performance assessment of each model. The second section entails the formulation of three distinct objectives, which are framed as a MOO problem. Utilizing the NSGA-II, the Pareto domain - a set of solutions representing the trade-offs between conflicting objectives - is identified. In the third and final section, a multi-criteria decision-making approach known as NFM is applied to rank the solutions approximating the Pareto domain. This method facilitates the identification of the optimal solution. The procedural workflow is illustrated in Figure 5.1.

Each section of this study will be explained in detail, providing comprehensive coverage of the respective steps involved. Furthermore, the rationale behind each methodological choice will be discussed, along with its relevance and contribution to the overall research objectives.

5.3.1 Artificial neural network model development

In 1943, McCulloch and Pitts proposed a computational model inspired by the human brain, which sparked research on ANNs [44]. ANN models have the ability to learn, recognize, and solve complex problems. Among the several types of ANNs, feedforward neural networks (FFNNs) are particularly interesting due to their structural representation as a computational model in a network form. This structural representation is what allows FFNNs to be a universal function approximator, capable of approximating any continuous function [45]. The FFNN can address a broad range of problems related to pattern recognition and prediction. This ability has been embraced by various researchers, who have appreciated FFNNs for their universal approximation ability [45–47]. FFNNs are computational models that comprise multiple interconnected neurons or nodes, arranged in a layered structure, where each layer is connected in a forward direction to the preceding layer. These neural networks have a specific structural configuration (Figure 5.2A) and are capable of processing information through the synaptic links or weights that connect the nodes (Figure 5.2B).

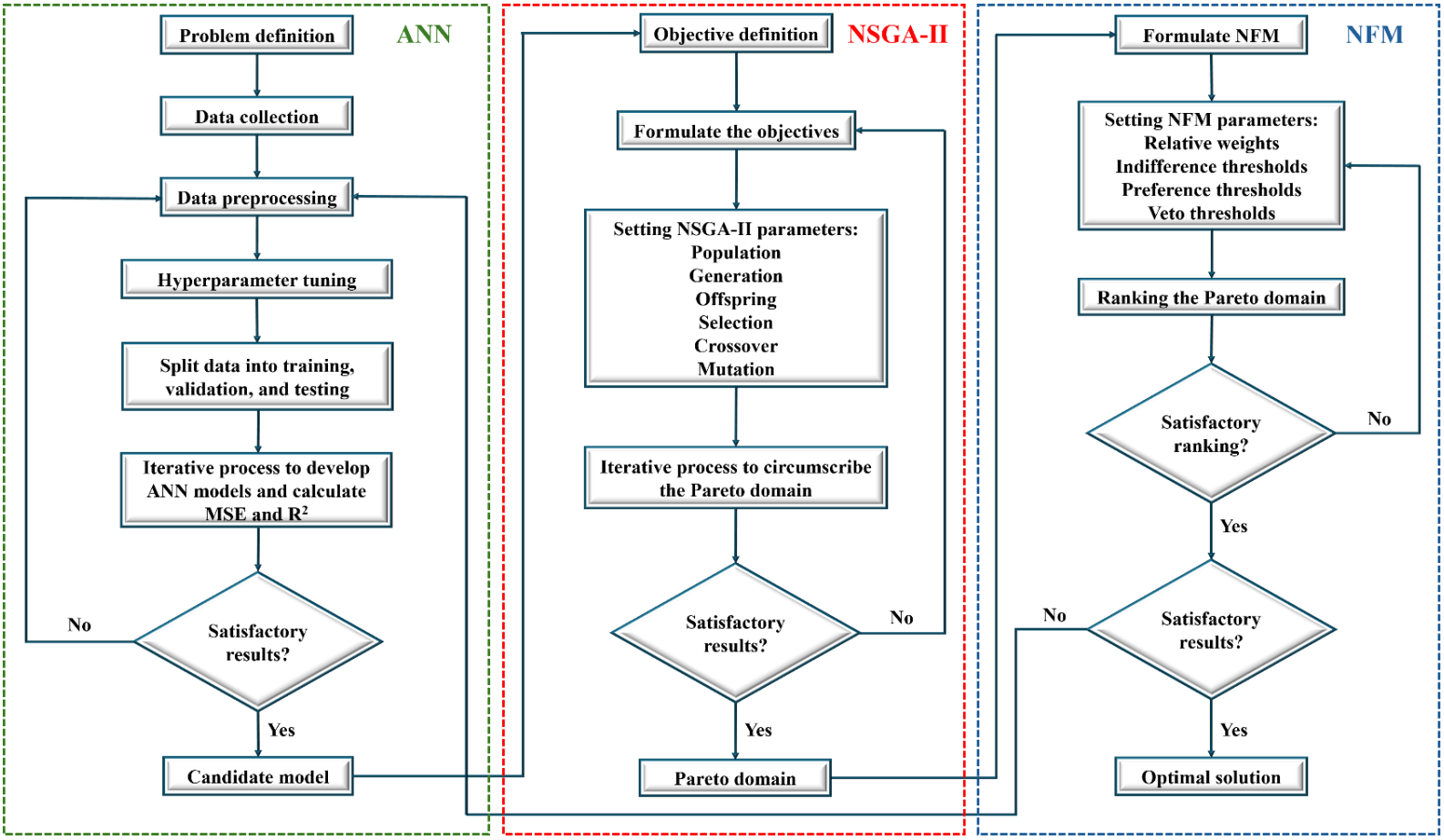


Figure 5.1. Flowchart of the complete steps toward the optimization of the catalytic conversion of SO_2.

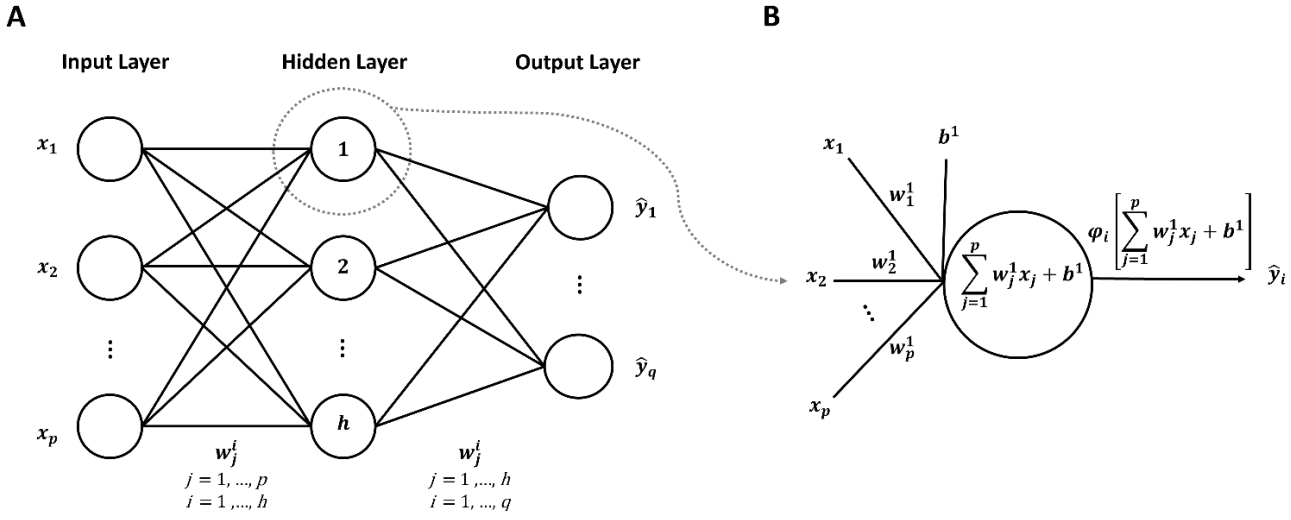


Figure 5.2. (A) FFNN with three layers and (B) a sample node of the network.

The process of supervised learning involves minimizing a cost function expressed as the difference between the desired output y_i and the output of the model $\hat{y}_i$. Various cost functions can be defined for this purpose. For example, mean squared error (MSE) is a commonly used cost function in regression problems, which can be expressed as [45]:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{q}(y_{ij} - \hat{y}_{ij})^2 \tag{5.3}$$

where N is the number of training points and q is the number of output neurons.

There are various conventional derivative-based methods, such as Quasi-Newton [48], Levenberg-Marquardt [49], adaptive moment estimation (Adam) [50], etc., used to adjust the connection weights to minimize the cost function. Adam is a popular algorithm for solving complex issues involving a large number of variables or data.

As depicted in Figure 5.1, the initial step in the implementation of an ANN involves defining the problem statement. In the context of this project, the objective is to construct an ANN model capable of predicting the conversion of SO_2. Following the problem definition, the subsequent step entails the acquisition of raw data. While direct experimentation within a specific system necessitates considerable resources in terms of time, labour, infrastructure, and financial investment, a more practical approach involves leveraging existing literature, which encompasses a wealth of research findings accessible online. Across various domains of chemical engineering, extensive experimental data have been amassed over numerous decades. However, collating this data is a labour-intensive endeavour, requiring a meticulous review of potential sources to ascertain their relevance to the project and extract pertinent information.

The inclusion criteria of data from scientific articles within the dataset necessitate the documentation of catalytic activity tests employing heterogeneous catalysts. Furthermore, these articles must provide details regarding the experimental conditions, encompassing independent variables, as well as the outcomes of these tests, represented by dependent variables. The dataset thus encompasses a spectrum of information, including the composition and properties of the catalysts (e.g., specific surface area, pore characteristics, particle size), synthesis methodologies (e.g., calcination parameters such as temperature and time), operating conditions including temperature, pressure, catalyst mass, inlet volumetric flow rate, as well as the mole fractions of O_2

and SO_2 within the system. Additionally, the dataset incorporates the conversion of SO_2 as the system's output variable.

For the dataset preparation, a comprehensive screening process was undertaken, considering 152 literature papers, culminating in the identification of 32 papers deemed suitable for data extraction. Numerous considerations influenced the selection criteria for inputs into the final database. Notably, a significant challenge arose from the inconsistent reporting of information within some of the candidate papers, resulting in the omission of certain variables. Consequently, catalysts and promoters characterized by a paucity of data points and minimal variance were excluded from consideration as final inputs.

Table 5.1 presents the finalized list of the input and output variables within the database. Notably, 14 active metals and promoters emerged as the preferred candidates for inclusion, along with three support materials denoted by binary indicators reflecting their presence or absence. Furthermore, operating parameters such as temperature, pressure, the ratio of catalyst weight to volumetric flow rate (W/V), and the mole fractions of SO_2 and O_2 complete the list of input variables. The output variable, representing the conversion of SO_2 to SO_3, completes the database structure. Although additional variables such as pore volume, pore size, and surface area were gathered from the literature review, they were not included in the main dataset due to the substantial amounts of data not reported in many papers.

Following the data collection, the subsequent stage entails preprocessing to render the raw data suitable for its utilization within a ML algorithm. Data preprocessing serves to facilitate the management and manipulation of complex datasets, a task that often necessitates a considerable amount of processing time [51]. Despite its pivotal role in model development, data preprocessing is occasionally overlooked compared to other stages; nevertheless, it typically consumes over 50% of the total time allocated to data mining endeavours [52]. Various techniques are employed in the preprocessing phase to enhance the quality of datasets, and these techniques will be succinctly discussed in the subsequent discussion.

Data in real-world scenarios often contains incompleteness, noise, and inconsistency. Consequently, data-cleaning efforts aim to address these issues meticulously by identifying outliers and rectifying inconsistencies within the dataset [53].

Table 5.1. SO_2 conversion databank description with 22 input variables and one output variable used to develop the ANN model to predict conversion.

ANN Variables	Description	Unit	Mean	Standard deviation	Min	Max
Calcium	Active metal / promoter	Mass fraction (%)	0.088	0.386	0	2.257
Cerium		Mass fraction (%)	0.123	0.662	0	4.885
Cesium		Mass fraction (%)	1.183	2.936	0	11.05
Copper		Mass fraction (%)	0.012	0.003	0	0.024
Iron		Mass fraction (%)	0.044	0.189	0	1.832
Lanthanum		Mass fraction (%)	0.094	0.687	0	6.826
Magnesium		Mass fraction (%)	0.001	0.010	0	0.100
Manganese		Mass fraction (%)	0.019	0.118	0	0.750
Palladium		Mass fraction (%)	0.022	0.144	0	2
Platinum		Mass fraction (%)	0.539	1.497	0	9.091
Potassium		Mass fraction (%)	4.052	5.545	0	18.428
Sodium		Mass fraction (%)	0.034	0.384	0	5.463
Tin		Mass fraction (%)	0.153	0.473	0	2.200
Vanadium		Mass fraction (%)	1.904	2.059	0	8
Alumina	Support material	0 or 1	0.161	0.367	0	1
Titania		0 or 1	0.065	0.246	0	1
Silicate		0 or 1	0.775	0.418	0	1
Temperature	Operating parameter	°C	457	90	204.15	799
Pressure		Atm	1.535	1.864	1	10
W/V		kg_{cat}·s/L	4.223	8.819	0.003	28.379
SO_2		Mole fraction (%)	6.022	4.358	0.001	20
O_2		Mole fraction (%)	13.292	5.299	0.03	20.393
SO_2 Conversion	Output	%	61.187	30.978	0	100

In this study, outliers were identified and removed from the dataset. Acknowledging the equal significance of all data elements, data normalization emerges as an essential data transformation technique employed to standardize all data elements within a predefined range [54]. This

normalization process not only expedites the learning phase within neural network backpropagation algorithms but also mitigates the potential dominance of attributes with larger ranges and/or values over attributes with smaller ranges and/or values [55]. In this work, min/max normalization was used to transfer the data to a range between 0 and 1. Furthermore, data transformation may serve as an effective strategy for reducing the dimensionality of input data, particularly when correlations exist among a series of variables. Principal component analysis (PCA) stands out as a widely adopted method in this regard [56]. Therefore, the use of PCA for data reduction was explored. In the current case study, given the low level of correlation among the input variables and especially for the large number of independent catalysts, the PCA did not result in a smaller number of input variables and was not found useful.

ML models generally assume datasets are complete, but missing values are often encountered during data collection [51,57]. To manage this, various data preprocessing techniques, including methods like autoassociative neural networks (AANNs), are employed for imputing missing data [58]. In this work, some missing data included pore size, pore volume, surface area, calcination time, temperature, and preparation methods. However, the effectiveness of AANNs is influenced by the correlation between variables [59]. In this case study, the AANN model struggled to accurately predict missing values due to the low correlation among features. As a result, the variables with missing values were removed from the main dataset, despite their recognized importance.

It is important to stress that to reach the final model, the procedure depicted in Figure 5.1 is not achieved in a single pass, but rather through multiple iterations. In this investigation, the main effort was devoted to the management and validation of the databank, especially determining which data points from the literature would be useful for inclusion in the databank. Initially, all literature data containing the necessary information were considered. However, some catalysts had a very limited number of instances and were removed from the database. Other papers had instances with identical inputs but with different conversions due to disparities in catalyst properties. In this case, the average conversion was calculated for similar data points, and this single average value was used to replace the redundant data points in the database.

According to Figure 5.1, the subsequent step following preprocessing entails in conducting hyperparameter tuning to identify the optimal parameters for the ANN model. Enhancing the

model's performance and minimizing errors necessitate the careful selection of the neural network's structure and associated parameters [60]. While the number of neurons in the input and output layers remains fixed to correspond with the model's inputs and outputs, respectively, the configuration of intermediate layers (hidden layers), as well as the number of neurons within each hidden layer, the number of epochs, the batch size, the learning rate, the optimizer, and the activation functions must be specified. These parameters play a pivotal role and can significantly influence model performance. Initially, models are constructed with parameters based on prior knowledge and expertise, with the initial weights assigned as small random values. In case of overfitting or underfitting, hyperparameter tuning techniques are employed to identify optimal parameters, aiming to minimize MSE for the validation set. These techniques often leverage grid search, random search, Bayesian optimization, and many other methodologies [60,61]. Ultimately, the selection of the final model involves a trade-off between model simplicity and MSE value, striking a balance to achieve satisfactory performance [60].

During the hyperparameter tuning phase, a series of models were trained using the 5-fold cross-validation and random search method to optimize parameters across two distinct ANN architectures. The first architecture consisted of three hidden layers, while the second had five hidden layers. Throughout this process, certain parameters were held constant, namely a sigmoid as the activation function, a learning rate of 0.01, 2000 epochs, and a batch size of 32. The following parameters were varied: the number of layers, the number of neurons within each layer, and the optimizer methods (Adam or Nadam). Numerous models were subsequently trained, with variations introduced by altering the fixed parameters.

As depicted in Figure 5.1, the subsequent procedural step is to partition the data records into three distinct subsets: the training set, the validation set, and the testing set. Different combinations were randomly tested, and the best combination was selected for the best model. In the subsequent phase depicted in Figure 5.1, the effectiveness of the constructed ANN model is assessed utilizing various criteria. Performance metrics serve to evaluate the goodness of the fit of the ML regression models [62]. The most common performance metrics are the root mean squared error (RMSE), mean absolute error (MAE), MSE, Pearson correlation coefficient, and coefficient of determination (R^2). In this study, MSE and R^2 are used.

5.3.2 Multi-objective optimization

In the field of engineering, ML is typically used in synergy with an optimization algorithm to determine the set of process input variables that optimizes specific objective functions in order to operate chemical processes under optimal conditions [63–66]. MOO problems are optimization problems that must satisfy multiple, often conflicting, objectives. A solution is defined as a vector of decision variables $X = \{x_1, x_2, x_3, \dots, x_n\}$ that optimizes the vector of objective functions $F(X) = \{f_1(X), f_2(X), f_3(X), \dots, f_M(X)\}$ within a feasible region of solutions which may be subjected to equality $h_g(X)$ and inequality $g_h(X)$ constraints. The decision variables are bounded between lower (x_i^{LL}) and upper (x_i^{UL}) limits, which constrain the search space for each variable [67–69]. This process is mathematically represented by Equation (5.4).

$$\begin{aligned} &\text{Min } F(X) = \{f_1(X), f_2(X), f_3(X), \dots, f_m(X), \dots, f_M(X)\}, \\ &\qquad\qquad m = 1, 2, 3, \dots, M \\ &\text{Subject to :} \\ &h_k(X) = 0, \quad g = 1, 2, 3, \dots, G \\ &g_h(X) \le 0, \quad h = 1, 2, 3, \dots, H \\ &x_i^{LL} \le x_i \le x_i^{UL}, \quad i = 1, 2, 3, \dots, n \end{aligned} \tag{5.4}$$

where M is the number of objectives. Each of the objective functions $f_m(X)$ can be either minimised or maximised [70]. A maximization problem can be converted into a minimization by negating the objective function [71]. With conflicting objectives, improving one objective may worsen the other objectives. The concept of Pareto optimality can address this issue [71,72] using the dominance relationship in an unbiased way to generate a large number of Pareto-optimal solutions rather than a single aggregate optimal solution. These solutions are referred to as non-dominated solutions. It is then up to a decision maker to select the best one among all Pareto-optimal solutions based on his/her preferences [67].

MOO techniques aim to identify a diverse set of solutions that reside on the Pareto front. These techniques, particularly the evolutionary MOO algorithms, are designed to circumscribe as accurately as possible the Pareto domain while overcoming challenges such as infeasible regions, local optima, and smooth regions of objective functions. Balancing computational cost and efficiency is crucial in addressing these challenges [73,74]. Various methods are available for solving MOO problems, with multi-objective evolutionary algorithms (MOEAs) being a popular choice

[72,75–77]. MOEAs utilize the dominance relationships in their quest to uncover Pareto-optimal solutions, a strategy shared by other optimization methods [78]. While MOEAs are effective in finding global optima and demonstrating robustness against noise, they have limitations, including potential redundancy and significant computational time [79].

NSGA-II is a popular method within MOEAs. It is specifically designed to tackle MOO problems and identify Pareto-optimal solutions. NSGA-II operates on the principles of elitism, an explicit diversity mechanism, and prioritization of non-dominated solutions. At each generation (t), the parent population (P_t) undergoes standard genetic operations such as selection, crossover, and mutation to generate the offspring population (Q_t). These two populations are then merged to form a new population (R_t) comprising $2N$ individuals. R_t is segmented into various non-dominance groups, and the new population of N fittest individuals is populated sequentially with points from the front with the least number of the domination score to fronts with higher domination scores. The selection of points in the last front necessary to reach N individuals is made in such a way as to maximize diversity using the crowding distance criterion. Figure 5.3 provides a visual representation of the process used at each generation [71].

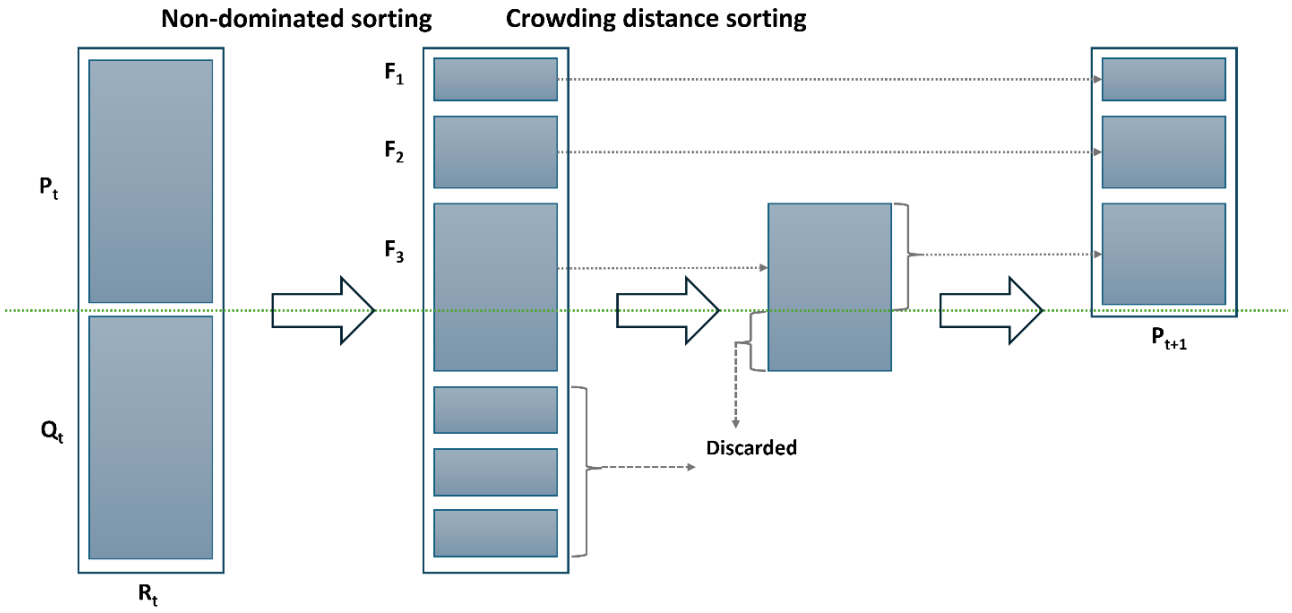


Figure 5.3. Schematic illustration of the procedure of NSGA-II.

In various chemical engineering applications, particularly in processes like SO_2 oxidation, the development of ML models presents an opportunity to optimize complex systems by leveraging

optimization algorithms. In this work, the optimization framework aims at maximizing conversion and productivity while minimizing catalyst cost, to strike a balance between these competing objectives.

The conversion (X) of SO_2, defined in Equation (5.5), is calculated from the ANN model.

$$X = \frac{F_{SO_2,in} - F_{SO_2,out}}{F_{SO_2,in}} \tag{5.5}$$

where $F_{SO_2,in}$ and $F_{SO_2,out}$ are the input and output molar flow rates of SO_2, respectively. The productivity is defined as the rate at which the moles of SO_2 are converted to SO_3 per unit time and per unit mass of the catalyst expressed by Equation (5.6).

$$P = \frac{y_{SO_2} X F_{in}}{W} \ (mol\ SO_3/kg_{cat} \cdot s) \tag{5.6}$$

where y_{SO_2} is the input SO_2 mole fraction, F_{in} is the total inlet molar flow rate, and W is the catalyst weight. This equation can be transformed in terms of the catalyst weight to the volumetric flow rate ratio under standard conditions (1 atm, 0°C) as well as expressing y_{SO_2} and X in percentage as per Table 5.1 and the input to the ANN model (Equation (5.7)).

$$P = 4.462 \times 10^{-6} \frac{y_{SO_2} X}{W/V} \ (mol\ SO_3/kg_{cat} \cdot s) \tag{5.7}$$

where W/V is the ratio of the weight of the catalyst bed and the volumetric flow rate in kg_{cat}·s/L. The third objective function is the cumulative cost of catalysts used in the process. This objective function involves a linear relationship between the mass fractions of individual catalysts and their corresponding costs (Equation (5.8)).

$$C = \sum_{i=1}^{r} c_i m_i \tag{5.8}$$

where r is the number of catalysts in the databank, c_i is the price of each catalyst and m_i is the mass fraction of each catalyst. The list of prices of the catalysts has been obtained using the Bloomberg databank and it is reported in Table 5.2 [80,81].

Table 5.2. Estimated price of each catalyst.

S.N.	Catalyst	Symbol	Price ($/kg)
1	Calcium	Ca	5
2	Cerium	Ce	4
3	Cesium	Cs	13000
4	Copper	Cu	8
5	Iron	Fe	0.5
6	Lanthanum	La	4
7	Magnesium	Mg	3.5
8	Manganese	Mn	2
9	Palladium	Pd	43500
10	Platinum	Pt	32000
11	Potassium	K	850
12	Sodium	Na	300
13	Tin	Sn	30
14	Vanadium	V	250

5.3.3 Multi-criteria decision-making

Once the Pareto domain is established based on domination principles (as depicted in Figure 5.1), the subsequent step involves ranking all non-dominated solutions. This ranking process requires the input of the decision-maker's preferences, which guide the selection of solutions according to the chosen ranking method. In this study, the NFM, a multi-criteria decision-making technique, is employed for ranking all Pareto-optimal solutions. The NFM incorporates the decision-maker's preferences through the utilization of four factors, which are outlined below [15].

The first factor involves assigning relative weights (W_m) to each criterion or objective 'm', indicating their importance. The sum of all relative weights must be equal to one. The second factor, known as the indifference threshold (Q_m), defines the difference in the values of objective 'm' between two solutions for which it is inconclusive to favour one solution over another for that specific objective. The third factor, the preference threshold (P_m), represents the threshold beyond which the difference in objective 'm' values between two solutions warrants a preference for the solution with the better objective value. Lastly, the veto threshold (V_m) is used to rule out the selection of one solution over another if the difference between their objective 'm' values exceeds

a certain threshold. The latter implies that even if a solution excels in other criteria, it may be disregarded based on a specific objective. The three thresholds are established individually for each criterion, in such a manner that:

$$0 \leq Q_m \leq P_m \leq V_m \tag{5.9}$$

In the following, we will briefly explain the NFM algorithm.

In Equations (5.10-5.13) are used to calculate the various indices: the difference $\Delta_m[i,j]$ between the objective m of solutions i and j, the individual concordance index $c_m[i,j]$, the global concordance index $C[i,j]$, and the discordance index $D_m[i,j]$. These indices are determined through a pairwise comparison. When the difference between values is less than the indifference threshold for a given criterion, the individual concordance index is assigned a value of unity. Within the range spanning from the indifference to the preference thresholds, the index diminishes linearly from 1 to 0. If the difference surpasses the preference threshold for a given criterion, the concordance index is set to zero. When the difference between values is below the preference threshold, the discordance index is assigned a value of 0. Within the range encompassing the preference and veto thresholds, the index exhibits a linear progression from 0 to 1. In instances where the difference exceeds the veto threshold, the discordance index is fixed at a value of 1.

$$\Delta_m[i,j] = F_m(j) - F_m(i) \qquad \begin{cases} i \in [1,N] \\ j \in [1,N] \\ m \in [1,M] \end{cases} \tag{5.10}$$

$$c_m[i,j] = \begin{cases} 1 & if \Delta_m[i,j] \leq Q_m \\ \dfrac{P_m - \Delta_m[i,j]}{P_m - Q_m} & if\ Q_m < \Delta_m[i,j] \leq P_m \\ 0 & if\ \Delta_m[i,j] > P_m \end{cases} \tag{5.11}$$

$$C[i,j] = \sum_{m=1}^{M} W_m c_m[i,j] \qquad \begin{cases} i \in [1,N] \\ j \in [1,N] \end{cases} \tag{5.12}$$

$$D_m[i,j] = \begin{cases} 0 & if \Delta_m[i,j] \leq P_m \\ \dfrac{\Delta_m[i,j] - P_m}{V_m - P_m} & if\ P_m < \Delta_m[i,j] \leq V_m \\ 1 & if\ \Delta_m[i,j] > V_m \end{cases} \tag{5.13}$$

Once the global concordance and discordance indices have been calculated, the process moves on to the comparative analysis of each pair of Pareto-optimal solutions. This evaluation involves calculating each element of the outranking matrix, denoted as $\sigma[i,j]$ using Equation (5.14). Essentially, the outranking matrix serves as a tool to determine which solutions outperform others based on their relative performance.

$$\sigma[i,j] = C[i,j]\left(\prod_{m=1}^{M}[1-(D_m[i,j])^3]\right) \qquad \begin{cases} i \in [1,N] \\ j \in [1,N] \end{cases} \tag{5.14}$$

In order to determine the best solution, the score of each solution i is calculated using Equation (5.15). The first term assesses how well solution i performs compared to all the other solutions in the Pareto domain. The second term evaluates the performance of all the other solutions relative to solution i. After computing the ranking score for each solution, they are sorted in descending order. This calculation is the final step in the solution evaluation process. The solution with the highest score is considered the best.

$$\sigma_i = \sum_{j=1}^{M}\sigma[i,j] - \sum_{j=1}^{M}\sigma[j,i] \tag{5.15}$$

5.4 Results and discussion

5.4.1 Final ANN model

As previously explained, various ANN models were trained and integrated into the MOO phase to circumscribe the Pareto domain and propose the optimal solution for catalytic SO_2 conversion. Initially, data were gathered from diverse sources, and a subset of variables was selected. The first version of our database was formed by extracting data from 32 papers. Then, the preprocessing stage was performed to increase the integrity and quality of the data. After finalizing the input and output parameters, several ANN model structures were randomly selected to assess their ability to predict conversion. Simultaneously exploring the structure of ANNs, the hyperparameter tuning was performed to enhance the performance of the ANNs and reduce errors. This procedure is guided by experience in developing ANNs, but still involves a degree of trial and error. Once the best-performing ANN model was identified, it was used in the MOO procedure, along with the solution ranking algorithm as per Figure 5.1, to determine the optimal solution.

Multiple ANN models were trained using the updated hyperparameters obtained from the hyperparameter tuning process. The architecture of the final ANN model, as detailed in Table 5.3, consists of three hidden layers, each accommodating 22 neurons. The model employs the Adam optimizer with a learning rate set at 0.0005. Additionally, a batch size of 32 and an epoch value of 50000 were selected. The dataset was partitioned into the training, validation, and testing sets in an 80:10:10 ratio, denoted as Tr/V/Te.

Figure 5.4 provides an overview of the performance of the final ANN model. As shown in Figure 5.4A, the MSE in the prediction of the normalized conversion for both the training and validation datasets decreases sharply at the beginning and then gradually converges to its final value as the number of epochs exceeds 10000, with the final weights and biases being recorded at 36782 epochs, which were achieved with the early stopping technique. The parity plot of the conversion for the training data is displayed in Figure 5.4B, which shows an MSE of 0.0025 and an R^2 value of 0.9745. Figure 5.4C and Figure 5.4D further illustrate the performance of the ANN model on the validation and testing datasets, respectively. The model's performance on these datasets is comparable to that of the training data, indicating its consistency. This uniformity across different datasets implies that the model exhibits a low error in the prediction of the conversion and can effectively generalize to unseen testing data, thereby mitigating any concerns regarding overfitting.

Table 5.3. Structure of the final ANN model.

Optimizer	Learning rate	Hidden layers with neurons	Batch size	Epoch	Tr/V/Te size
Adam	0.0005	3 HL: 22, 22, 22	32	36782	80/10/10

To comprehensively assess the capacity of the ANN model to surpass conventional statistical and regression techniques, its performance was benchmarked against linear regression, ridge regression, and polynomial regression models on a 10% testing dataset. The comparative results, presented in Table 5.4, highlight the superior performance of the ANN model in all evaluated metrics. Notably, the ANN model exhibits a marked improvement over its counterparts, as evidenced by significantly lower MSE and higher R^2 values. This analysis underscores the ANN model's robust predictive capability, achieving both greater accuracy and better generalizability compared to the tested regression approaches.

Table 5.4. Comparative performance of the final ANN and traditional regression models on the testing data.

Performance criteria	Linear regression	Ridge regression	Polynomial regression	ANN
MSE	0.0461	0.0463	0.0128	0.0031
R^2	0.4970	0.4939	0.8597	0.9641

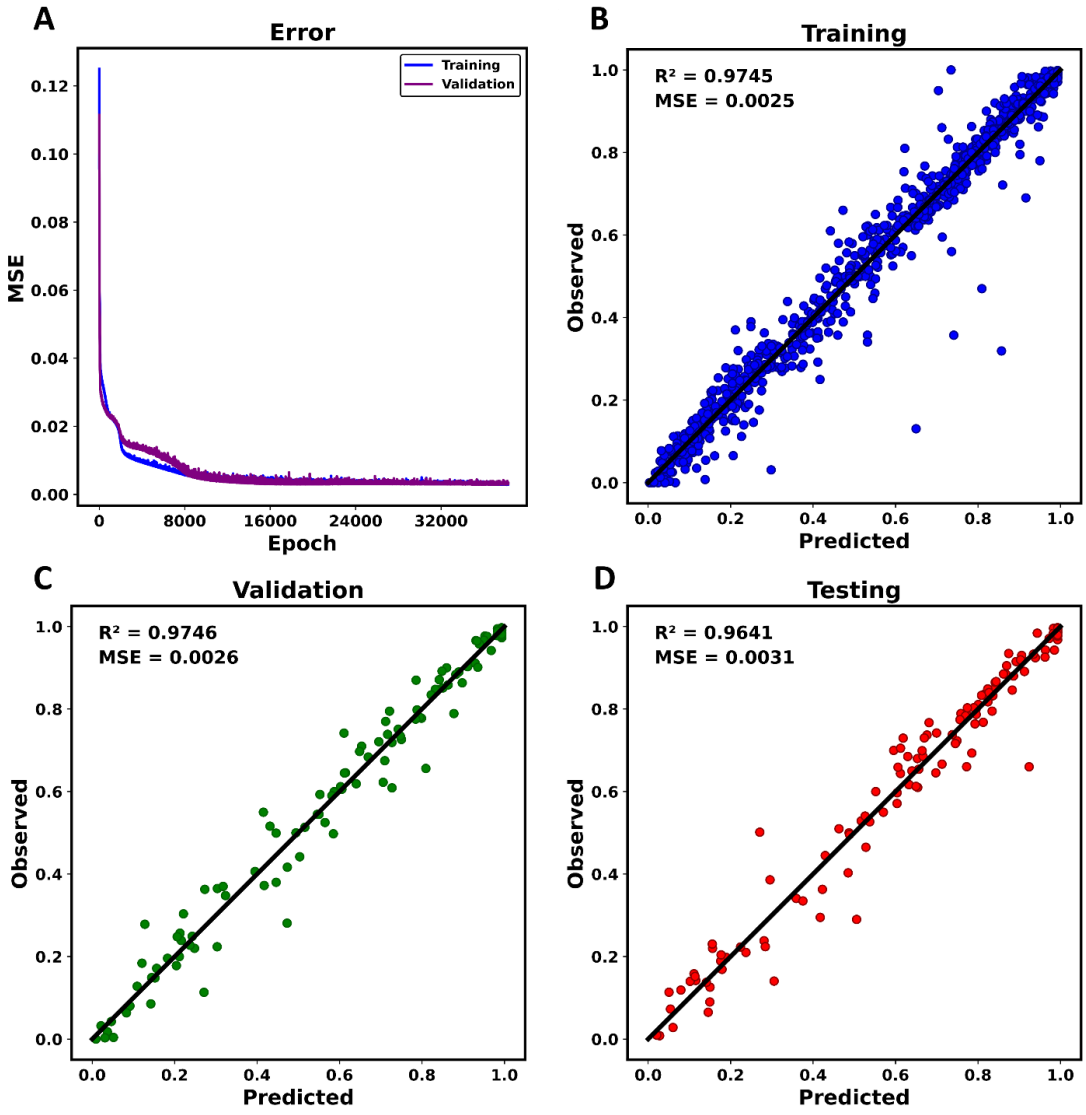


Figure 5.4. Performance of the final ANN model: (A) Training and validation MSE as a function of the number of epochs, and the parity plots of the conversion for the (B) training dataset, (C) validation dataset, and (D) testing dataset.

The dataset, as mentioned earlier, comprises information about two specific applications of SO_2 catalytic conversion: FGD and H_2SO_4 production. Both processes convert SO_2 into SO_3 using different metal catalysts supported by various materials. A key distinguishing factor between these applications within the dataset is the mole fraction of SO_2 in the feed. Data points with a low SO_2 concentration (less than 1%) correspond to FGD, whereas those with a higher concentration (above 7%) are associated with H_2SO_4 production.

This study primarily focuses on optimizing the production of H_2SO_4, given its significant economic importance as one of the most widely produced chemicals worldwide. However, exploring optimal conditions for FGD applications is also achievable since the data points in the dataset are comprised of both applications. Consequently, to tailor the problem formulation specifically for H_2SO_4 production, the mole fraction of SO_2 was fixed at 10%. Additionally, to mimic industrial practice, the oxygen concentration was set at 11%.

5.4.2 Case study 1: Vanadium-potassium catalyst

The MOO problem allows for the exploration of various combinations of metals, either individually or in conjunction, supported by the three available support materials. However, given the impracticality of examining all possible combinations, a prominent bimetallic catalyst featuring a mixture of vanadium and potassium was first selected for further investigation in the MOO problem. This selection was motivated by the prevalence of reported instances involving these two metals, which collectively constitute approximately 35% of the entire databank.

Table 5.5 presents the fixed and variable parameters that have been set for the MOO problem. The fraction of the active phase was selected to be optimized along with operating parameters such as temperature, pressure, and the W/V. It is important to note that each variable was constrained within its range of gathered data in the databank as reported in Table 5.1. The temperature range was set between 310-622°C, which is the range where both vanadium and potassium coexist. The pressure was kept constant at 1 atm, while the W/V was allowed to vary between 0.02-1.00 kg_{cat}·s/L. The lower limit of 0.02 for W/V corresponds to the smallest W/V ratio observed when both vanadium and potassium are present in the dataset. The support material was also specified to be silica, as this material is the only one in the dataset that was used for both vanadium and potassium-based catalysts.

Table 5.5. Values of the decision parameters for the catalytic combination of vanadium and potassium.

Decision Parameters	State	Value
Components	Fixed	V and K
Active phase fraction	Variable	V = 2.0-8.0 wt% K = 2.0-18.42 wt%
Support	Fixed	SiO_2
Temperature	Variable	310-622°C
Pressure	Fixed	1 atm
Catalyst W/V	Variable	0.02-1.00 kg_{cat}·s/L
SO_2 fraction	Fixed	0.10
O_2 fraction	Fixed	0.11

Process variables that were not involved in this specific case study were set to zero. The size of the population and the number of generations of NSGA-II were respectively set at 1000 and 10000. Other NSGA-II factors depicted in Figure 5.1 were carefully adjusted to circumscribe a well-defined Pareto front. The selection of these parameters guides the optimization process within the MOO framework, facilitating the exploration of diverse solutions while ensuring robustness and efficiency in identifying non-dominated solutions.

The next phase involved the ranking of all Pareto-optimal solutions by the NFM, as shown in Figure 5.1. Table 5.6 outlines the set of parameters of the NFM used in this study to rank all solutions of the Pareto domain. The relative weights for the conversion and productivity were set at 0.45, whereas the weight for the cost was set at 0.1. A higher significance was placed on the first two objectives. This strategic emphasis underscores the objective of having both high conversion and high productivity, albeit potentially resulting in higher costs. Furthermore, in the NFM ranking algorithm, the specification of the indifference, preference, and veto thresholds often involves some form of expert judgment or statistical analysis. The selection of these thresholds is

a critical aspect of the NFM ranking algorithm, as they directly influence the resulting ranking of all Pareto-optimal solutions.

Table 5.6. NFM parameters for vanadium and potassium case study.

Parameters	Conversion	Productivity	Cost
W	0.45	0.45	0.1
Q	2	0.01	50
P	5	0.03	100
V	10	0.1	200

Figure 5.5 provides a visual representation of the Pareto domain, highlighting the interaction among the three objective functions. In Figure 5, the three-dimensional Pareto domain is represented by three two-dimensional projections. All Pareto-optimal solutions are partitioned based on their ranking using different colours as per the legend of Figure 5.5B. Figure 5.5A examines the relationship between conversion and productivity. Conversion is defined as the extent to which SO_2 is transformed into SO_3 during the catalytic process, while productivity quantifies the speed at which this conversion is achieved. A preliminary analysis of the Pareto domain reveals two distinct phenomena. Firstly, both conversion and productivity generally increase simultaneously. Second, a clear conflict among these objectives is also revealed when examining the ranking partitions, where a trade-off situation is observed with the enhancement of conversion, which inevitably leads to a decrease in productivity. This inverse correlation is more visually apparent at the right edge of the Pareto domain, with the highest conversion percentages coinciding with the lowest productivity values. The highest-ranked solution, namely the champion, is located at the highest productivity, accompanied by a relatively small compromise in the conversion percentage.

Figure 5.5B explores the relationship between the percentage conversion and the cost of the catalyst of Pareto-optimal solutions. The NFM relative weight for the cost was 0.1 compared to 0.45 for the other two objectives, as they were seen as more significant. It is therefore not surprising to find the highest-ranked solution at a higher cost, which implies a larger amount of catalyst to favour higher conversion and productivity. The plot of Figure 5.5B clearly shows the conflict that exists between the desire to maximize conversion and minimize the cost. Figure 5.5C further

investigates the relationship between cost and productivity. Concentrating on the right edge of the Pareto domain, it is clear that minimizing the cost results in lower productivity. Considering the three objectives and the relative weight for each, the champion solution was located at the highest productivity (0.193 mol SO_3/kg_{cat}·s), relatively high conversion (86.3%), and a high cost (163.3$/kg), resulting from the greater priority attributed to the conversion and productivity.

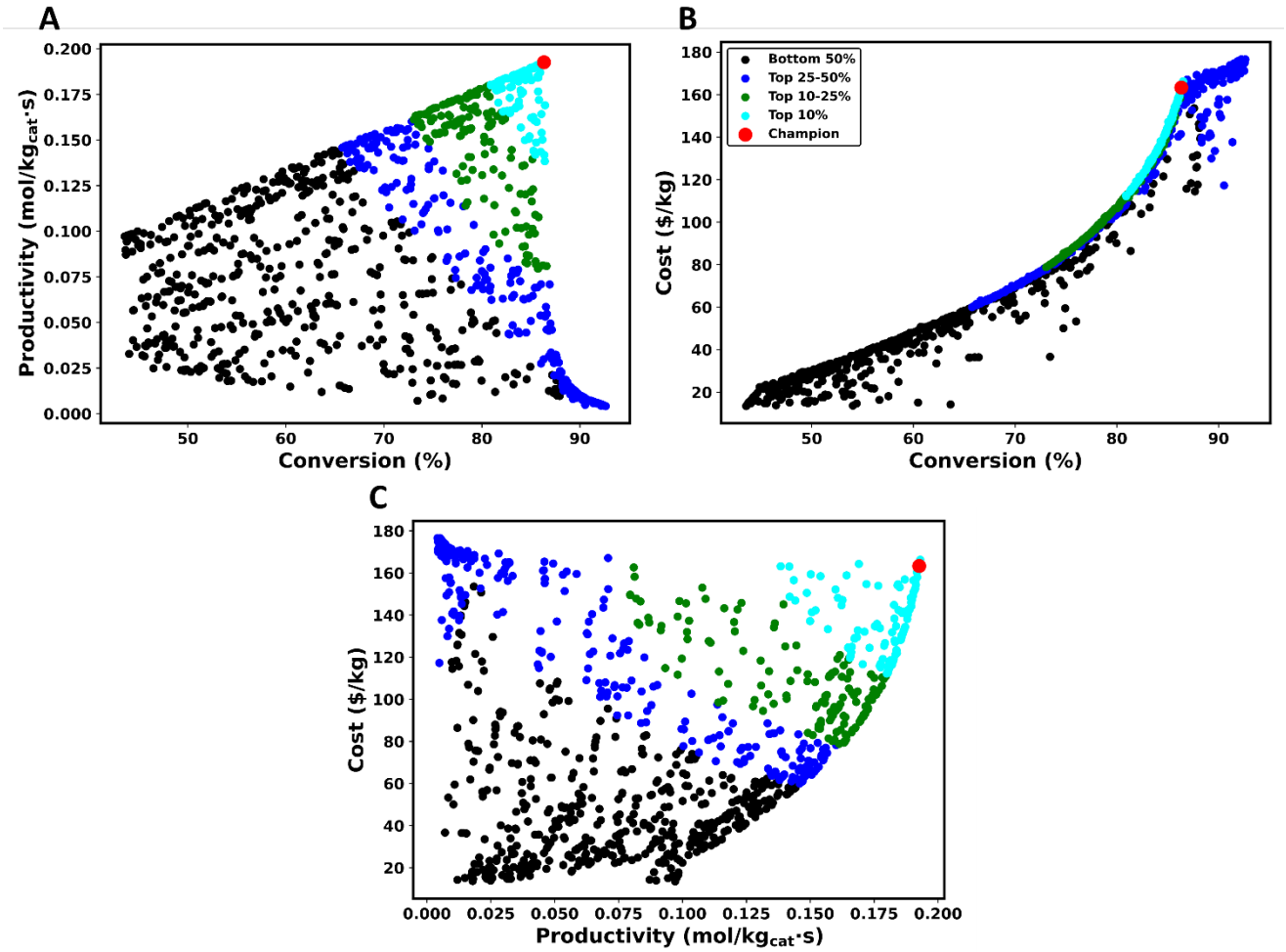


Figure 5.5. Pareto domain: (A) conversion vs productivity, (B) conversion vs cost, and (C) cost vs productivity for vanadium and potassium case study.

Our attention is now turned to the decision space. In this case, the composition of the selected catalyst, W/V, and the reaction temperature were the process variables that were allowed to vary. Figure 5.6 presents the plots of the four decision variables associated with all Pareto-optimal solutions. Figure 5.6A presents a plot depicting the active phase fraction for vanadium and potassium. Higher amounts of these metals are associated with increased percentage conversion,

productivity, and cost. The region containing the highly-ranked Pareto-optimal solutions is located at a high fraction of potassium (17%) and a fraction of vanadium between 3 and 4%. For lower fractions of potassium, the fraction of vanadium is located at its lower value. The best Pareto-optimal solution corresponds to vanadium and potassium mass fractions of 3.8% and 18%, respectively.

Figure 5.6B provides an overview of the other two decision variables: the W/V ratio and the temperature. Near the strike temperature, in the vicinity of 400-425°C, there is a sharp increase in the W/V ratio when decreasing temperature, which implies a larger amount of catalyst is required to achieve higher conversion and productivity. In addition, the rate of reaction is lower at a lower temperature. Above this limiting temperature, the W/V ratio hovers around its bounded lower limit. It is important to recall that the SO_2 to SO_3 reaction is a reversible exothermic reaction, such that increasing the temperature eventually leads to a decrease in the achievable conversion and productivity. It is therefore not surprising to observe that the highly-ranked Pareto-optimal solutions are located in a very narrow range of temperatures just slightly above the strike temperature. The champion is obtained with a W/V value of 0.02 $kg_{cat}\cdot s/L$ and a temperature of 436°C, consistent with the reported strike temperature for the SO_2 reaction. This finding underscores the critical balance required between the amount of catalyst and the reaction temperature needed to maximize conversion and productivity, while also minimizing the negative impact of the temperature increase on the reaction equilibrium concentration.

5.4.3 Case study 2: Platinum catalyst

An additional case study was undertaken to investigate the catalyst platinum and determine the optimal active phase and operating conditions. The selection of platinum for this investigation is motivated by two main factors. Firstly, platinum is renowned for its potent active sites and its high catalytic efficiency [82,83]. Secondly, platinum-related data constitutes approximately 20% of the information gathered in the databank. Akin to the preceding case study, the objective was to determine the optimal composition and operating parameters for platinum supported on silica, under a fixed pressure of 1 atm, and fixed mole fractions of SO_2 and O_2 at 10% and 11% respectively. Therefore, the variables under consideration are the active phase fraction of platinum, temperature, and W/V.

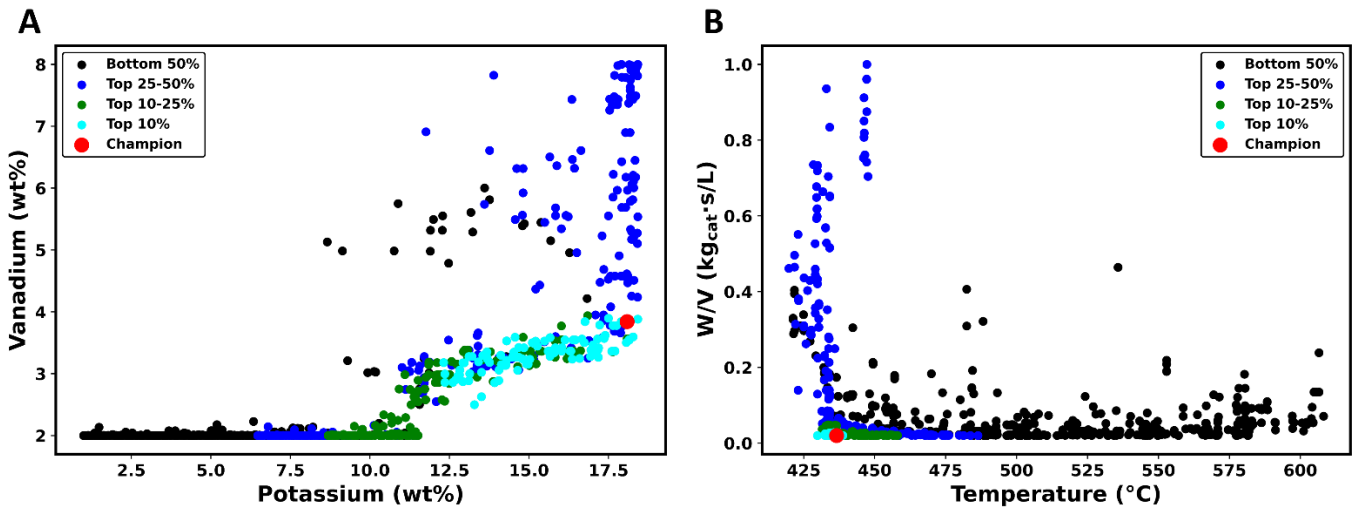


Figure 5.6. Decision space: (A) vanadium wt% vs potassium wt% and (B) temperature vs W/V for vanadium and potassium case study.

The same ANN model employed in optimizing other metals was also applied to platinum, given that it was trained on all available metals. During the MOO phase, the lower and upper bounds of the decision variables were set based on the minimum and maximum values observed for each variable when the platinum active phase was non-zero. This approach ensures that the extrapolation does not exceed the dataset's range for each specific metal. The remaining parameters of NSGA-II were kept consistent with the previous case study. However, the NFM parameters for this case study needed to be adjusted due to differing objective ranges compared to vanadium and potassium. This adjustment was necessary because of the platinum's wider operating range and considerably higher cost compared to vanadium and potassium. Table 5.7 outlines the NFM parameters used in the platinum case study. Notably, the emphasis is placed on conversion and productivity over cost, as depicted in Table 5.6.

Table 5.7. NFM parameters for platinum case study.

Parameters	Conversion	Productivity	Cost
W	0.45	0.45	0.1
Q	1	0.2	300
P	5	0.4	700
V	10	1.0	1500

Figure 5.7 presents the Pareto domain for the platinum case study, focusing on the same three objectives previously considered: conversion, productivity, and catalyst cost. In Figure 5.7A, a consistent pattern emerges, akin to the vanadium-potassium case study, where two similar trends are observed. Examining the oblique left edge of the Pareto domain, it is clear that higher conversion is accompanied by higher productivity to a certain point. This trend makes sense as the definition of productivity embeds the conversion. However, examining the coloured bands of similarly-ranked Pareto-optimal solutions and the right edge of the Pareto domain shows a clear trend of the decrease in productivity as the conversion continues to increase. This trend is logical since to achieve very high conversion, the flowing gas must spend significantly more time in the reactor and/or a greater amount of platinum catalyst must be used. As a result, the productivity decreases as it includes both the reaction time and the amount of catalyst. Indeed, it is possible to achieve near 100% conversion, but at the expense of reduced productivity. Additionally, Figure 5.7B corroborates the notion that increased conversion demands a larger amount of the catalyst's active phase, consequently increasing the overall cost. Figure 5.7C elucidates the interplay between cost and productivity. Notably, it reveals that achieving high productivity does not inherently entail incurring exorbitant costs. This suggests that high productivity levels can be attained without substantial increases in cost, particularly when the amount of platinum used is minimal. This highlights the feasibility of optimizing productivity while mitigating costs. The highest-ranked Pareto-optimal solution is achieved at a percentage conversion of 90%, a productivity of 1.34 mol $SO_3/kg_{cat}\cdot s$, and a cost of \$294/kg. It is important to recognize that this result does not simulate the steady-state process involved in H_2SO_4 production, where the goal is to achieve nearly 100% conversion. This high conversion is attained using multiple packed beds with intermediate heat exchangers, where the temperature and conversion vary continuously along the packed bed. The results of the current investigation are for a constant temperature and steady-state homogeneous process.

Figure 5.8 depicts the decision space corresponding to the Pareto domain illustrated in Figure 5.7. This analysis offers valuable insights into the relationship between temperature, the platinum active phase, and W/V ratio. In Figure 5.8A, the temperature variation is juxtaposed with changes in the platinum active phase. Initially, temperature bounds were set within the range of 204-780°C. However, the MOO process has identified Pareto-optimal solutions in a temperature range

spanning 430-570°C. Figure 5.8A can be partitioned into three distinct sections based on the platinum active phase concentration.

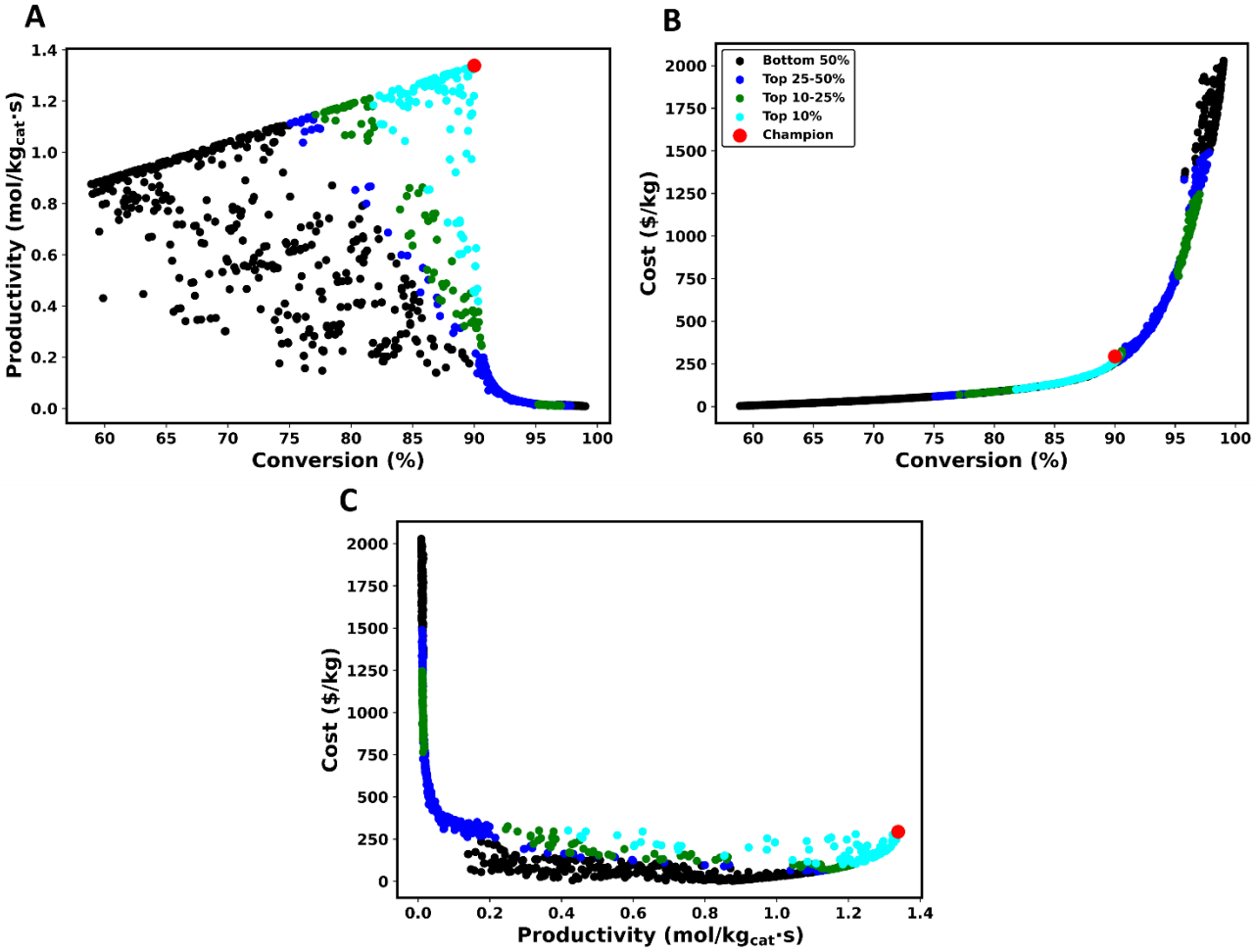


Figure 5.7. Pareto domain: (A) conversion vs productivity, (B) conversion vs cost, and (C) cost vs productivity for the platinum case study.

The first portion of the plot of Figure 5.8A encompasses instances where the platinum concentration remains below 1%. Within this range, increasing the platinum active phase content coincides with enhanced conversion and productivity, particularly at lower temperatures. Remarkably, this region aligns with the lower segment of Figure 5.8B representing the W/V ratio. This juxtaposition indicates that within this zone, optimal productivity levels alongside a 90% conversion are attainable when the temperature is on the low side. In the second segment of Figure 5.8A, the temperature remains relatively constant as the platinum active phase concentration increases. This phenomenon is mirrored in Figure 5.8B, where at low and nearly constant

temperature, the W/V ratio exhibits an upward trend. Consequently, this trend corresponds to an increasing percentage conversion, and at the same time, there is a trade-off in terms of reduced productivity.

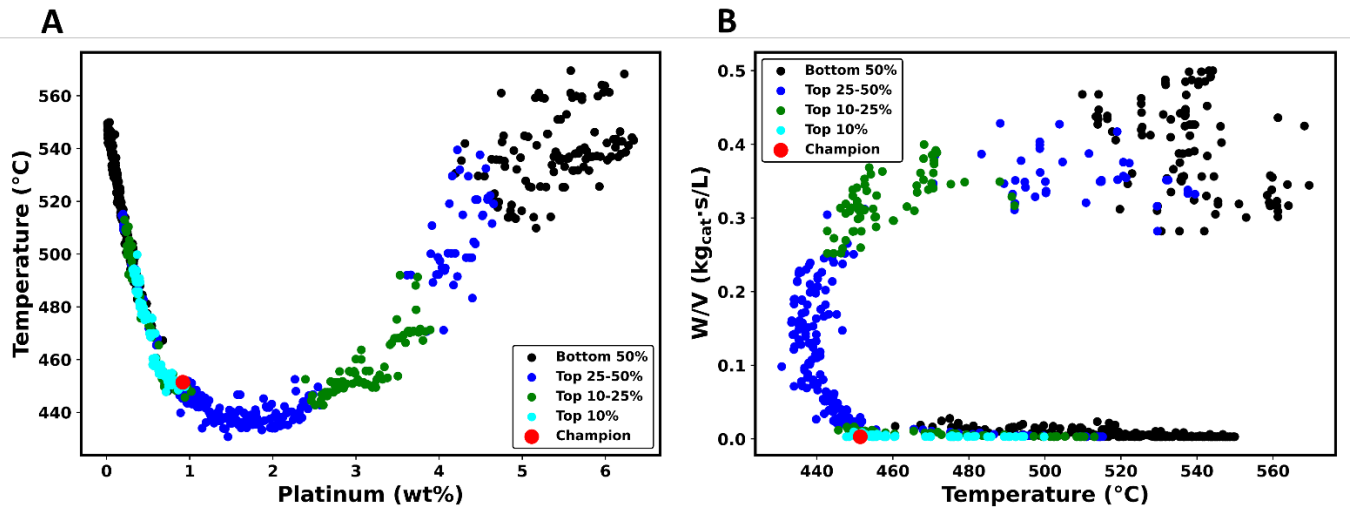


Figure 5.8. Decision space: (A) platinum wt% vs temperature and (B) temperature vs W/V for the platinum case study.

The third segment of Figure 5.8A illustrates a simultaneous increase in the temperature and the amount of platinum active phase. This trend is also evident in Figure 5.8B, particularly at higher values of the W/V ratio. With an increase in the amount of the platinum active phase, in the temperature, and the W/V ratio, the productivity decreases while the percentage conversion and the catalyst cost surge to peak levels. A delicate balance between conversion efficiency and overall productivity must be managed. While achieving peak percentage conversion, care must be exercised to mitigate costs.

The methodologies developed in this study offer significant real-world applications across various industries, facilitating optimized catalyst design and operating conditions for enhanced efficiency, cost-effectiveness, and sustainability. In the H_2SO_4 production industry, this approach can be used to improve conversion and productivity while simultaneously minimizing costs and resource consumption. By applying the optimization techniques, designers can precisely tailor catalyst compositions and reaction conditions, thereby achieving higher yields and reduced energy requirements. Similarly, in the field of FDG, the method can be utilized to meet stringent environmental regulations by optimizing catalysts for SO_2 removal, ensuring both cost-efficiency

and compliance with environmental standards. In the petrochemical sector, the framework can enhance the refinement of various products, increasing the selectivity and yield of desired compounds while lowering production costs. In hydrogen production, the same optimization process can balance energy consumption and catalyst performance, extending the lifespan of catalysts while maintaining high efficiency. Moreover, the methods are applicable to emerging sustainable technologies, such as carbon capture and utilization, and biomass conversion, where optimizing catalysts can significantly improve reaction outcomes and reduce waste. By integrating ANN predictions with MOO, industries can accelerate the design and testing of catalysts, reducing the need for costly and time-consuming experimental work. This integrated approach not only drives innovation but also provides a pathway to solving complex challenges in a range of sectors, from chemical production to energy generation. Overall, the insights gained from this study can be leveraged to foster sustainable, efficient, and cost-effective operations, benefiting industries seeking to improve both their economic outcomes and environmental performance.

5.5 Conclusion

The catalytic conversion of SO_2 is critical in various industrial applications, including FGD and H_2SO_4 production. This study delves into optimizing this conversion process using a multifaceted approach that incorporates ML, MOO, and decision-making techniques. Our efforts shed light on the complex interplay between various parameters and objectives, providing valuable insights into enhancing conversion efficiency while balancing productivity and cost considerations, even for a relatively simple reactive system. In this study, data were gathered from literature sources and underwent preprocessing to improve their quality. The hyperparameter tuning was then conducted to identify the best AAN model to predict SO_2 conversion with the lowest error. Using this AAN model, three objectives (conversion, productivity, and cost of the catalyst) were formulated, and the NSGA-II optimization algorithm was used to circumscribe the Pareto domain. The Pareto-optimal solutions were subsequently ranked using the NFM.

Two case studies were considered for H_2SO_4 production: one using a vanadium-potassium catalyst and the other using platinum as the catalyst. For each case study, the same AAN model, obtained to predict conversion, was utilized, and along with other objectives, the Pareto domain and optimal decision variables were illustrated. It was observed that for both case studies, conversion and productivity have trade-offs; increasing one objective leads to a decrease in the

other. These trade-offs also impact the cost of the catalysts. The analysis underscored the critical role of key parameters, such as catalyst composition, temperature, and W/V ratio, in shaping process performance. Optimal solutions often require careful adjustments to these parameters, reflecting the delicate balance needed to achieve desired conversion levels without compromising productivity or incurring excessive costs. The optimal findings for the first case study revealed a conversion of 86.3%, a productivity of 0.193 mol $SO_3/kg_{cat}\cdot s$, and a cost of 163.3$/kg with vanadium and potassium mass fractions of 3.8% and 18%, respectively. These outcomes were achieved with an optimal W/V value of 0.02 $kg_{cat}\cdot s/L$ and a temperature of 436°C. For the second case study, a conversion of 90%, a productivity of 1.34 mol $SO_3/kg_{cat}\cdot s$, and a cost of $294/kg were obtained. These results were achieved with 1% platinum wt% and with a W/V value of 0.003 $kg_{cat}\cdot s/L$ and a temperature of 451°C.

In conclusion, this study offers a comprehensive exploration of the catalytic conversion of SO_2, integrating ML, MOO, and decision-making techniques to optimize process performance. By leveraging advanced methodologies and rigorous analysis, we elucidated key insights into the intricate dynamics governing conversion efficiency, productivity, and cost. Our findings pave the way for informed decision-making and strategic optimization initiatives aimed at enhancing the sustainability and efficiency of catalytic conversion processes in diverse industrial sectors.

5.6 References

1. Islam, A., Teo, S. H., Ng, C. H., Taufiq-Yap, Y. H., Choong, S. Y. T., & Awual, M. R. (2022). Progress in recent sustainable materials for greenhouse gas (NO_x and SO_x) emission mitigation. *Progress in Materials Science*, *132*, 101033. https://doi.org/10.1016/j.pmatsci.2022.101033
2. Lowe, E. A., & Evans, L. K. (1995). Industrial ecology and industrial ecosystems. *Journal of Cleaner Production*, *3*(1–2), 47–53. https://doi.org/10.1016/0959-6526(95)00045-G
3. Jacobson, M. Z. (2012). *Air pollution and global warming: History, science, and solutions*. Cambridge University Press.
4. Srivastava, R. K., Jozewicz, W., & Singer, C. (2001). SO_2 scrubbing technologies: A review. *Environmental Progress*, *20*(4), 219–228. https://doi.org/https://doi.org/10.1002/ep.670200410

5. Gound, T. U., Ramachandran, V., & Kulkarni, S. (2014). Various methods to reduce SO_2 emission - A review. *International Journal of Ethics In Engineering & Management Education*, *1*(1), 1–6.
6. Hanif, M. A., Ibrahim, N., & Abdul Jalil, A. (2020). Sulfur dioxide removal: An overview of regenerative flue gas desulfurization and factors affecting desulfurization capacity and sorbent regeneration. *Environmental Science and Pollution Research*, *27*(22), 27515–27540. https://doi.org/https://doi.org/10.1007/s11356-020-09191-4
7. Lum, M. M. X., Ng, K. H., Lai, S. Y., Mohamed, A. R., Alsultan, A. G., Taufiq-Yap, Y. H., Koh, M. K., Mohamed, M. A., Vo, D.-V. N., & Subramaniam, M. (2023). Sulfur dioxide catalytic reduction for environmental sustainability and circular economy: A review. *Process Safety and Environmental Protection*, *176*, 580–604. https://doi.org/https://doi.org/10.1016/j.psep.2023.06.035
8. Karatepe, N. (2000). A comparison of flue gas desulfurization processes. *Energy Sources*, *22*(3), 197–206. https://doi.org/10.1080/00908310050013983
9. Ng, K. H., Lai, S. Y., Jamaludin, N. F. M., & Mohamed, A. R. (2022). A review on dry-based and wet-based catalytic sulphur dioxide (SO_2) reduction technologies. *Journal of Hazardous Materials*, *423*, 127061. https://doi.org/10.1016/j.jhazmat.2021.127061
10. del Valle-Zermeño, R., Formosa, J., & Chimenos, J. M. (2015). Wet flue gas desulfurization using alkaline agents. *Reviews in Chemical Engineering*, *31*(4), 303–327. https://doi.org/10.1515/revce-2015-0002
11. Duan, R., Chen, W., Chen, Z., Gu, J., Dong, Z., He, B., Liu, L., & Wang, X. (2022). Mechanistic and Experimental Study of the Cu_xO@C Nanocomposite Derived from $Cu_3(BTC)_2$ for SO_2 Removal. *Catalysts*, *12*(7), 689. https://doi.org/10.3390/catal12070689
12. Chen, R., Zhang, T., Guo, Y., Wang, J., Wei, J., & Yu, Q. (2021). Recent advances in simultaneous removal of SO_2 and NO_x from exhaust gases: Removal process, mechanism and kinetics. *Chemical Engineering Journal*, *420*, 127588. https://doi.org/10.1016/j.cej.2020.127588
13. Davenport, W. G., King, M. J., Rogers, B., & Weissenberger, A. (2006). Sulphuric acid manufacture. *Southern African Pyrometallurgy*, *1*(1–16).
14. Sidana, S. (2016). Sulfuric acid production using catalytic oxidation. *International Journal of Engineering and Technical Research*, *6*, 65–68.

15. Zaker, M. R., Fauteux-Lefebvre, C., & Thibault, J. (2021). Modelling and Multi-Objective Optimization of the Sulphur Dioxide Oxidation Process. *Processes*, *9*(6), 1072. https://doi.org/10.3390/pr9061072
16. García-Labiano, F., Diego, L. F. de, Cabello, A., Gayán, P., Abad, A., Adánez, J., & Sprachmann, G. (2016). Sulphuric acid production via Chemical Looping Combustion of elemental sulphur. *Applied Energy*, *178*, 736–745. https://doi.org/10.1016/j.apenergy.2016.06.110
17. Vangapally, N., Penki, T. R., Elias, Y., Muduli, S., Maddukuri, S., Luski, S., Aurbach, D., & Martha, S. K. (2023). Lead-acid batteries and lead–carbon hybrid systems: A review. *Journal of Power Sources*, *579*, 233312. https://doi.org/https://doi.org/10.1016/j.jpowsour.2023.233312
18. Kutney, G. (2023). *Sulfur: History, technology, applications and industry*. ChemTec Publishing, Elsevier.
19. Moeller, W., & Winkler, K. (1968). The double contact process for sulfuric acid production. *Journal of the Air Pollution Control Association*, *18*(5), 324–325. https://doi.org/https://doi.org/10.1080/00022470.1968.10469134
20. Kiss, A. A., Bildea, C. S., & Grievink, J. (2010). Dynamic modeling and process optimization of an industrial sulfuric acid plant. *Chemical Engineering Journal*, *158*(2), 241–249. https://doi.org/https://doi.org/10.1016/j.cej.2010.01.023
21. Oni, A. O., Fadare, D. A., Sharma, S., & Rangaiah, G. P. (2018). Multi-objective optimisation of a double contact double absorption sulphuric acid plant for cleaner operation. *Journal of Cleaner Production*, *181*, 652–662. https://doi.org/https://doi.org/10.1016/j.jclepro.2018.01.239
22. Ashar, N. G., & Golwalkar, K. R. (2013). A practical guide to the manufacture of sulfuric acid, oleums, and sulfonating agents. *Springer*.
23. Tejeda-Iglesias, M., Szuba, J., Koniuch, R., & Ricardez-Sandoval, L. (2018). Optimization and modeling of an industrial-scale sulfuric acid plant under uncertainty. *Industrial & Engineering Chemistry Research*, *57*(24), 8253–8266. https://doi.org/10.1021/acs.iecr.8b00785

24. Loskyll, J., Stöwe, K., & Maier, W. F. (2013). Search for New Catalysts for the Oxidation of SO_2. *ACS Combinatorial Science*, *15*(9), 464–474. https://doi.org/https://doi.org/10.1021/co400023j
25. Wang, X., Wang, A., Li, N., Wang, X., Liu, Z., & Zhang, T. (2006). Catalytic reduction of SO_2 with CO over supported iron catalysts. *Industrial & Engineering Chemistry Research*, *45*(13), 4582–4588. https://doi.org/https://doi.org/10.1021/ie0600947
26. Wilburn, M. S., & Epling, W. S. (2018). Formation and decomposition of sulfite and sulfate species on Pt/Pd catalysts: An SO_2 oxidation and sulfur exposure study. *ACS Catalysis*, *9*(1), 640–648. https://doi.org/https://doi.org/10.1021/acscatal.8b03529
27. Du, X., Xue, J., Wang, X., Chen, Y., Ran, J., & Zhang, L. (2018). Oxidation of sulfur dioxide over V_2O_5/TiO_2 catalyst with low vanadium loading: A theoretical study. *The Journal of Physical Chemistry C*, *122*(8), 4517–4523. https://doi.org/https://doi.org/10.1021/acs.jpcc.8b00296
28. Shokri, A., Rabiee, F., & Mahanpoor, K. (2017). Employing a novel nanocatalyst (Mn/Iranian hematite) for oxidation of SO_2 pollutant in aqueous environment. *International Journal of Environmental Science and Technology*, *14*, 2485–2494. https://doi.org/10.1007/s13762-017-1346-7
29. Yuan, P., Ma, H., Shen, B., & Ji, Z. (2022). Abatement of NO/SO_2/Hg0 from flue gas by advanced oxidation processes (AOPs): Tech-category, status quo and prospects. *Science of The Total Environment*, *806*, 150958. https://doi.org/10.1016/j.scitotenv.2021.150958
30. Boucheikhchoukh, A., Thibault, J., & Fauteux-Lefebvre, C. (2020). Catalyst design using artificial intelligence: SO_2 to SO_3 case study. *The Canadian Journal of Chemical Engineering*, *98*(9), 2016–2031. https://doi.org/https://doi.org/10.1002/cjce.23756
31. Wang, X., Kang, Y., Li, J., & Li, D. (2019). Influence of cerium and cesium promoters on vanadium catalyst for sulfur dioxide oxidation. *Korean Journal of Chemical Engineering*, *36*, 650–659. https://doi.org/10.1007/s11814-019-0257-3
32. Mazidi, M., Behbahani, R. M., & Fazeli, A. (2017). Ce promoted V2O5 catalyst in oxidation of SO2 reaction. *Applied Catalysis B: Environmental*, *209*, 190–202. https://doi.org/10.1016/j.apcatb.2017.02.034

33. Doering, F. J., & Berkel, D. A. (1987). Comparison of kinetic data for KV and CsV sulfuric acid catalysts. *Journal of Catalysis*, *103*(1), 126–139. https://doi.org/10.1016/0021-9517(87)90099-6
34. Nikiforova, A., Kozhura, O., & Pasenko, O. (2016). Leaching of vanadium by sulfur dioxide from spent catalysts for sulfuric acid production. *Hydrometallurgy*, *164*, 31–37. https://doi.org/https://doi.org/10.1016/j.hydromet.2016.05.004
35. Simonova, L. G., Bulgakova, Y. O., Lapina, O. B., & Bal'zhinimaev, B. S. (1997). Catalytic properties of alkali-promoted vanadium catalysts for oxidation of sulfur dioxide. *Kinetics and Catalysis*, *38*(6), 820–824.
36. Wang, T., Liu, J., Yang, Y., Sui, Z., Zhang, Y., Wang, J., & Pan, W.-P. (2020). Catalytic conversion of mercury over Ce doped Mn/SAPO-34 catalyst: Sulphur tolerance and SO_2/SO_3 conversion. *Journal of Hazardous Materials*, *381*, 120986. https://doi.org/10.1016/j.jhazmat.2019.120986
37. Mazidi, M., Mosayebi Behbahani, R., & Fazeli, A. (2017). Screening of treated diatomaceous earth to apply as V_2O_5 catalyst support. *Materials Research Innovations*, *21*(5), 269–278. https://doi.org/10.1080/14328917.2016.1211478
38. Goldsmith, B. R., Esterhuizen, J., Liu, J.-X., Bartel, C. J., & Sutton, C. (2018). *Machine learning for heterogeneous catalyst design and discovery*. *64*(7), 2311–2323 . https://doi.org/10.1002/aic.16198
39. Toyao, T., Maeno, Z., Takakusagi, S., Kamachi, T., Takigawa, I., & Shimizu, K. (2020). Machine Learning for Catalysis Informatics: Recent Applications and Prospects. *ACS Catalysis*, *10*(3), 2260–2297. https://doi.org/10.1021/acscatal.9b04186
40. Xin, H. (2022). Catalyst design with machine learning. *Nature Energy*, *7*(9), 790–791. https://doi.org/10.1038/s41560-022-01112-8
41. Kurogi, T., Etou, M., Hamada, R., & Sakai, S. (2021). Catalyst Design by Machine Learning and Multiobjective Optimization. *Journal of the Japan Petroleum Institute*, *64*(5), 256–260. https://doi.org/10.1627/jpi.64.256
42. Guan, Y., Chaffart, D., Liu, G., Tan, Z., Zhang, D., Wang, Y., Li, J., & Ricardez-Sandoval, L. (2022). Machine learning in solid heterogeneous catalysis: Recent developments, challenges and perspectives. *Chemical Engineering Science*, *248*, 117224. https://doi.org/10.1016/j.ces.2021.117224

43. Schlexer Lamoureux, P., Winther, K. T., Garrido Torres, J. A., Streibel, V., Zhao, M., Bajdich, M., Abild-Pedersen, F., & Bligaard, T. (2019). Machine learning for computational heterogeneous catalysis. *ChemCatChem*, *11*(16), 3581–3601. https://doi.org/https://doi.org/10.1002/cctc.201900595
44. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*, 115–133. https://doi.org/10.1007/BF02478259
45. Ojha, V. K., Abraham, A., & Snášel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, *60*, 97–116.
46. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8
47. Huang, G.-B., Chen, L., & Siew, C. K. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Networks*, *17*(4), 879–892. https://doi.org/10.1109/TNN.2006.875977
48. Gill, P. E., & Murray, W. (1972). Quasi-Newton methods for unconstrained optimization. *IMA Journal of Applied Mathematics*, *9*(1), 91–108. https://doi.org/10.1093/imamat/9.1.91
49. Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, *11*(2), 431–441. https://doi.org/10.1137/0111030
50. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv Preprint arXiv:1412.6980*. https://doi.org/10.48550/arXiv.1412.6980
51. García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: Methods and prospects. *Big Data Analytics*, *1*(1), 1–22. https://doi.org/10.1186/s41044-016-0014-0
52. Obaid, H. S., Dheyab, S. A., & Sabry, S. S. (2019). The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning. *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (Iemecon)*, 279–283.

53. Chu, X., Ilyas, I. F., Krishnan, S., & Wang, J. (2016). Data cleaning: Overview and emerging challenges. *Proceedings of the 2016 International Conference on Management of Data*, 2201–2206.
54. Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524. https://doi.org/10.1016/j.asoc.2019.105524
55. Malik, J. S., Goyal, P., & Sharma, A. K. (2010). A comprehensive approach towards data preprocessing techniques & association rules. *Proceedings of the 4th National Conference*, *132*.
56. Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(4), 433–459. https://doi.org/10.1002/wics.101
57. Wang, H., & Wang, S. (2010). Mining incomplete survey data through classification. *Knowledge and Information Systems*, *24*, 221–233. https://doi.org/10.1007/s10115-009-0245-8
58. Kramer, M. A. (1992). Autoassociative neural networks. *Computers & Chemical Engineering*, *16*(4), 313–328. https://doi.org/10.1016/0098-1354(92)80051-A
59. Agin, F., Thibault, J., & Fauteux-Lefebvre, C. (2024). Autoassociative neural network for missing data imputation: A case study via the styrene production process. *The Canadian Journal of Chemical Engineering*, *103*, 339–358. https://doi.org/10.1002/cjce.25377
60. Yu, T., & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. *arXiv Preprint arXiv:2003.05689*. https://doi.org/10.48550/arXiv.2003.05689
61. Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. *Automated Machine Learning: Methods, Systems, Challenges*, 3–33. https://doi.org/10.1007/978-3-030-05318-5_6
62. Gooijer, J. G. D., & Hyndman, R. J. (2006). 25 years of time series forecasting. *International Journal of Forecasting*, *22*(3), 443–473. https://doi.org/10.1016/j.ijforecast.2006.01.001
63. Qu, Q., Ma, Z., Clausen, A., & Jørgensen, B. N. (2021). A comprehensive review of machine learning in multi-objective optimization. *2021 IEEE 4th International Conference on Big Data and Artificial Intelligence (BDAI)*, 7–14.
64. Safari, M., Sohani, A., & Sayyaadi, H. (2021). A higher performance optimum design for a tri-generation system by taking the advantage of water-energy nexus. *Journal of Cleaner Production*, *284*, 124704. https://doi.org/10.1016/j.jclepro.2020.124704

65. Rangaiah, G. P., & Bonilla-Petriciolet, A. (2013). *Multi-objective optimization in chemical engineering: Developments and applications*. John Wiley & Sons.
66. Rangaiah, G. P., Feng, Z., & Hoadley, A. F. (2020). Multi-objective optimization applications in chemical process engineering: Tutorial and review. *Processes*, *8*(5), 508. https://doi.org/10.3390/pr8050508
67. Pereira, J. L. J., Oliver, G. A., Francisco, M. B., Cunha, S. S., & Gomes, G. F. (2021). A review of multi-objective optimization: Methods and algorithms in mechanical engineering problems. *Archives of Computational Methods in Engineering*, *29*(4), 2285–2308. https://doi.org/10.1007/s11831-021-09663-x
68. Chiandussi, G., Codegone, M., Ferrero, S., & Varesio, F. E. (2012). Comparison of multi-objective optimization methodologies for engineering applications. *Computers & Mathematics with Applications*, *63*(5), 912–942. https://doi.org/10.1016/j.camwa.2011.11.057
69. Baril, C., Yacout, S., & Clément, B. (2011). Design for Six Sigma through collaborative multiobjective optimization. *Computers & Industrial Engineering*, *60*(1), 43–55. https://doi.org/10.1016/j.cie.2010.09.015
70. Ojha, M., Singh, K. P., Chakraborty, P., & Verma, S. (2019). A review of multi-objective optimisation and decision making using evolutionary algorithms. *International Journal of Bio-Inspired Computation*, *14*(2), 69–84. https://doi.org/10.1504/IJBIC.2019.101640
71. Deb, K. (2011). *Multi-objective optimisation using evolutionary algorithms: An introduction*. Springer.
72. Miettinen, K. (1999). *Nonlinear multiobjective optimization* (Vol. 12). Springer.
73. Deb, K. (2008). Introduction to evolutionary multiobjective optimization. *Multiobjective Optimization: Interactive and Evolutionary Approaches*, 59–96. https://doi.org/10.1007/978-3-540-88908-3_3
74. Gomes, G. F., de Almeida, F. A., da Silva Lopes Alexandrino, P., da Cunha, S. S., de Sousa, B. S., & Ancelotti, A. C. (2019). A multiobjective sensor placement optimization for SHM systems considering Fisher information matrix and mode shape interpolation. *Engineering with Computers*, *35*, 519–535. https://doi.org/10.1007/s00366-018-0613-7
75. Zeleny, M. (1973). Multiple criteria decision making. *Compromise Programming*, 262–301.

76. Gunantara, N. (2018). A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, *5*(1), 1502242. https://doi.org/10.1080/23311916.2018.1502242
77. Odu, G. O., & Charles-Owaba, O. E. (2013). Review of multi-criteria optimization methods–theory and applications. *IOSR Journal of Engineering*, *3*(10), 01–14. https://doi.org/10.9790/3021-031020114
78. Bui, L. T., & Alam, S. (2008). *Multi-objective optimization in computational intelligence: Theory and practice*. IGI Global.
79. Cáceres Sepúlveda, G. (2019). *Relevance of multi-objective optimization in the chemical engineering field* [Master thesis, University of Ottawa]. https://doi.org/10.20381/ruor-24026
80. *Metals on Bloomberg*. (2023). Bloomberg Terminal
81. *World Bank Commodity Markets*. (2023). https://www.worldbank.org/en/research/commodity-markets
82. Hamzehlouyan, T., Sampara, C., Li, J., Kumar, A., & Epling, W. (2016). Sulfur poisoning of a Pt/Al_2O_3 oxidation catalyst: Understanding of SO_2, SO_3 and H_2SO_4 impacts. *Topics in Catalysis*, *59*, 1028–1032. https://doi.org/10.1007/s11244-016-0592-0
83. Benzinger, W., Goerke, O., & Pfeifer, P. (2016). Catalytic coating in microstructured devices and their performance in terms of the SO_2 oxidation. *Journal of Sol-Gel Science and Technology*, *80*, 802–813. https://doi.org/10.1007/s10971-016-4174-1

Chapter 6

Machine Learning–Driven Design of Catalytic Processes for Sulfur Dioxide Oxidation: Lessons from the Trenches

Farough Agin, Jules Thibault and Clémence Fauteux-Lefebvre*

Department of Chemical and Biological Engineering

University of Ottawa, Ottawa, Ontario, K1N 6N5, Canada

*Corresponding Author: Clémence Fauteux-Lefebvre

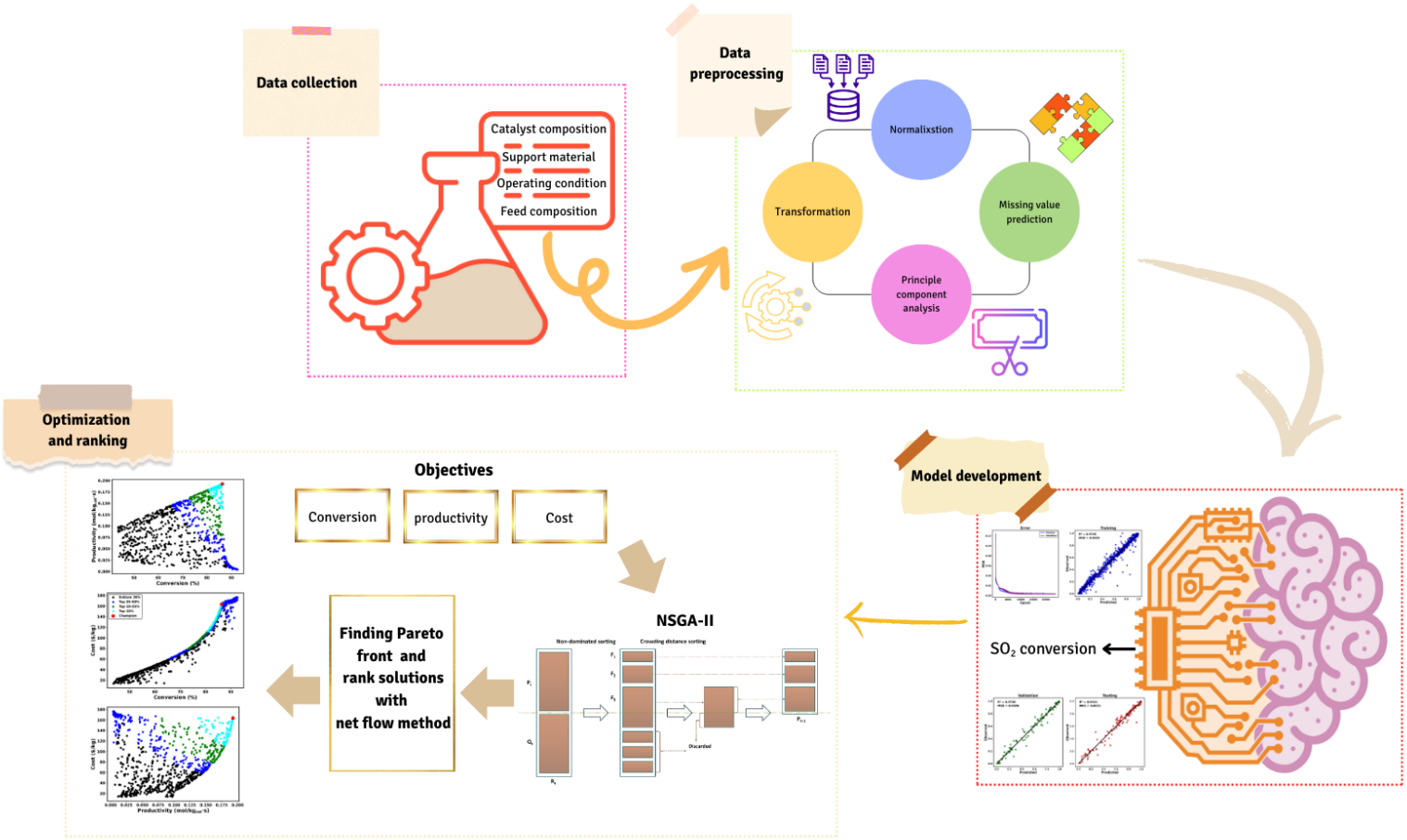
This chapter is a manuscript submitted to AIChE Journal.

6.1 Abstract

Despite the growing use of ML in chemical engineering, the catalytic conversion of sulfur dioxide (SO_2) to sulfur trioxide (SO_3) remains underexplored. This study presents a systematic application of ML-based modelling and optimization to this reaction, addressing a critical gap in catalyst design and process understanding. It includes a comprehensive step-by-step generalizable approach for catalytic processes, outlining the entire process from problem definition and data collection to identifying an optimal catalyst for the catalytic conversion of SO_2 to SO_3. Various preprocessing techniques to enhance the quality of raw data gathered from diverse literature sources are compared, leading to the training of a refined ML model. Multi-objective optimization (MOO) is applied to each scenario to circumscribe the Pareto domain while multi-criteria decision-making (MCDM) technique ranks the Pareto-optimal solutions to determine the best candidate for subsequent experimental validation. This study integrates ML in catalyst optimization with a comprehensive methodology for catalyst discovery.

Keywords: Catalyst design, Data collection, Data preprocessing, Machine learning, Optimization, Solution

6.2 Graphical abstract



6.3 Introduction

For over 130 years, engineers have used mathematical modelling to understand and design chemical processes [1]. Foundational models, such as the Navier-Stokes [2] equations for viscous fluids, are pivotal but often require numerical solutions due to their complexity, leading engineers to resort to simplified models. Recently, chemical engineering has benefited from extensive data availability due to industrial sensors, literature experimental data, and advancements in process modelling. Nevertheless, extracting useful insights from this data remains challenging [1]. The increase in computational power and the development of mathematical tools, especially machine learning (ML), have changed this landscape. ML, a subset of artificial intelligence (AI), applies mathematical and statistical methods to learn from data and find patterns without explicit programming. This paradigm shift has spurred significant interest in AI and ML within chemical engineering, particularly for complex tasks in the process industry (e.g., batch or semi-batch production, material discovery and development, and model parameter estimation) [3,4] and catalyst design [5,6]. Within this broader shift toward data-driven chemical engineering, certain environmentally and industrially critical reactions remain understudied from a modelling perspective.

The conversion of SO_2 to SO_3 is a particularly significant process in chemical engineering due to its applications in pollution control and sulfuric acid (H_2SO_4) production. To mitigate pollution, industries employ flue-gas desulfurization (FGD) methods, which can be categorized into wet and dry processes [7–10]. The oxidation of SO_2 to SO_3 is also explored as a novel approach for deeper SO_2 capture in dry methods, where the generated SO_3 can react with a solid sorbent to form stable sulfates. This reaction is often facilitated by catalysts that enhance the efficient conversion of SO_2 to SO_3 under controlled conditions [9]. According to available literature, sodium-vanadium, platinum-palladium, platinum, and platinum-tin are widely recognized as the catalysts predominantly utilized in FGD [11–16]. The oxidation of SO_2 to SO_3 is also a critical step in the production of H_2SO_4, the most widely produced chemical globally, used in manufacturing fertilizers, chemicals, metal processing, car batteries, and petroleum refining [17–19]. This reaction is typically performed in adiabatic catalytic packed bed reactors with intercooling heat exchangers to manage the operating temperature to optimize the reaction rate and final conversion [20–22]. For the H_2SO_4 production, the catalysts employed in research include potassium-vanadium, cesium-vanadium, platinum, as well as mixtures comprising vanadium combined with other compounds

and/or promoters such as calcium, copper, iron, barium, manganese, and magnesium [11–16]. Current research still aims to improve catalyst formulations, enhancing their effectiveness and durability for both environmental and industrial applications. Despite significant advances in applying ML to catalytic systems, the catalytic conversion of SO_2 to SO_3 has received virtually no attention in the literature. Existing studies have primarily focused on reactions such as CO oxidation, water-gas shift (WGS), oxidative coupling of methane (OCM), and methanol synthesis, leaving a critical gap in understanding and optimizing the SO_2 oxidation process through data-driven methods. This lack of ML studies stands in contrast to the substantial progress achieved for many other catalytic reactions.

Moreover, the application of ML in catalysis discovery remains in its early stages relative to other fields. This is mainly because comprehensive datasets are not sufficiently available and developing them requires considerable time and resources. Furthermore, predicting catalyst properties is challenging due to their dependence on numerous independent variables [23,24]. One recent advancement in catalysis involves meta-analysis, which applies various ML techniques to datasets obtained from different sources [5,6]. Extensive datasets in catalysis can be generated internally through high-throughput experimentation [25,26] or computational density functional theory [27–30]. Additionally, external sources such as publicly available scientific literature offer valuable, pre-tested data. The former methods are expensive yet yield precise results, whereas the latter is cost-effective but may include noise, missing data, and inconsistencies across different sources [31].

Catalyst design involves several interrelated factors, such as the type and amount of active metal, the type and quantity of promoters, the nature of the support, the method of preparation, and pretreatment conditions, all of which influence the catalyst's properties. Therefore, modifications in any of these aspects can substantially impact the catalyst's activity and selectivity. Furthermore, variations in operating conditions, including reaction temperature, feed flow rate, feed composition, and the amount of catalyst, can result in significantly different catalytic reaction outcomes [32]. A survey of prior work using literature-based datasets further illustrates the wide range of catalytic reactions examined through ML. Table 6.1 provides a brief overview of the literature on data extraction and catalyst discovery, categorized by reaction type as it relates to a selection of ML applications. The full description of these works is presented in Appendix B.

Table 6.1. Summary of ML applications in catalysis using literature data.

Reaction	Dataset Size	# Papers	ML Methods	Input Features	Output Variable	Reference
Ethylbenzene Dehydrogenation (SnO catalysts)	38	1	Artificial neural networks (ANN)	Surface area; valence; catalyst composition; ionic radius; coordination number; electronegativity; partial charge of O ions; heat of formation	Selectivity	[33]
CO Oxidation (Cu-based)	1337	20	ANN	Cu loading; second metal type & loading; support type; preparation method; calcination temperature/time; time on stream; reaction temperature; feed compositions (H_2, O_2, CO, H_2O, CO_2); flow rate/catalyst ratio	CO conversion	[32]
CO Oxidation (Noble metals)	5610	80	Decision tree (DT), ANN	Noble metal type; preparation method; support type; promoters; temperature; feed $H_2/O_2/CO/H_2O/CO_2$; time on stream; flow rate/catalyst mass	CO conversion	[34]
CO Oxidation Kinetics (Au catalysts)	882	19	ANN	Catalyst composition; supports; promoters; reaction temperature; feed composition; flow rate; time on stream	Reaction rate	[34]
WGS — Mixed catalysts	283	Not reported	ANN	Temperature; pressure; catalyst mass; GHSV; feed composition; active phase; support; promoters; surface area; calcination temp/time	CO conversion	[35]
WGS (Pt/Au catalysts)	4360	84	DT, ANN, support vector machine (SVM)	Metal wt%; preparation method; support; promoter wt%; calcination temperature/time; feed composition; reaction temp; time on stream; mass/flow rate	CO conversion	[36]
WGS (Pt/$Ce_xZr_{1-x}O_2$ catalysts)	251	Not reported	ANN	Catalyst composition; Ce/Zr ratio; loadings; preparation; impregnation; calcination; reaction temperature; feed ratios; space velocity	CO conversion	[37]
OCM	4759	Not reported	ANN, random forest (RF)	Elemental descriptors (radius, electronegativity, group, affinity, ionization); catalyst composition; temp; CH_4/O_2 ratio; flow rate	C_2 yield	[38]
OCM/WGS/CO oxidation (elemental method)	Not reported	Not reported	ANN, SVM, Gaussian process (GP)	Elemental descriptors of metals, promoters, supports; reaction conditions including temperature and CH_4/O_2 ratio	Conversion / yield / selectivity	[39]
Methane Steam Reforming	5508	81	DT	Metal wt%; support; promoter; preparation method; temp; pressure; feed composition; GHSV; catalyst mass; reactor type	CH_4 conversion	[40]
Methane Dry Reforming	5521	101	DT, ANN	Support properties; metal type; promoter; preparation; activation; temperature; CH_4/CO_2 feed; flow rate; residence time	CH_4 conversion	[41]
CO_2-to-Methanol Synthesis	206	27	ANN, RF, Gradient Boosting	Catalyst composition; calcination temperature/time; prep method; pore size; surface area; temperature; pressure; H_2/CO_2 ratio	CO_2 conversion; methanol selectivity	[42]
SCR (NO_x)	2126	41	ANN	Composition; structure; morphology; preparation; temperature; $NO/NO_2/NH_3$ feed; space velocity	NO_x conversion	[43]
SCR (Ce-based)	5654	126	XGBoost (primary), and multiple ML models	Composition; synthesis parameters; metal loading; calcination; reaction temperature; GHSV; feed (NO, NO_2, NH_3, O_2, H_2O, SO_2)	NO_x conversion	[44]

As shown in Table 6.1, ML approaches have been applied across a broad spectrum of catalytic reactions, supported by datasets that vary widely in size, complexity, and feature diversity. These studies collectively demonstrate that data-driven models can successfully capture structure–performance relationships when sufficient experimental information is available, reinforcing the value of literature-derived datasets for catalyst discovery and optimization. Comprehensive reviews further highlight the transformative role of AI and ML in catalyst discovery, optimization, and process understanding can be found in [31,45,46]. Despite these advances, these studies also reveal persistent challenges associated with heterogeneous literature datasets. Common obstacles include inconsistencies across datasets, the prevalence of missing and noisy data, limited sample sizes for specific catalyst classes, restricted operating conditions, and the inherent nature of catalyst structures [6,28,30].

Motivated by these challenges and the gap in SO_2 oxidation modelling, the present study develops the first systematic ML-based framework for this reaction. The SO_2 oxidation process represents a compelling and important case for exploring complex structure and performance relationships under diverse operating conditions. Building upon the foundation established in our previous contributions [47,48], the present study expands the dataset to include comprehensive experimental information available up to 2024 and introduces additional variables and advanced modelling strategies. To the best of our knowledge, this constitutes the first systematic investigation of SO_2 to SO_3 conversion using an integrated framework of literature data mining, ML, and multi-objective optimization (MOO). In this work, we aim to overcome the challenges associated with non-uniform literature data by proposing a detailed, step-by-step procedure for developing a generalized predictive model. The compiled dataset serves as the foundation for training ML models that are subsequently integrated into a MOO framework to delineate the Pareto domain for the SO_2 to SO_3 reaction. Finally, a multi-criteria decision-making (MCDM) technique is applied to rank Pareto optimal solutions according to expert preferences. Collectively, this methodology, encompassing data collection, model development, optimization, and decision analysis, provides a robust and scalable framework for advancing catalyst design and deepening process-level understanding in SO_2 oxidation, and with broader applicability to other catalytic systems.

6.4 Methodology

The comprehensive structure of this project is depicted in Figure 6.1. The proposed framework includes defining the problem and establishing the project's primary objective. Subsequently, relevant data is collected from various sources. The subsequent steps involve iterative cycles of data preprocessing, model development, optimization, and solution ranking, continuing until the most logical and optimal solutions are achieved. The results were generated using Python, leveraging libraries including TensorFlow, Scikit-learn, and Pymoo, within the computational framework provided by the Digital Research Alliance of Canada. In the subsequent sections, we will briefly elaborate on each of these steps and detail the specific procedures undertaken in this project.

6.4.1 Problem definition

As outlined in the introduction, the problem defined in this study is to model and evaluate the catalytic conversion of SO_2 to SO_3 across a wide range of catalyst formulations and operating conditions reported in the literature. In this context, the objective is to compile the available experimental information, represent the key catalytic and reaction variables in a structured form, and establish a predictive framework that enables systematic analysis of conversion behaviour under different conditions.

6.4.2 Literature data

To acquire raw data, direct experimentation within a specific system requires significant resources, including time, labour, infrastructure, and financial investment. Therefore, leveraging existing literature (meta-analysis), which provides extensive research findings accessible online, is a more practical approach.

In this work, a comprehensive screening process of 152 papers published between 1929 and 2024 was undertaken to identify suitable articles through a search conducted in the Scopus database. The search queries included keywords such as "catalyst", " SO_2", "sulfur dioxide", "sulfuric acid", "kinetics", and related terms. To be included in the dataset of this project, scientific articles must report catalytic activity involving heterogeneous catalysts. These articles should provide comprehensive details of the experimental conditions, clearly distinguishing the independent variables and the resulting dependent variables. The dataset captures information

related to catalyst composition and properties (e.g., specific surface area, pore characteristics, particle size), synthesis methodologies (e.g., calcination parameters such as temperature and time). Additionally, it includes operating conditions such as temperature, pressure, catalyst mass, inlet volumetric flow rate, and the mole fractions of O_2 and SO_2 within the system. Finally, the dataset includes the conversion of SO_2 as the output variable.

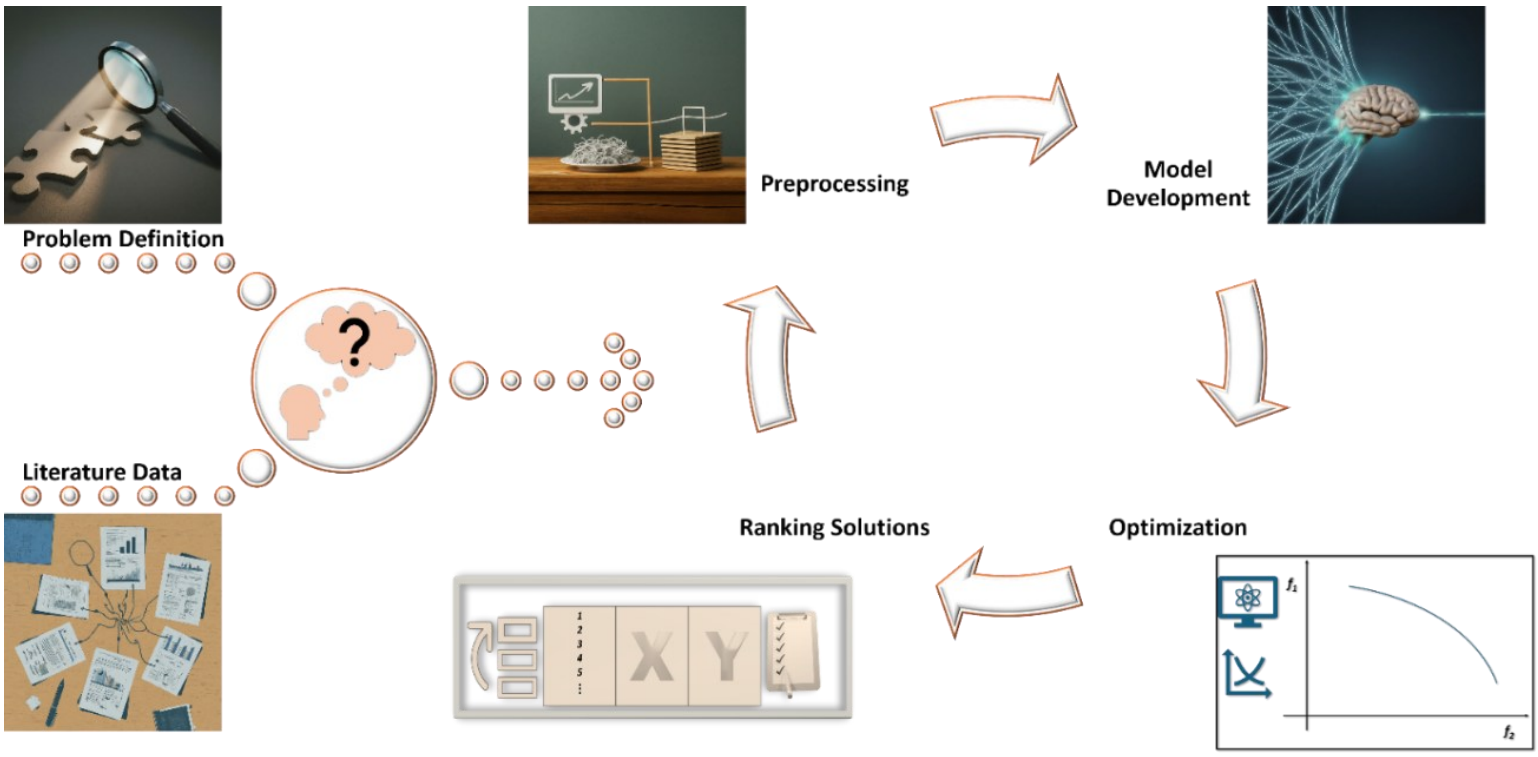


Figure 6.1. Proposed framework of optimized catalysts with knowledge extraction.

Consequently, 32 scientific articles were identified as suitable for data extraction. The selection criteria for inputs into the final database were further influenced by inconsistent reporting in some papers, such as incomplete data on critical variables like calcination parameters and ambiguity in the description of synthesis methods. Further inconsistencies included varying experimental conditions, inconsistent measurement techniques for conversion, and inadequate sample sizes or a lack of replication in some studies. Accordingly, catalysts and promoters with insufficient records and minimal variation of experimental conditions were omitted. The first screening provides a dataset of 21 catalysts and promoters, three support materials indicated by binary indicators, and operating parameters such as temperature, pressure, the ratio of catalyst weight to volumetric flow

rate (W/V), and the mole fractions of SO_2 and O_2. The output variable, representing the conversion of SO_2 to SO_3, completes the database structure. Therefore, the initial number of records before preprocessing is 1 922. Figure 6.2 summarizes the inputs and the output of the dataset.

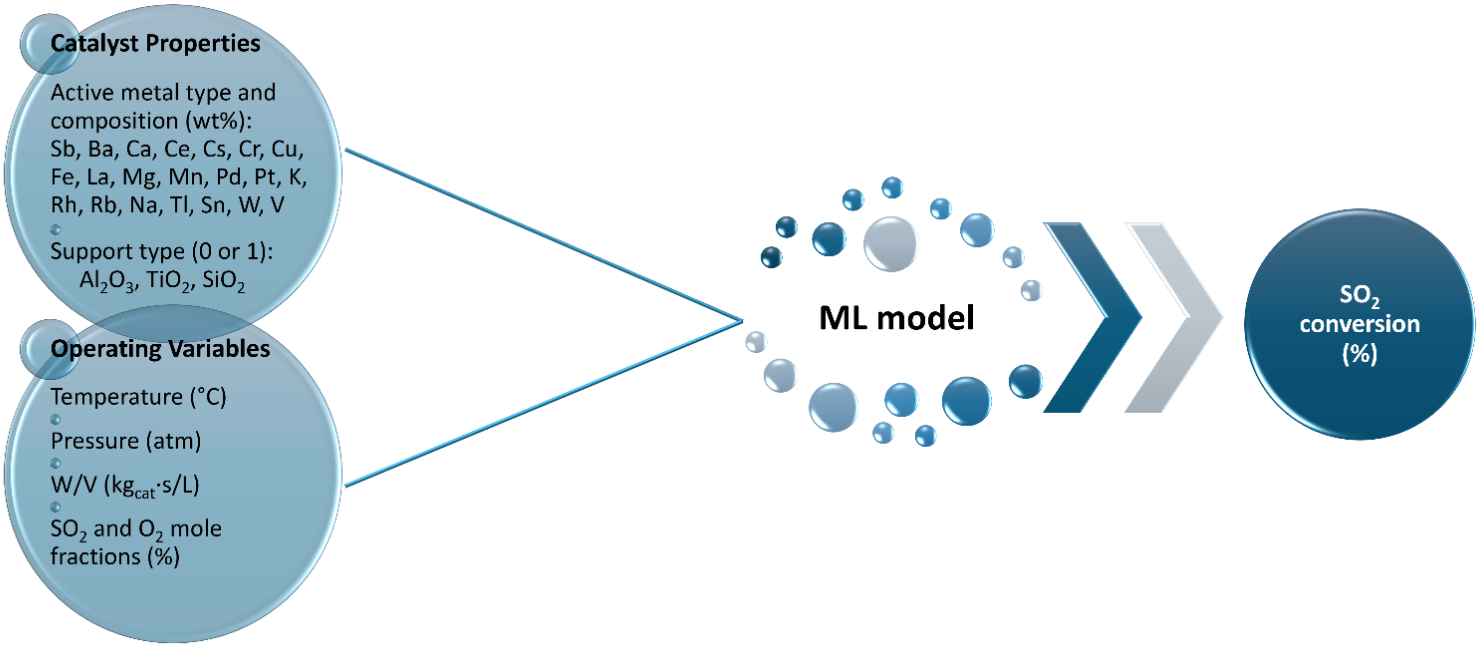


Figure 6.2. Raw dataset without any missing variables.

6.4.3 Data preprocessing

The accuracy, efficiency, and robustness of the developed model rely directly on the availability of high-quality data (i.e., data without missing, constant, redundant, or irrelevant values), encapsulated by the popular notion of "garbage in, garbage out" [49]. Data preprocessing involves five main steps: data integration, cleaning, normalization, handling missing data, and noise identification [50,51]. Additionally, feature engineering is conducted as part of data preprocessing, with dimensionality reduction being a critical component [52,53]. Dimensionality reduction reduces computational time and cost, prevents overfitting, and improves interpretability for high-dimensional data. It includes two main approaches: feature extraction and feature selection [49]. The complete description can be found in Appendix B.

6.4.3.1 Missing data handling

The initial dataset contained 1,922 records and 30 variables (29 inputs and 1 output), but several input variables exhibited substantial missing data, ranging from about 26% for pore size to nearly 98% for calcination time. To investigate the suitability of using an autoassociative neural network (AANN) for imputing missing data, the model was first tested on the available and complete dataset to evaluate its ability to predict missing variables [54]. The dataset was divided into training, validation, testing, and an additional 40-record evaluation subset to assess the model's performance. After hyperparameter tuning, the AANN achieved low mean square error (MSE) values and high R^2 scores, indicating strong reconstruction capability. However, when tested on unseen data with artificially introduced missing values, the model did not predict them accurately due to weak correlations among variables. Indeed, estimating a missing value for a given variable depends on its correlation with one or more other variables in the dataset. As the low inter-variable correlation hindered reliable imputation, variables with missing data were removed, resulting in a refined dataset containing 30 variables for subsequent modelling. The full description can be found in Appendix B.

6.4.3.2 Feature engineering

During data preprocessing, feature engineering was applied to reduce dimensionality and overfitting using random forest feature importance and principal component analysis (PCA). As shown in Figure 6.3a, the random forest regression model, optimized through random search and five-fold cross-validation, achieved strong performance (MSE = 0.059, $R^2 = 0.94$) and identified W/V ratio, temperature, vanadium, SO_2 and O_2 mole percentages, potassium, platinum, pressure, cesium, titania, and silica as the most influential variables. Figure 6.3b presents the PCA results, indicating that 23 components captured 95% of the total variance, supporting moderate potential dimensionality reduction. The dataset was then validated by reviewing all sources, consolidating duplicates, and averaging conversion values for identical inputs. These steps ensured a consistent, high-quality dataset for subsequent ANN model training and optimization.

6.4.3.3 Additional preprocessing

Certain variables were excluded from the dataset due to limited records and low variance, which reduced their contribution to model accuracy. These included antimony, chromium, barium, rhodium, rubidium, thallium, and tungsten. Although relevant to catalyst composition, these metals were typically reported under fixed or nearly identical experimental conditions, resulting in

insufficient variability for meaningful analysis. Their removal is consistent with the random forest and PCA findings, which indicated low feature importance and confirmed that excluding them did not significantly affect the total explained variance. After refining the dataset, additional preprocessing steps were applied, including outlier removal for W/V ratio, SO_2, and O_2 mole fractions using the Z-score method ($|Z| > 3$), followed by normalization of all variables to a [0,1] range. The final refined dataset contained 22 input variables and 1 270 records, ensuring higher quality and consistency for the subsequent development of ANN models. The full description is presented in Appendix B.

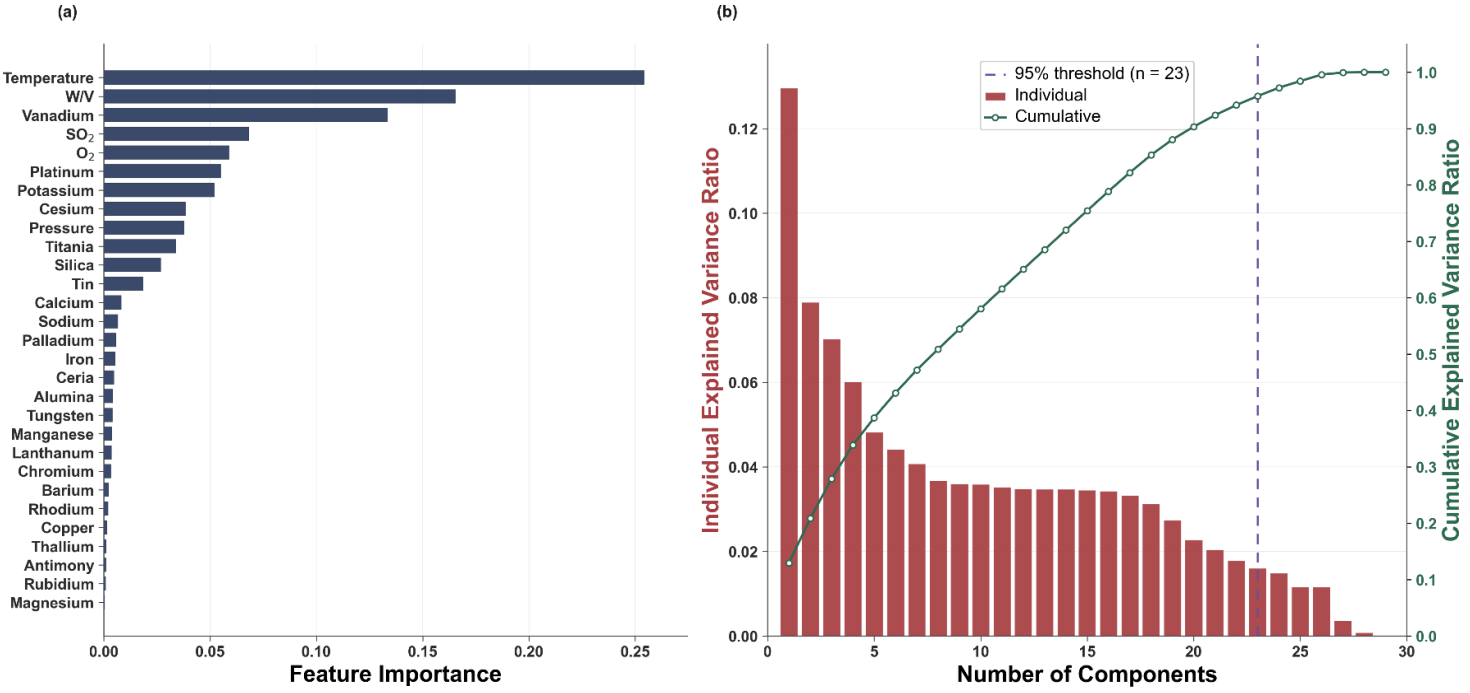


Figure 6.3. Feature importance visualization of the 29 variables (conversion excluded) determined by (a) random forest regressor and (b) PCA.

6.4.4 Model development

Among ML techniques, ANNs are particularly favoured due to their capacity to manage non-linearity, high-dimensionality, uncertainties, and multi-parametric challenges [55,56]. In chemical process engineering, a significant proportion of research papers and ML model applications have utilized ANNs, constituting over 35% of the total supervised ML models and more than 43% in hybrid modelling approaches up to 2021 [57]. Thus, our objective is to employ ANNs to develop the necessary data-driven model for this project.

The design of any ANN model often relies on prior experience, which may yield functional yet suboptimal structures. To enhance ANN model performance, hyperparameter tuning is performed, involving the adjustment of parameters fixed before training. These parameters include the number of hidden layers, the number of neurons per layer, the type of activation function, the optimizer, the learning rate, the batch size, and the number of epochs. Several methods are employed to identify the optimal hyperparameters, with grid search, random search, and Bayesian optimization being the most prominent [58,59]. Additionally, techniques such as early stopping, regularization, initialization, and batch normalization are utilized during the training phase to further improve network performance [59,60]. Hyperparameter tuning and network optimization during the training phase are performed using a validation dataset. To assess the overall performance of the model and its ability to generalize to unseen data, the entire dataset used during the model development phase was partitioned into training, validation, and test subsets [61]. In this project, hyperparameter tuning was carried out using a random search strategy combined with 5-fold cross-validation to develop accurate and reliable ANN models.

The next step involves selecting the appropriate performance metrics to evaluate the ANN model. In regression models, these metrics assess forecasting accuracy by comparing predicted values against actual data from the testing set, quantifying the discrepancies. Commonly used regression error metrics include root mean squared error (RMSE), mean absolute error (MAE), MSE, Pearson correlation coefficient, and the coefficient of determination (R^2). These metrics quantify prediction errors and facilitate comparison of model performance [62]. For this project, MSE and R^2 are used to better assess and compare the performances of the models.

6.4.5 Optimization

The next step, as shown in Figure 6.1, involves optimizing the trained model to determine the optimal catalyst/promoter composition and operating parameters. While conventional optimization might focus on a single objective, such as maximizing SO_2 conversion, real-world chemical processes typically require the simultaneous optimization of multiple, often competing, objectives. For this project, three objective functions are considered: conversion, productivity, and catalyst cost, to better represent real-world performance criteria, leading to a MOO problem.

The aim of MOO is to identify a set of solutions that lie on the optimal Pareto front and assemble a diverse set of solutions that accurately represent this front. A widely used method for

this purpose is the multi-objective evolutionary algorithm (MOEA), a stochastic method. Like other optimization methods, MOEAs seek to find optimal Pareto solutions by employing the concept of dominance [63]. In each iteration, the value of each objective of individual solutions is calculated to determine their dominance within the population. This population may be combined with parent groups to form the next generation. Additionally, the objective space allows MOEAs to incorporate traditional support techniques, such as niching [64,65]. One prominent MOEA method is the non-dominated sorting genetic algorithm (NSGA-II), which aims to solve MOO problems by identifying Pareto-optimal solutions through an elitist strategy, an explicit diversity-preserving mechanism, and a focus on non-dominated solutions [66].

The first objective, namely the conversion of SO_2, is evaluated using the ANN model. The conversion of SO_2 is defined as the ratio of the number of moles of SO_2 converted at the reactor outlet to the number of moles of SO_2 introduced at the reactor inlet, as expressed in Equation (6.1).

$$X = \frac{F_{SO_2(in)} - F_{SO_2(out)}}{F_{SO_2(in)}} \tag{6.1}$$

where $F_{SO_2(in)}$ and $F_{SO_2(out)}$ are the inlet and outlet molar flow rates of SO_2, respectively.

The second objective, productivity, is defined as the rate at which the moles of SO_2 are converted to SO_3 per unit time and per unit mass of the catalyst and is quantified according to Equation (6.2).

$$P = \frac{y_{SO_2} XF}{W} \ (mol\ SO_3/kg - cat \cdot s) \tag{6.2}$$

where y_{SO_2} is the input SO_2 mole fraction, X is the conversion, F is the total inlet molar flow rate, and W is the catalyst weight.

The productivity expression (Equation (6.2)) can be reformulated in terms of the ratio between the catalyst mass and the total volumetric flow rate. To achieve this representation, the molar flow rate is converted into a volumetric flow rate by applying the ideal gas law under standard conditions (1 atm, 273.15 K), leading to Equation (6.3).

$$P = 4.462 \times 10^{-2} \frac{y_{SO_2} X}{W/V} \ (mol\ SO_3/kg - cat \cdot s) \tag{6.3}$$

Here W/V denotes the ratio of the catalyst bed mass to the volumetric flow rate, expressed in kg_{cat}·s/L. Notably, Equations (2) and (3) are equivalent and yield identical results.

The third objective function pertains to the cost of the catalyst used in the process. This objective function is expressed as a linear relationship between the mass fractions of constituent active metals and their respective costs. The total catalyst cost is obtained by summing the product of the price and mass fraction of each active metal, as expressed by Equation (6.4). While this objective function provides a simplified representation of catalyst cost considerations, it serves as a foundational example within the broader optimization framework, facilitating computation and interpretation.

$$C = \sum_{i}^{r} c_i m_i \tag{6.4}$$

where r is the number of active metals in the databank, c_i is the price per kg of each active metal, and m_i is the mass fraction of each active metal composing a given catalyst. The list of prices of the active metal has been obtained by searching in Bloomberg, and it is reported in Table 6.2 [67,68].

The dataset encompasses information on two distinct applications of SO_2 catalytic conversion: FGD and H_2SO_4 production. Both processes involve converting SO_2 into SO_3, but they utilize different metal catalysts supported by various materials. In addition, a crucial differentiating factor within the dataset is the SO_2 mole fraction in the feed. Records with an SO_2 concentration below 1% correspond to FGD, whereas those with a concentration above 7% pertain to H_2SO_4 production. This research primarily aims to optimize H_2SO_4 production due to its substantial economic significance as one of the most extensively produced chemicals worldwide. However, since the dataset includes data for both applications, it also allows for the exploration of optimal conditions for desulfurization. For this study, to specifically focus on H_2SO_4 production, the SO_2 mole fraction was fixed at 10%. Furthermore, to simulate industrial conditions, the oxygen concentration was maintained at 11%.

One of the case studies in this work focuses on identifying the optimal compositions and operating conditions for a vanadium and potassium catalyst system. Vanadium serves as the primary active metal catalyst, while potassium acts as a promoter. The selection was guided by the abundance of records for these catalysts and their higher contributions, as shown in the feature importance analysis in Figure 6.3. The optimization process involved varying the compositions of these catalysts, along with different temperatures and W/V ratios. Silica was chosen as the support material due to its reported efficacy with vanadium and potassium. The pressure was fixed at 1

atm, and the compositions of other components were set to zero. Conversion was determined using the ANN model, while the other two objectives were directly calculated from the predicted conversion and the decision variables (V and K wt%, temperature and W/V ratio). The NSGA-II algorithm, with a population of 1 000 and 10 000 generations, was employed to circumscribe the Pareto domain for the three objectives.

Table 6.2. The estimated price of each metal.

S.N.	Catalyst	Symbol	Price ($/kg)
1	Calcium	Ca	5
2	Cerium	Ce	4
3	Cesium	Ce	13 000
4	Copper	Cu	8
5	Iron	Fe	0.5
6	Lanthanum	La	4
7	Magnesium	Mg	3.5
8	Manganese	Mn	2
9	Palladium	Pd	43 500
10	Platinum	Pt	32 000
11	Potassium	K	850
12	Sodium	Na	300
13	Tin	Sn	30
14	Vanadium	V	250

6.4.6 Ranking Pareto-optimal solutions

After establishing the Pareto domain using domination principles, the final step involves ranking all non-dominated or Pareto-optimal solutions based on the decision-maker's preferences. In this study, the Net Flow Method (NFM) is employed for ranking solutions by incorporating four key factors for each objective: the relative weight (W_m), the indifference threshold (Q_m), the preference threshold (P_m), and the veto threshold (V_m). The first factor, W_m, represents the importance assigned to each objective or criterion, with the sum of all weights equalling one. The Q_m defines the range within which differences in the values of a specific objective between two solutions are deemed inconclusive, meaning neither solution is preferred for that objective. The P_m establishes the minimum difference required for one solution to be favoured over another based on the values of a specific objective. Finally, the V_m excludes solutions from consideration if their differences in a specific objective exceed a predefined limit, ensuring that even if a solution performs well in other criteria, it can be disregarded if it fails to meet acceptable standards in a critical objective.

By integrating these factors, NFM ensures that selected solutions align with decision-makers' priorities and effectively balance competing objectives. [69]. In this project, solutions are ranked using the parameters of the NFM method, specified in Table 6.3.

Table 6.3. NFM parameters for the vanadium and potassium case study.

Parameters	Conversion	Productivity	Cost
W	0.45	0.45	0.1
Q	2	0.01	50
P	5	0.03	100
V	10	0.1	200

In the following section, we adhere to the procedure outlined in Figure 6.1, testing various preprocessing methods to enhance the quality of the data and conducting hyperparameter tuning to ensure the optimal performance of the ANN model. For each case study, we have defined the Pareto domain and ranked the solutions using the NFM method. As depicted in Figure 6.1, this process is iterative and requires significant time and effort to design a reliable and robust model.

6.5 Results and discussion

After completing the preprocessing steps, several ANN models were developed and trained. First, to address the publication bias in the literature, where mainly positive outcomes are reported, an additional approach was attempted by introducing 45 logical records representing zero conversion under certain conditions, such as when potassium was used alone as a promoter. The dataset was reprocessed, normalized, and ANN models optimized through hyperparameter tuning, resulting in a model that initially showed acceptable performance. However, further analysis revealed that the inclusion of logical data introduced another bias, causing the model to underestimate the catalytic role of potassium even in the presence of vanadium. As this approach produced unsatisfactory results, the logical records were excluded, and other data transformation techniques were later explored to improve model accuracy and reliability. More details of this section can be found in Appendix B.

Upon analyzing the distribution of variables such as pressure, W/V, and SO_2 mole fraction, it was observed that these variables exhibited a left-skewed distribution with a wide range. For instance, approximately 60% of W/V values fell within the range of [0.003, 1], yet the overall range extended from [0.003, 195] before preprocessing and outlier removal, and narrowed to [0.003, 28.37] after these steps. Similar patterns were observed for pressure and SO_2 mole fraction.

To address this issue and normalize the distribution of these variables, three transformation methods were applied: uniform quantile [70], normal quantile [71], and Box-Cox transformations [72]. Uniform quantile transformation is a preprocessing technique that standardizes features to a uniform distribution between 0 and 1. This method is advantageous for ML, as it harmonizes feature scales and mitigates the influence of outliers, thereby improving model performance. Normal quantile transformation, on the other hand, adjusts features to approximate a normal distribution, aligning with the assumptions of many statistical tests and ML algorithms. Lastly, the Box-Cox transformation offers a flexible approach to normalizing data by applying a power transformation controlled by a lambda (λ) parameter, optimizing the distribution to meet statistical assumptions and stabilize variance. Each of these transformations was employed to rectify the skewness and improve the overall quality of the dataset, thereby enhancing the performance and reliability of subsequent analyses and models. Five variables were identified for applying the transformation techniques to modify their distributions. These variables include temperature, pressure, W/V, and the mole fractions of SO_2 and O_2. This procedure was consistently applied across all the explained transformation methods.

6.5.1 Scenario 1: Implementation of the uniform quantile transformer

In the context of utilizing the uniform quantile transformer, outliers were systematically removed from the dataset, followed by the application of Min/Max normalization for data normalization. Hyperparameter optimization for the final dataset was conducted. The optimal ANN model was trained with a learning rate of 0.001 with a batch size of 32 using the Adam optimizer, achieving the lowest error at epoch 14 609 based on the validation dataset. Figure 6.4 illustrates the parity plots for the training, validation, and testing datasets, demonstrating that the ANN model performs well and generalizes effectively on the testing data.

Figure 6.5 depicts the Pareto front generated by the NSGA-II algorithm, based on the ANN model trained using the uniform quantile method across the five variables previously discussed. As illustrated in Figure 6.5a-c, a general inverse relationship is observed—conversion tends to decrease as productivity increases, particularly noticeable on the right edge of the plot of Figure 6.5a. Additionally, attaining higher levels of both conversion and productivity typically incurs a higher associated cost. Figure 6.5d-e depict the distributions of the decision variables, revealing noticeable gaps in certain regions. The majority of non-dominated points are clustered near the

lower bounds of potassium and vanadium weight percentages, particularly at higher W/V ratios. Moreover, some non-dominated points appear at the minimum potassium level, while others correspond to the maximum vanadium concentration.

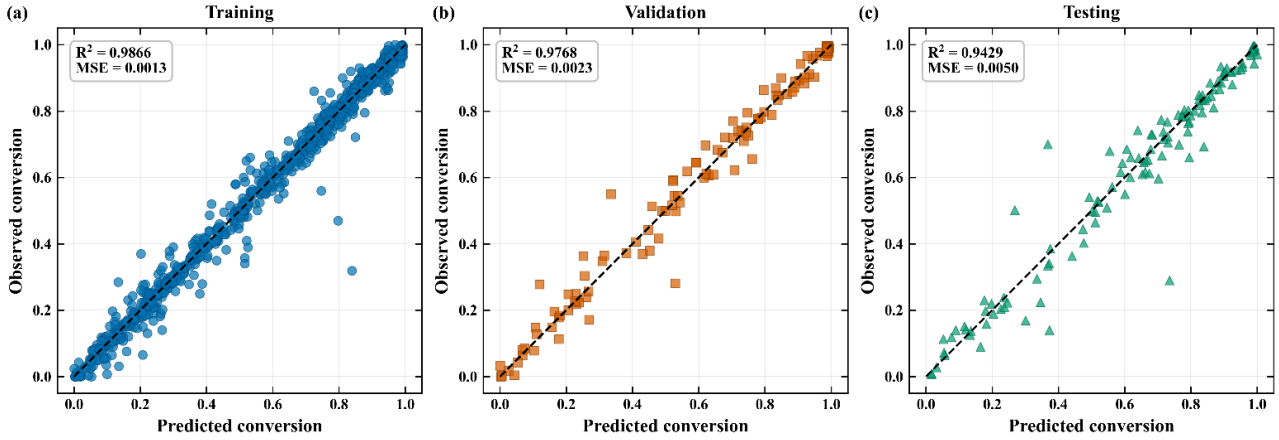


Figure 6.4. Parity plots of the ANN model for scenario 1 (uniform quantile transformer).

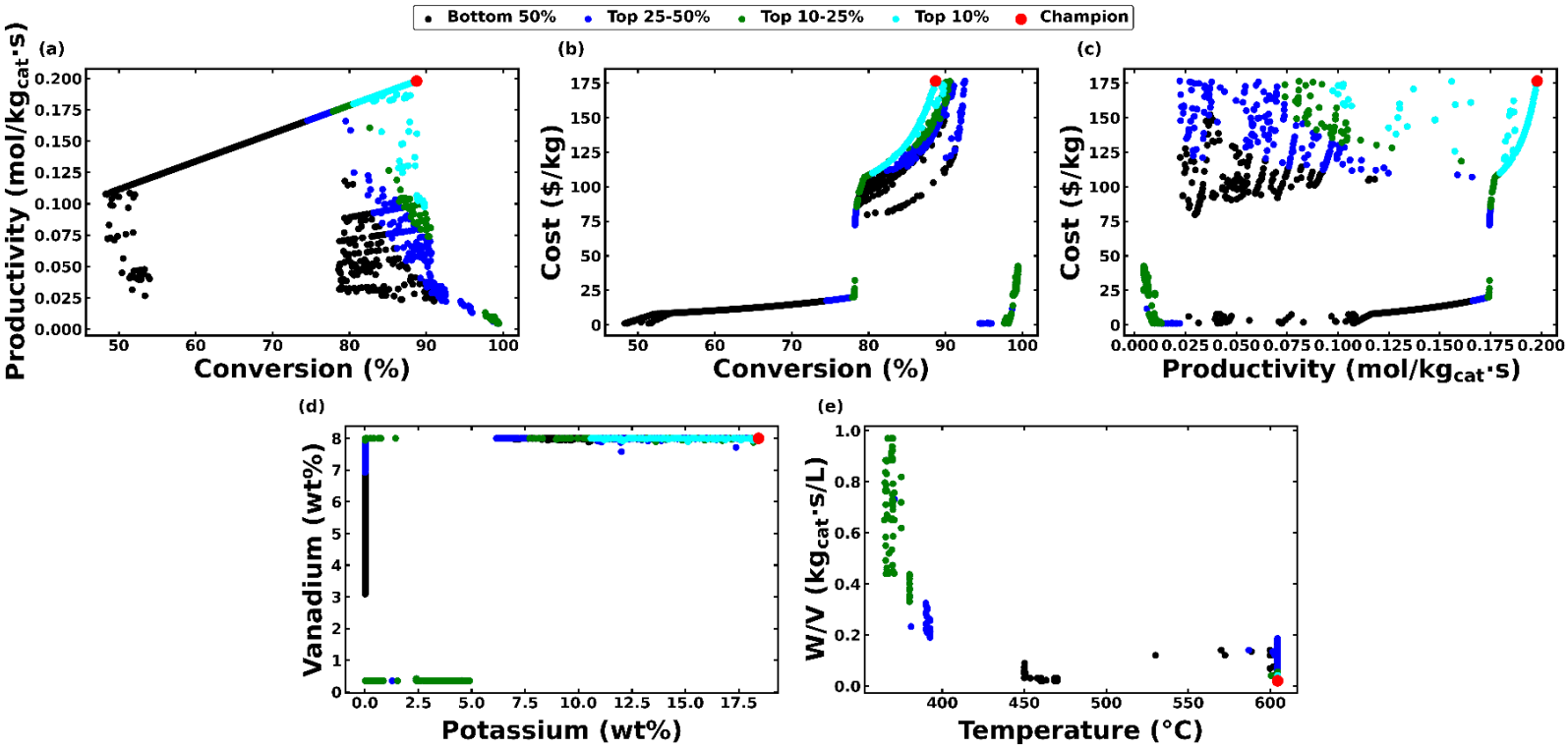


Figure 6.5. Results of MOO for vanadium-potassium case study and scenario 1 (uniform quantile transformer) (a) conversion vs productivity, (b) conversion vs cost, (c) productivity vs cost, (d) vanadium wt% vs potassium wt%, and (e) W/V vs temperature.

6.5.2 Scenario 2: Implementation of the normal quantile transformer

In this scenario, a normal quantile transformation was applied to five key variables, followed by the development of a robust ANN model for optimization. After outlier removal and data normalization, hyperparameter tuning was performed on the final dataset. The optimal ANN model was obtained and trained using the Adam optimizer with a 0.0005 learning rate with a batch size of 32, achieving minimum error at epoch 28 102. Figure 6.6 shows the parity plots confirming the strong model performance and good generalization on testing data.

The Pareto domain was generated using NSGA-II, with results shown in Figure 6.7. Figure 6.7a-c depict the Pareto front for the three objectives, highlighting that as conversion increases, productivity tends to decrease, with costs typically rising at higher productivity. In Figure 6.7d-e discontinuities appear, with non-dominated points in two regions: one at the lowest vanadium percentage with low temperatures and high W/V, and the other at maximum concentration of vanadium and potassium at high temperatures and low W/V. These discontinuities reflect the ANN model's limited extrapolation capability in data-sparse regions.

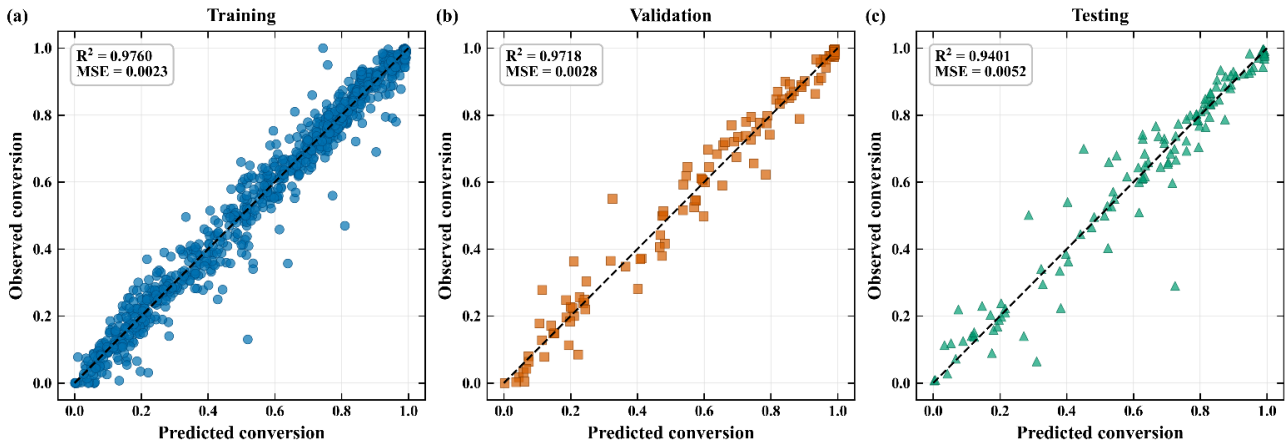


Figure 6.6. Parity plots of the ANN model for scenario 2 (normal quantile transformer).

6.5.3 Scenario 3: Implementation of the Box-Cox transformer

In the next scenario, the Box-Cox transformation was applied to the five previously mentioned variables, followed by the construction of a robust ANN model for the optimization phase. Outliers were removed, and normalization was performed before meticulously tuning the hyperparameters of the final dataset, which included variables transformed by the Box-Cox. The resulting optimal ANN model was trained using the Adam optimizer with a learning rate of 0.0005 with a batch size of 32, ultimately reaching its minimum error at epoch 29 733, utilizing the early stopping method.

Figure 6.8 displays the parity plots for the training, validation, and testing datasets, clearly demonstrating that the ANN model performs well and generalizes effectively to the testing data.

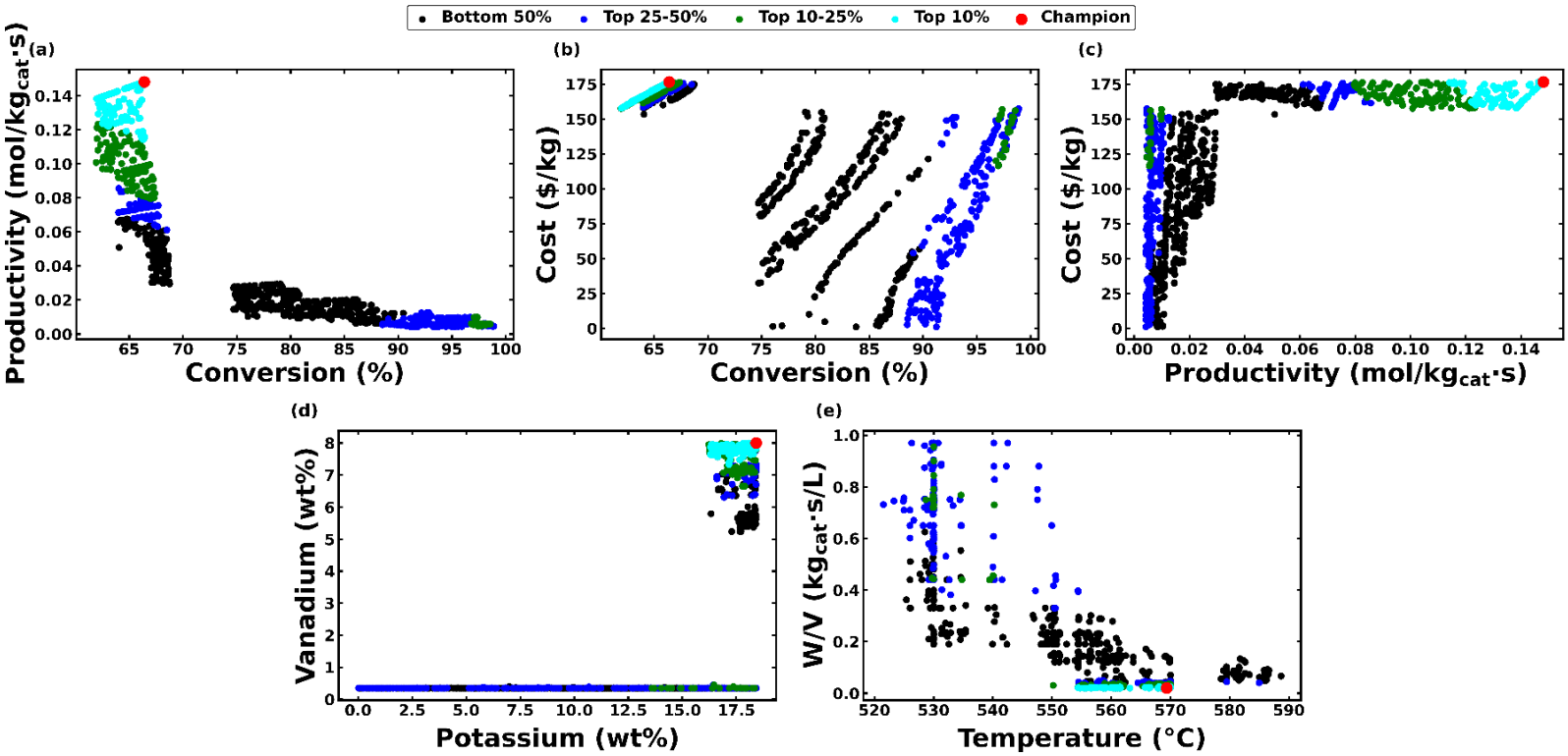


Figure 6.7. Results of MOO for vanadium-potassium case study for scenario 2 (normal quantile transformer) (a) conversion vs productivity, (b) conversion vs cost, (c) productivity vs cost, (d) vanadium wt% vs potassium wt%, and (e) W/V vs temperature.

Figure 6.9 depicts the Pareto domain derived from the NSGA-II optimization, encompassing both the objectives and the decision space. The Pareto fronts for the objectives are shown in Figure 6.9a-c. Notably, as conversion increases, the productivity decreases. Unlike the scenarios using uniform and normal quantile transformations, both conversion and productivity exhibit a narrower range of values. Additionally, costs rise with productivity, though the relationship between conversion and cost varies depending on whether productivity increases. Figure 6.9d-e illustrate the effects of varying potassium and vanadium levels on the objectives. Specifically, at low potassium levels, an increase in vanadium impacts the objectives, while at higher vanadium levels, a simultaneous increase in potassium is observed, which correlates with higher productivity, lower conversion, higher temperatures, and lower W/V ratios.

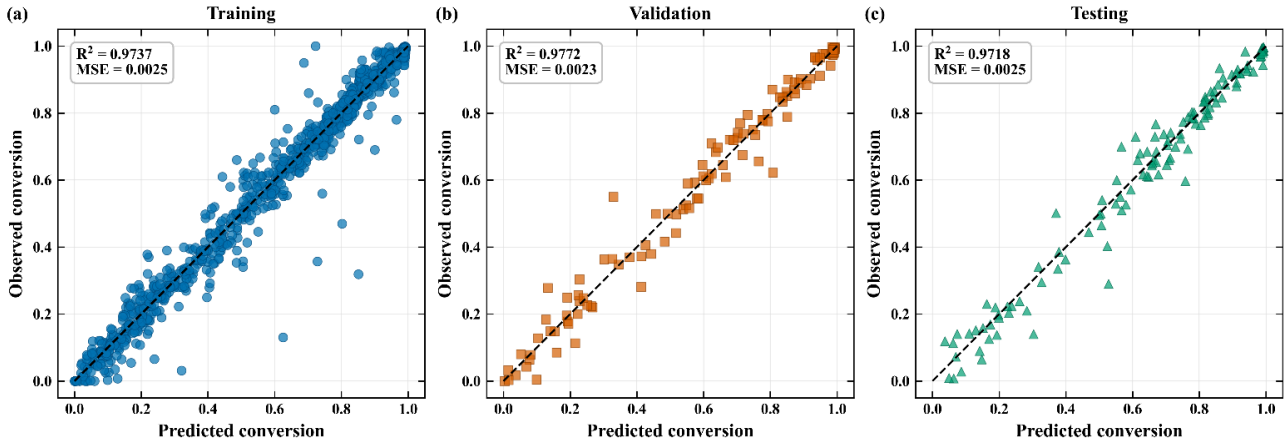


Figure 6.8. Parity plots of the ANN model for scenario 3 (Box-Cox transformer).

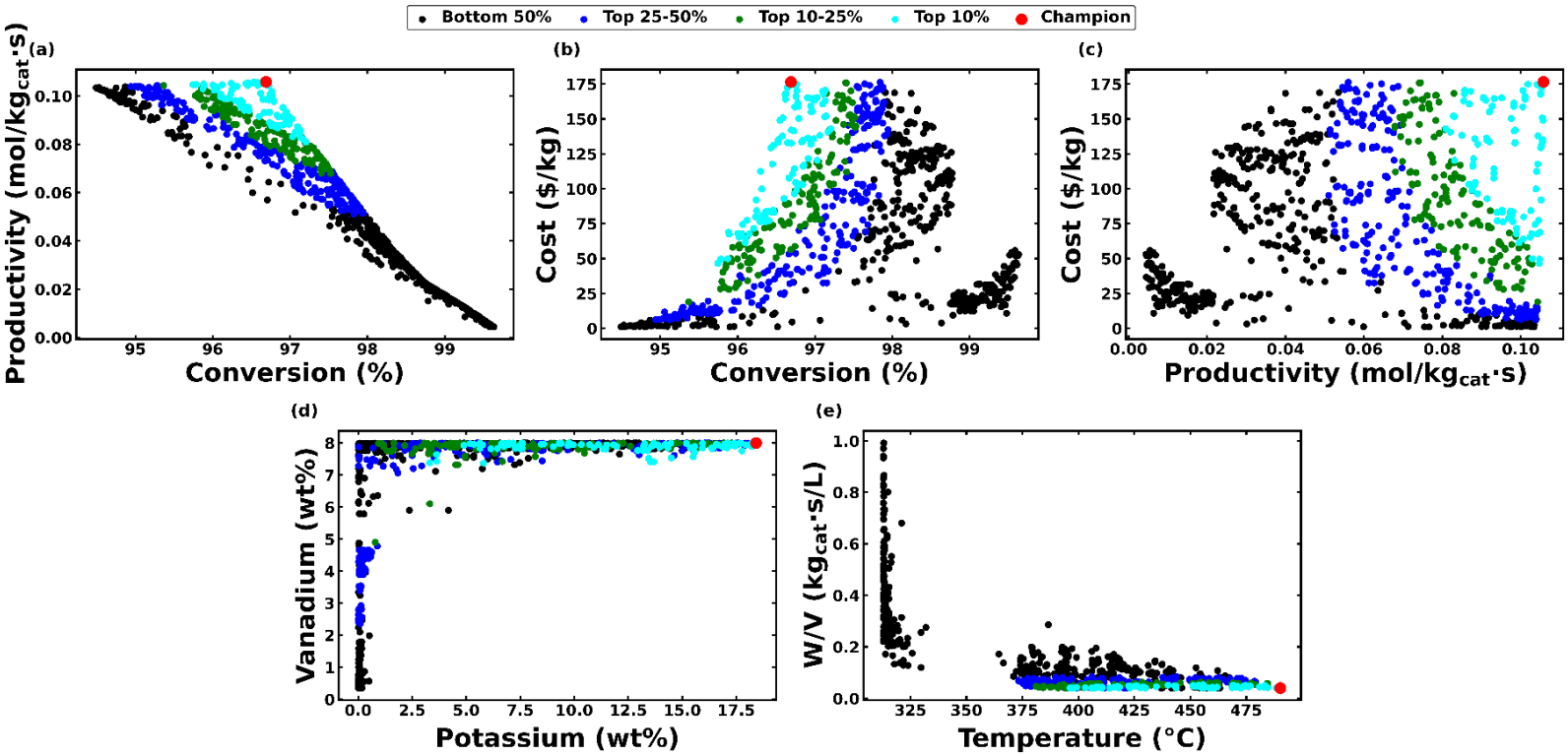


Figure 6.9. Results of MOO for vanadium-potassium case study and scenario 3 (Box-Cox transformer) (a) conversion vs productivity, (b) conversion vs cost, (c) productivity vs cost, (d) vanadium wt% vs potassium wt%, and (e) W/V vs temperature.

6.5.4 Scenario 4: Dataset with no data transformer

In the final scenario, no transformation was applied to any of the variables, and the ANN model was developed using the final data (removing outliers and performing normalization) in their original ranges. After that, the hyperparameters were carefully fine-tuned. The optimized ANN model was trained using the Adam optimizer with a learning rate of 0.0005 and a batch size of 32, and achieved minimum error at epoch 18 373. Figure 6.10 presents the parity plots for the training, validation, and testing datasets, clearly indicating that the ANN model exhibits strong performance and effectively generalizes to unseen data.

A preliminary examination of the Pareto domain for the scenario without any transformation techniques is presented in Figure 6.11, highlighting two notable patterns. First, conversion and productivity tend to increase concurrently. Second, a distinct conflict between these objectives is evident, particularly when analyzing the ranking partitions, where an increase in conversion leads to a corresponding decline in productivity. This inverse relationship is most prominently displayed on the right edge of the Pareto domain, where the highest conversion aligns with the lowest productivity levels. Figure 6.11b delves into the correlation between conversion percentage and catalyst cost within the Pareto-optimal solutions, suggesting that higher conversion and productivity require a larger quantity of catalyst. The plot clearly illustrates the tension between maximizing conversion and minimizing cost.

Figure 6.11c further explores the interplay between cost and productivity, with the right edge of the Pareto domain showing that reducing costs typically results in lower productivity. Figure 6.11d-e provide insights into decision variables associated with all Pareto-optimal solutions. Figure 6.11d presents the active phase weight fraction of vanadium and potassium, where higher concentrations of these metals correspond to increased conversion, productivity, and cost. Figure 6.11e examines the W/V ratio versus temperature, revealing that near the strike temperature of 400-425°C, a significant rise in the W/V ratio occurs as temperature decreases, indicating a greater catalyst requirement for achieving higher conversion and productivity. Additionally, the reaction rate diminishes at lower temperatures. Beyond this critical temperature, the W/V ratio stabilizes near its lower bound. It is crucial to remember that the SO_2 to SO_3 reaction is a reversible exothermic process, meaning that increasing the temperature beyond a certain point results in decreased conversion and productivity.

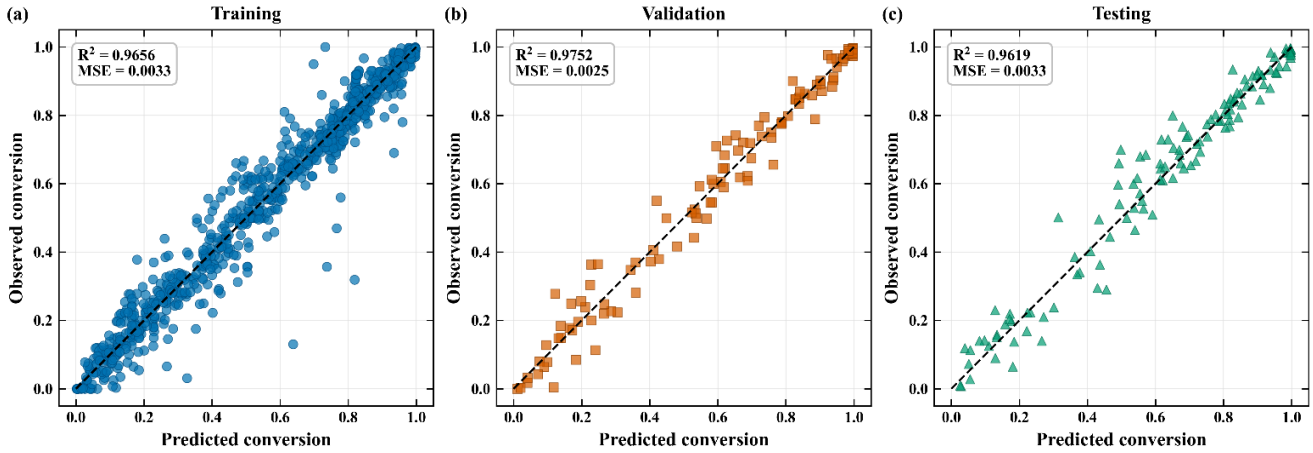


Figure 6.10. Parity plots of the ANN model for scenario 4 (Dataset without transformation).

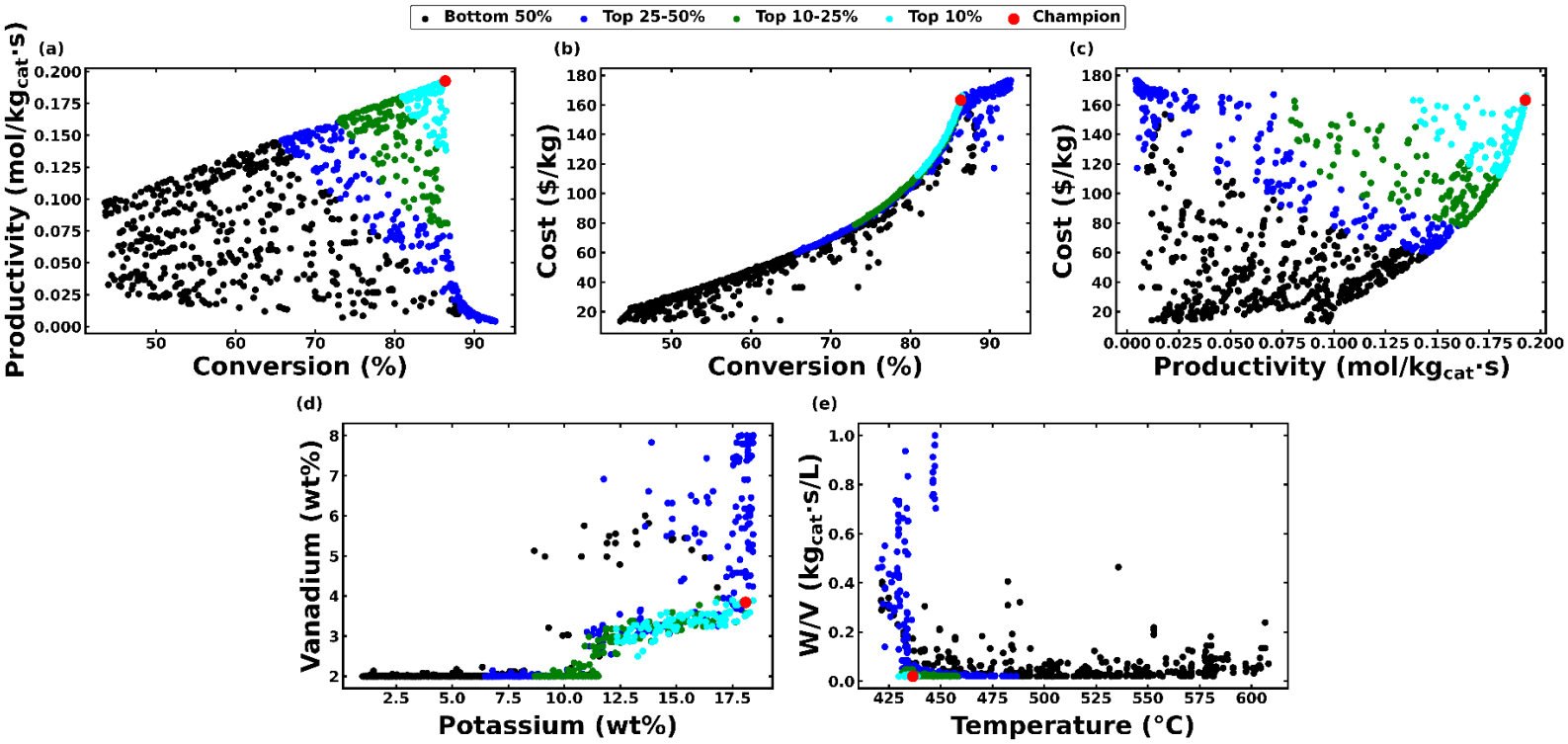


Figure 6.11. Results of MOO for vanadium-potassium case study and scenario 4 (Dataset without transformation) (a) conversion vs productivity, (b) conversion vs cost, (c) productivity vs cost, (d) vanadium wt% vs potassium wt%, and (e) W/V vs temperature.

Table 6.4 presents a summary of the key findings from the scenarios analyzed. This table aims to identify and propose the most appropriate candidate for the subsequent experiment. The initial section of Table 6.4 details the inputs that result in the highest conversion. The scenario without any transformation yields a slightly lower conversion of 93%, while all four scenarios report

identical productivity levels. To achieve maximum conversion, each method suggests maintaining the W/V ratio at its peak value. However, they diverge in their recommendations for temperature, as well as vanadium and potassium mass fractions. We recommend selecting an average temperature of approximately 420°C, which is close to the strike temperature and aligns with the scenario without transformation—deemed more favourable due to its smoother Pareto front and broader coverage of records. For vanadium and potassium, opting for the highest values appears logical, as these quantities result in the highest productivity levels, as shown in the subsequent sections of Table 6.4. In the following section of Table 6.4, the focus shifts to examining the highest productivity, with an average productivity of 0.165 mol $SO_3/kg_{cat}\cdot s$ across the four scenarios being a reasonable estimate. This level of productivity can be achieved with a W/V ratio of approximately 0.02 $kg_{cat}\cdot s/L$ in all scenarios, a potassium weight percentage of 18.42% in all scenarios, and a vanadium weight percentage of 8% in all scenarios except the one without transformation, which suggests that a value of 3.88% is sufficient for maximum productivity. The temperature range varies across scenarios, making the temperature from the no-transformation scenario a logical choice due to fewer gaps in the Pareto front. The conversion corresponding to the highest productivity would be an average of all scenarios, excluding the uniform quantile scenario, which deviates significantly from the others. Therefore, a conversion of 91% is proposed. The lowest-cost region in Table 6.4 yields catalyst costs of approximately 1 \$/kg in Scenarios 1–3 and 13.5 \$/kg in the no-transformation scenario. These values correspond to very low K (≈0.018–1 wt%) and V (≈0.35–2 wt%) loadings, with modest variations in W/V ratio. However, these economic gains significantly degrade performance: productivity drops from 0.20 to as low as 0.004 mol $SO_3/kg_{cat}\cdot s$, and conversion in the no-transformation scenario declines to 44%, far below the levels required for feasible operation. Therefore, despite their low cost, these solutions fail to provide an acceptable balance between economic and catalytic performance. Consequently, the "champion of champions" across the four scenarios would have a conversion of 91%, a productivity of 0.165 mol $SO_3/kg_{cat}\cdot s$, a cost of approximately 177 \$/kg, a temperature of 436°C (derived from the no-transformation scenario), a W/V ratio of 0.02 $kg_{cat}\cdot s/L$, a potassium content of 18.42 wt%, and a vanadium content of 8 wt%. These inputs form the foundation for selecting the next experiment, helping to refine the search space for exploring new catalyst formulations with desired outcomes.

Applying different transformation methods to the same dataset produced varying results, highlighting the significant impact of data preprocessing on catalyst optimization. Each transformation modified the statistical distribution of variables, consequently influencing the predicted outcomes for SO_2 conversion, productivity, and cost. These variations indicate that no single transformation method offers a comprehensive representation of the optimal conditions, as each approach emphasizes different characteristics of the dataset.

The uniform quantile transformation redistributed data evenly across its range, leading to high conversion values but lower productivity estimates, likely due to the stretching or compression of key parameters. The normal quantile transformation forced data into a normal distribution, which influenced cost estimations and modified parameter relationships, altering the trade-offs between conversion and productivity. The Box-Cox transformation adjusted skewed variables while preserving relationships, producing a more balanced outcome between conversion, productivity, and cost. The no-transformation scenario maintained the original data structure, resulting in a more interpretable parameter set but predicting a slightly lower optimal conversion.

These variations highlight the importance of considering multiple transformations rather than relying on a single approach. Since each method reshapes the data differently, the observed discrepancies in results reinforce the need for a broader analysis framework that integrates multiple perspectives. This ensures a more robust understanding of the dataset, reducing biases introduced by any single transformation and improving the reliability of catalyst optimization strategies.

6.6 Challenges and recommendations for development of ML models for catalytic processes

The application of ML models trained on literature data presents several challenges, primarily due to inconsistencies and varying methodologies employed by researchers when addressing specific problems. Therefore, it is crucial to recognize and understand these challenges, particularly in the emerging field of ML for catalyst discovery aimed at improving performance and optimizing characteristics. In this paper, we outlined several key issues encountered during our research:

A) Bias in literature data: Published scientific data in the field of heterogeneous catalytic reactions, as for many other fields, tend to disproportionately represent positive and successful outcomes, neglecting failed experiments that could significantly enrich the databank and lead to

more robust models. To address this issue, synthetic and logically derived data were incorporated into the databank. However, the resulting model performance was suboptimal, largely due to the limited size of the databank, where even minor disturbances may cause substantial misinterpretations.

Table 6.4. Highlights of the Pareto fronts obtained from the different scenarios.

	Variables	Scenario 1: Uniform quantile	Scenario 2: Normal quantile	Scenario 3: Box-Cox	Scenario 4: No transformation
Highest conversion	Conversion (%)	**99**	**99**	**99**	**93**
	Productivity (mol SO_3/kg_{cat}·s)	0.0046	0.0046	0.0044	0.0041
	Cost ($/kg)	43	158	53	177
	Temperature (°C)	370	540	313	447
	W/V (kg_{cat}·s/L)	0.97	0.97	0.99	1.00
	Potassium (wt%)	4.91	18.42	3.87	18.42
	Vanadium (wt%)	0.35	0.35	8.00	8.00
Highest productivity	Conversion (%)	89	66	97	87
	Productivity (mol SO_3/kg_{cat}·s)	**0.20**	**0.15**	**0.11**	**0.20**
	Cost ($/kg)	177	177	177	166
	Temperature (°C)	605	570	490	436
	W/V (kg_{cat}·s/L)	0.02	0.02	0.04	0.02
	Potassium (wt%)	18.42	18.42	18.42	18.42
	Vanadium (wt%)	8.00	8.00	8.00	3.88
Lowest cost	Conversion (%)	96	91	95	44
	Productivity (mol SO_3/kg_{cat}·s)	0.2	0.004	0.08	0.1
	Cost ($/kg)	**1**	**1**	**1**	**13.5**
	Temperature (°C)	390	526	462	504
	W/V (kg_{cat}·s/L)	0.2	0.97	0.04	0.02
	Potassium (wt%)	0.018	0.018	0.018	1.00
	Vanadium (wt%)	0.352	0.352	0.352	2.00
Champion	Conversion (%)	89	66	97	87
	Productivity (mol SO_3/kg_{cat}·s)	0.20	0.15	0.11	0.20
	Cost ($/kg)	177	177	177	163
	Temperature (°C)	605	570	490	436
	W/V (kg_{cat}·s/L)	0.02	0.02	0.04	0.02
	Potassium (wt%)	18.42	18.42	18.42	18.10
	Vanadium (wt%)	8.00	8.00	8.00	3.88

B) Impact of missing data: The absence of some critical data poses significant challenges to the construction of reliable models, as essential information—pertaining to catalyst properties or synthesis characteristics such as pore size, particle size, preparation methods, morphology,

calcination temperature, and specific surface area—had to be excluded. Attempts were made to predict these missing values using ML models, but due to very weak correlations among variables, the predictions lacked accuracy, necessitating the removal of this data from the databank.

C) Limited data range: For certain catalysts, the available data range was too limited to warrant meaningful inclusion in the databank. Initially, the databank contained 26 catalysts and promoters, but following two rounds of screening, only 14 candidates remained. Typically, training and testing datasets used for model evaluation share the same distribution. However, during the optimization process, the model often encounters new search spaces beyond its training experience. This extrapolation can reduce model reliability, suggesting that alternative approaches may be necessary to achieve more robust and dependable results.

D) Catalyst quantity and reaction equilibrium: An excessive amount of catalyst in reversible reactions does not necessarily result in higher conversion, as equilibrium may be reached with a smaller quantity. In experimental settings, excess catalysts are often used by default to ensure sufficient conversion, but this practice may introduce bias toward full conversion into the model.

E) SO_2 conversion data: In the catalytic conversion of SO_2 to SO_3, two distinct ranges of SO_2 mole fractions were observed—those relevant to desulfurization and those related to H_2SO_4 production. While both applications were incorporated into the databank, this study presents results from the latter. A more comprehensive understanding would require investigation of the desulfurization process, and future studies should examine these applications separately, comparing the outcomes.

F) Time-consuming data collection: The process of data collection is labour-intensive and time-consuming, as relevant data are scattered across various published papers. Researchers must diligently search for pertinent publications, carefully select those of value, and extract data sample by sample.

G) Non-standardized and unreliable data: Published data may not always reflect actual experimental conditions accurately, requiring researchers to rely heavily on the integrity of the data. Furthermore, there is no standardized protocol for reporting or collecting data, which introduces additional challenges in ensuring data reliability.

For future research and to enhance the application of ML in catalyst design and optimization (especially the catalytic conversion of SO_2 to SO_3), several key recommendations can be made:

A) Comprehensive data reporting: It is essential to report as much data as possible. This includes the active phase concerning the overall catalyst mass, catalyst preparation methods, catalyst properties such as void fraction and density, and other relevant parameters. Detailed data reporting will significantly improve the quality and accuracy of ML models, leading to better predictions and optimizations.

B) Data sharing and accessibility: Making data more accessible is crucial. Currently, in ML-based studies, supporting materials are often included as additional documents. This practice should be adopted in research on the catalytic conversion of SO_2, with supplementary materials providing detailed experimental data. Additionally, publishing data in formats like Excel spreadsheets alongside supporting documents would facilitate easier data collection and usage by other researchers.

C) Standardization of experimental data collection: To ensure consistency and reliability across different studies, there could be a standardized procedure for experimental data collection. Establishing a coherent and uniform approach would lead to more robust data reporting, which would reduce missing values and increase the overall integrity of the dataset. This standardization would enable researchers to better use ML models by working with high-quality, consistent data.

D) Need for more experimental data: To develop more reliable ML models, additional experiments are required to generate larger datasets. These expanded datasets would enhance the model's ability to make reliable predictions and enable the use of optimization techniques for the discovery of new catalysts. A larger volume of data would improve the extrapolation capabilities of the models, leading to more innovative breakthroughs in catalyst design.

In conclusion, future research should focus on detailed data reporting, promoting data accessibility, standardizing experimental procedures, and generating more experimental data to advance the integration of ML in catalyst optimization and discovery. This aligns with the growing trend among scientific journals requiring data availability statements, supporting materials with additional data, and access to raw data to ensure transparency and reproducibility in research. Linking these efforts to existing guidelines, such as those outlined by Elsevier for data availability

statements [73], further emphasizes the importance of adopting robust data-sharing practices to meet current publishing standards and foster trust in ML-driven discoveries. Moreover, to further inform the next steps in both experimental and modelling research, a systematic approach to integrating experimental findings into ML frameworks should be explored. This includes developing hybrid modelling techniques that combine first-principles simulations with ML algorithms to enhance predictive accuracy. Additionally, establishing benchmark datasets and validation protocols would allow for consistent model assessment across different studies. Collaboration between experimentalists and data scientists should also be strengthened to ensure that ML models are trained on high-quality, domain-relevant data, ultimately leading to more efficient catalyst design and optimization strategies.

6.7 Conclusion

This study has demonstrated the potential of integrating ML techniques with MOO for catalyst design and process optimization in SO_2 oxidation. By employing rigorous data preprocessing, ANN modelling, and MOO, a robust methodology has been developed to identify promising catalyst formulations and operating conditions. The research has yielded several notable outcomes. High-performing ANN models were developed, accurately predicting SO_2 conversion with strong correlations observed between predicted and actual values. The successful application of MOO elucidated the complex trade-offs between conversion, productivity, and cost in SO_2 oxidation catalysis, providing critical insights into the optimization process.

The NFM technique was utilized to rank solutions based on expert preferences. Optimal ranges for critical variables were identified, including a temperature range of approximately 420–436°C, near the reaction's strike temperature; a W/V ratio of 0.02–1.0 kg_{cat}·s/L, depending on the optimization priority; vanadium content between 3.88–8.00 wt%, with higher values generally improving performance; and potassium content consistently high at 18.10–18.42 wt% across different scenarios. High-performance benchmarks were achieved, with maximum SO_2 conversion reaching up to 99% under optimal conditions, the highest productivity recorded at 0.2 mol SO_3/kg_{cat}·s, and the resulting operating configurations (temperature, W/V ratio, and K/V loadings) achieving 87–97% conversion with productivity of 0.11–0.20 mol SO_3/kg_{cat}·s representing an optimal cost-performance ratio.

The study further demonstrated the interplay between catalyst composition and reaction conditions, emphasizing the critical role of temperature and W/V ratio in balancing conversion and productivity. These findings provide valuable insights for catalyst design and process optimization in SO_2 oxidation. The methodology developed in this research, combining data-driven modelling with MOO and MCDM, offers a powerful tool for navigating the complex parameter space of catalytic reactions. This approach not only aids in optimizing existing catalytic systems but also establishes a framework for the rational design of new catalysts with enhanced performance characteristics. The insights and methodologies developed here pave the way for significant advancements in industrial processes relying on SO_2 to SO_3 conversion, contributing to more efficient and effective catalyst design strategies.

6.8 References

1. Dobbelaere, M. R., Plehiers, P. P., Van de Vijver, R., Stevens, C. V., & Van Geem, K. M. (2021). Machine learning in chemical engineering: Strengths, weaknesses, opportunities, and threats. *Engineering*, *7*(9), 1201–1211. https://doi.org/10.1016/j.eng.2021.03.019
2. Stokes, G. G. (1848). On the steady motion of incompressible fluids. *Transactions of the Cambridge Philosophical Society*, *7*, 439.
3. Thebelt, A., Wiebe, J., Kronqvist, J., Tsay, C., & Misener, R. (2022). Maximizing information from chemical engineering data sets: Applications to machine learning. *Chemical Engineering Science*, *252*, 117469. https://doi.org/10.1016/j.ces.2022.117469
4. Lee, J. H., Shin, J., & Realff, M. J. (2018). Machine learning: Overview of the recent progresses and implications for the process systems engineering field. *Computers & Chemical Engineering*, *114*, 111–121. https://doi.org/10.1016/j.compchemeng.2017.10.008
5. Guan, Y., Chaffart, D., Liu, G., Tan, Z., Zhang, D., Wang, Y., Li, J., & Ricardez-Sandoval, L. (2022). Machine learning in solid heterogeneous catalysis: Recent developments, challenges and perspectives. *Chemical Engineering Science*, *248*, 117224. https://doi.org/10.1016/j.ces.2021.117224
6. Goldsmith, B. R., Esterhuizen, J., Liu, J.-X., Bartel, C. J., & Sutton, C. (2018). *Machine learning for heterogeneous catalyst design and discovery*. *64*(7), 2311–2323. https://doi.org/10.1002/aic.16198

7. Srivastava, R. K., Jozewicz, W., & Singer, C. (2001). SO_2 scrubbing technologies: A review. *Environmental Progress*, *20*(4), 219–228. https://doi.org/10.1002/ep.670200410
8. Gound, T. U., Ramachandran, V., & Kulkarni, S. (2014). Various methods to reduce SO_2 emission - A review. *International Journal of Ethics In Engineering & Management Education*, *1*(1), 1–6.
9. Hanif, M. A., Ibrahim, N., & Abdul Jalil, A. (2020). Sulfur dioxide removal: An overview of regenerative flue gas desulfurization and factors affecting desulfurization capacity and sorbent regeneration. *Environmental Science and Pollution Research*, *27*(22), 27515–27540. https://doi.org/10.1007/s11356-020-09191-4
10. Lum, M. M. X., Ng, K. H., Lai, S. Y., Mohamed, A. R., Alsultan, A. G., Taufiq-Yap, Y. H., Koh, M. K., Mohamed, M. A., Vo, D.-V. N., & Subramaniam, M. (2023). Sulfur dioxide catalytic reduction for environmental sustainability and circular economy: A review. *Process Safety and Environmental Protection*, *176*, 580–604. https://doi.org/10.1016/j.psep.2023.06.035
11. Loskyll, J., Stöwe, K., & Maier, W. F. (2013). Search for New Catalysts for the Oxidation of SO2. *ACS Combinatorial Science*, *15*(9), 464–474. https://doi.org/10.1021/co400023j
12. Wang, X., Wang, A., Li, N., Wang, X., Liu, Z., & Zhang, T. (2006). Catalytic reduction of SO_2 with CO over supported iron catalysts. *Industrial & Engineering Chemistry Research*, *45*(13), 4582–4588. https://doi.org/10.1021/ie0600947
13. Wilburn, M. S., & Epling, W. S. (2018). Formation and decomposition of sulfite and sulfate species on Pt/Pd catalysts: An SO_2 oxidation and sulfur exposure study. *ACS Catalysis*, *9*(1), 640–648. https://doi.org/10.1021/acscatal.8b03529
14. Du, X., Xue, J., Wang, X., Chen, Y., Ran, J., & Zhang, L. (2018). Oxidation of sulfur dioxide over V_2O_5/TiO_2 catalyst with low vanadium loading: A theoretical study. *The Journal of Physical Chemistry C*, *122*(8), 4517–4523. https://doi.org/10.1021/acs.jpcc.8b00296
15. Islam, A., Teo, S. H., Ng, C. H., Taufiq-Yap, Y. H., Choong, S. Y. T., & Awual, M. R. (2022). Progress in recent sustainable materials for greenhouse gas (NO_x and SO_x) emission mitigation. *Progress in Materials Science*, *132*, 101033. https://doi.org/10.1016/j.pmatsci.2022.101033

16. Nikiforova, A., Kozhura, O., & Pasenko, O. (2016). Leaching of vanadium by sulfur dioxide from spent catalysts for sulfuric acid production. *Hydrometallurgy*, *164*, 31–37. https://doi.org/10.1016/j.hydromet.2016.05.004
17. García-Labiano, F., Diego, L. F. de, Cabello, A., Gayán, P., Abad, A., Adánez, J., & Sprachmann, G. (2016). Sulphuric acid production via Chemical Looping Combustion of elemental sulphur. *Applied Energy*, *178*, 736–745. https://doi.org/10.1016/j.apenergy.2016.06.110
18. Vangapally, N., Penki, T. R., Elias, Y., Muduli, S., Maddukuri, S., Luski, S., Aurbach, D., & Martha, S. K. (2023). Lead-acid batteries and lead–carbon hybrid systems: A review. *Journal of Power Sources*, *579*, 233312. https://doi.org/10.1016/j.jpowsour.2023.233312
19. Kutney, G. (2023). *Sulfur: History, technology, applications and industry*. ChemTec Publishing, Elsevier.
20. Moeller, W., & Winkler, K. (1968). The double contact process for sulfuric acid production. *Journal of the Air Pollution Control Association*, *18*(5), 324–325. https://doi.org/10.1080/00022470.1968.10469134
21. Kiss, A. A., Bildea, C. S., & Grievink, J. (2010). Dynamic modeling and process optimization of an industrial sulfuric acid plant. *Chemical Engineering Journal*, *158*(2), 241–249. https://doi.org/10.1016/j.cej.2010.01.023
22. Oni, A. O., Fadare, D. A., Sharma, S., & Rangaiah, G. P. (2018). Multi-objective optimisation of a double contact double absorption sulphuric acid plant for cleaner operation. *Journal of Cleaner Production*, *181*, 652–662. https://doi.org/10.1016/j.jclepro.2018.01.239
23. Li, H., Zhang, Z., & Liu, Z. (2017). Application of artificial neural networks for catalysis: A review. *Catalysts*, *7*(10), 306. https://doi.org/10.3390/catal7100306
24. Semnani, P., Bogojeski, M., Bley, F., Zhang, Z., Wu, Q., Kneib, T., Herrmann, J., Weisser, C., Patcas, F., & Müller, K.-R. (2024). A Machine Learning and Explainable AI Framework Tailored for Unbalanced Experimental Catalyst Discovery. *The Journal of Physical Chemistry C*, *128*(50), 21349–21367. https://doi.org/10.1021/acs.jpcc.4c05332
25. Li, Z., Wang, S., Chin, W. S., Achenie, L. E., & Xin, H. (2017). High-throughput screening of bimetallic catalysts enabled by machine learning. *Journal of Materials Chemistry A*, *5*(46), 24131–24138. https://doi.org/10.1039/C7TA01812F

26. McCullough, K., Williams, T., Mingle, K., Jamshidi, P., & Lauterbach, J. (2020). High-throughput experimentation meets artificial intelligence: A new pathway to catalyst discovery. *Physical Chemistry Chemical Physics*, *22*(20), 11174–11196. https://doi.org/10.1039/D0CP00972E
27. Cao, Y., Taghvaie Nakhjiri, A., & Ghadiri, M. (2024). Different applications of machine learning approaches in materials science and engineering: Comprehensive review. *Engineering Applications of Artificial Intelligence*, *135*, 108783. https://doi.org/10.1016/j.engappai.2024.108783
28. Choung, S., Park, W., Moon, J., & Han, J. W. (2024). Rise of machine learning potentials in heterogeneous catalysis: Developments, applications, and prospects. *Chemical Engineering Journal*, *494*, 152757. https://doi.org/10.1016/j.cej.2024.152757
29. Mace, S., Xu, Y., & Nguyen, B. N. (2024). Automated Transition Metal Catalysts Discovery and Optimisation with AI and Machine Learning. *ChemCatChem*, *16*(10), e202301475. https://doi.org/10.1002/cctc.202301475
30. Schlexer Lamoureux, P., Winther, K. T., Garrido Torres, J. A., Streibel, V., Zhao, M., Bajdich, M., Abild-Pedersen, F., & Bligaard, T. (2019). Machine learning for computational heterogeneous catalysis. *ChemCatChem*, *11*(16), 3581–3601. https://doi.org/10.1002/cctc.201900595
31. Erdem Günay, M., & Yıldırım, R. (2021). Recent advances in knowledge discovery for heterogeneous catalysis using machine learning. *Catalysis Reviews*, *63*(1), 120–164. https://doi.org/10.1080/01614940.2020.1770402
32. Günay, M. E., & Yildirim, R. (2011). Neural network Analysis of Selective CO Oxidation over Copper-Based Catalysts for Knowledge Extraction from Published Data in the Literature. *Industrial & Engineering Chemistry Research*, *50*(22), 12488–12500. https://doi.org/10.1021/ie2013955
33. Kite, S., Hattori, T., & Murakami, Y. (1994). Estimation of catalytic performance by neural network—Product distribution in oxidative dehydrogenation of ethylbenzene. *Applied Catalysis A: General*, *114*(2), L173–L178. https://doi.org/10.1016/0926-860X(94)80169-X
34. Günay, M. E., & Yildirim, R. (2013). Knowledge Extraction from Catalysis of the Past: A Case of Selective CO Oxidation over Noble Metal Catalysts between 2000 and 2012. *ChemCatChem*, *5*(6), 1395–1406. https://doi.org/10.1002/cctc.201200665

35. Cavalcanti, F. M., Schmal, M., Giudici, R., & Alves, R. M. B. (2019). A catalyst selection method for hydrogen production through Water-Gas Shift Reaction using artificial neural networks. *Journal of Environmental Management*, *237*, 585–594. https://doi.org/10.1016/j.jenvman.2019.02.092
36. Odabaşı, Ç., Günay, M. E., & Yıldırım, R. (2014). Knowledge extraction for water gas shift reaction over noble metal catalysts from publications in the literature between 2002 and 2012. *International Journal of Hydrogen Energy*, *39*(11), 5733–5746. https://doi.org/10.1016/j.ijhydene.2014.01.160
37. Kim, C., & Kim, J. (2022). Machine learning-based high-throughput screening, strategical design and knowledge extraction of $Pt/Ce_XZr_{1-X}O_2$ catalysts for water gas shift reaction. *International Journal of Energy Research*, *46*(15), 21293–21308. https://doi.org/10.1002/er.8488
38. Mine, S., Takao, M., Yamaguchi, T., Toyao, T., Maeno, Z., Hakim Siddiki, S. M. A., Takakusagi, S., Shimizu, K., & Takigawa, I. (2021). Analysis of Updated Literature Data up to 2019 on the Oxidative Coupling of Methane Using an Extrapolative Machine-Learning Method to Identify Novel Catalysts. *ChemCatChem*, *13*(16), 3636–3655. https://doi.org/10.1002/cctc.202100495
39. Suzuki, K., Toyao, T., Maeno, Z., Takakusagi, S., Shimizu, K., & Takigawa, I. (2019). Statistical Analysis and Discovery of Heterogeneous Catalysts Based on Machine Learning from Diverse Published Data. *ChemCatChem*, *11*(18), 4537–4547. https://doi.org/10.1002/cctc.201900971
40. Baysal, M., Günay, M. E., & Yıldırım, R. (2017). Decision tree analysis of past publications on catalytic steam reforming to develop heuristics for high performance: A statistical review. *International Journal of Hydrogen Energy*, *42*(1), 243–254. https://doi.org/10.1016/j.ijhydene.2016.10.003
41. Şener, A. N., Günay, M. E., Leba, A., & Yıldırım, R. (2018). Statistical review of dry reforming of methane literature using decision tree and artificial neural network analysis. *Catalysis Today*, *299*, 289–302. https://doi.org/10.1016/j.cattod.2017.05.012
42. Tripathi, K., Gupta, V., Awasthi, V., Pant, K. K., & Upadhyayula, S. (2023). Forecasting Catalytic Property-Performance Correlations for CO_2 Hydrogenation to Methanol via

Surrogate Machine Learning Framework. *Advanced Sustainable Systems*, *7*(3), 2200416. https://doi.org/10.1002/adsu.202200416

43. Chen, Y., Li, R., Suo, H., & Liu, C. (2021). Evaluation of a Data-Driven, Machine Learning Approach for Identifying Potential Candidates for Environmental Catalysts: From Database Development to Prediction. *ACS ES&T Engineering*, *1*(8), 1246–1257. https://doi.org/10.1021/acsestengg.1c00125
44. Lu, M., Gao, F., Tan, Y., Yi, H., Gui, Y., Xu, Y., Wang, Y., Zhou, Y., Tang, X., & Chen, L. (2024). Knowledge-Driven Experimental Discovery of Ce-Based Metal Oxide Composites for Selective Catalytic Reduction of NO_x with NH_3 through Interpretable Machine Learning. *ACS Applied Materials & Interfaces*, *16*(3), 3593–3604. https://doi.org/10.1021/acsami.3c18490
45. Toyao, T., Maeno, Z., Takakusagi, S., Kamachi, T., Takigawa, I., & Shimizu, K. (2020). Machine Learning for Catalysis Informatics: Recent Applications and Prospects. *ACS Catalysis*, *10*(3), 2260–2297. https://doi.org/10.1021/acscatal.9b04186
46. Medford, A. J., Kunz, M. R., Ewing, S. M., Borders, T., & Fushimi, R. (2018). Extracting Knowledge from Data through Catalysis Informatics. *ACS Catalysis*, *8*(8), 7403–7429. https://doi.org/10.1021/acscatal.8b01708
47. Agin, F., Fauteux-Lefebvre, C., & Thibault, J. (2025). Optimal Design of Catalytic Conversion of SO_2 to SO_3 via Machine Learning. *Journal of Machine Intelligence and Data Science (JMIDS)*, *6*(1), 1–22. https://doi.org/10.11159/jmids.2025.001
48. Boucheikhchoukh, A., Thibault, J., & Fauteux-Lefebvre, C. (2020). Catalyst design using artificial intelligence: SO_2 to SO_3 case study. *The Canadian Journal of Chemical Engineering*, *98*(9), 2016–2031. https://doi.org/10.1002/cjce.23756
49. Trinh, C., Tbatou, Y., Lasala, S., Herbinet, O., & Meimaroglou, D. (2023). On the Development of Descriptor-Based Machine Learning Models for Thermodynamic Properties: Part 1—From Data Collection to Model Construction: Understanding of the Methods and Their Effects. *Processes*, *11*(12), 3325. https://doi.org/10.3390/pr11123325
50. García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: Methods and prospects. *Big Data Analytics*, *1*(1), 1–22. https://doi.org/10.1186/s41044-016-0014-0

51. Jia, W., Sun, M., Lian, J., & Hou, S. (2022). Feature dimensionality reduction: A review. *Complex & Intelligent Systems*, *8*(3), 2663–2693. https://doi.org/10.1007/s40747-021-00637-x
52. Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, *1*(1), 56–70. https://doi.org/10.38094/jastt1224
53. Khaire, U. M., & Dhanalakshmi, R. (2022). Stability of feature selection algorithm: A review. *Journal of King Saud University-Computer and Information Sciences*, *34*(4), 1060–1073. https://doi.org/10.1016/j.jksuci.2019.06.012
54. Agin, F., Thibault, J., & Fauteux-Lefebvre, C. (2024). Autoassociative neural network for missing data imputation: A case study via the styrene production process. *The Canadian Journal of Chemical Engineering*, *103*, 339–358. https://doi.org/10.1002/cjce.25377
55. Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, *4*(11), e00938. https://doi.org/10.1016/j.heliyon.2018.e00938
56. Gurney, K. (2018). *An introduction to neural networks*. CRC Press.
57. Trinh, C., Meimaroglou, D., & Hoppe, S. (2021). Machine learning in chemical product engineering: The state of the art and a guide for newcomers. *Processes*, *9*(8), 1456. https://doi.org/10.3390/pr9081456
58. Ojha, V. K., Abraham, A., & Snášel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, *60*, 97–116.
59. Yu, T., & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. *arXiv Preprint arXiv:2003.05689*. https://doi.org/10.48550/arXiv.2003.05689
60. Kumar, A., & Flores-Cerrillo, J. (2022). *Machine Learning in Python for Process Systems Engineering: Achieve Operational Excellence Using Process Data*. MLforPSE.
61. Xu, Y., & Goodacre, R. (2018). On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of Analysis and Testing*, *2*(3), 249–262. https://doi.org/10.1007/s41664-018-0068-2

62. Plevris, V., Solorzano, G., Bakas, N. P., & Ben Seghier, M. E. A. (2022). Investigation of performance metrics in regression analysis and machine learning-based prediction models. *8th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2022)*. https://doi.org/10.23967/eccomas.2022.155
63. Bui, L. T., & Alam, S. (2008). *Multi-objective optimization in computational intelligence: Theory and practice*. IGI Global.
64. Cáceres Sepúlveda, G. (2019). *Relevance of multi-objective optimization in the chemical engineering field* [Master thesis, University of Ottawa]. https://doi.org/10.20381/ruor-24026
65. Gunantara, N. (2018). A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, *5*(1), 1502242. https://doi.org/10.1080/23311916.2018.1502242
66. Deb, K. (2011). *Multi-objective optimisation using evolutionary algorithms: An introduction*. Springer.
67. *Metals on Bloomberg*. (2023). Bloomberg Terminal
68. *World Bank Commodity Markets*. (2023). https://www.worldbank.org/en/research/commodity-markets
69. Zaker, M. R., Fauteux-Lefebvre, C., & Thibault, J. (2021). Modelling and Multi-Objective Optimization of the Sulphur Dioxide Oxidation Process. *Processes*, *9*(6), 1072. https://doi.org/10.3390/pr9061072
70. Chanal, D., Steiner, N. Y., Chamagne, D., & Pera, M.-C. (2021). Impact of standardization applied to the diagnosis of LT-PEMFC by Fuzzy C-Means clustering. *2021 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 1–6. https://doi.org/10.1109/VPPC53923.2021.9699234
71. Sefara, T. J. (2019). The Effects of Normalisation Methods on Speech Emotion Recognition. *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 1–8. https://doi.org/10.1109/IMITEC45504.2019.9015895
72. Xiong, B., Chen, Y., Chen, D., Fu, J., & Zhang, D. (2025). Deep probabilistic solar power forecasting with Transformer and Gaussian process approximation. *Applied Energy*, *382*, 125294. https://doi.org/10.1016/j.apenergy.2025.125294
73. *Elsevier Data Availability Statement Guidelines, [online]*. (2024, December 17). https://scientific-publishing.webshop.elsevier.com/manuscript-preparation/data-availability-statement/

Chapter 7

A Data-Driven Approach to Competitive Hydrogenation: Structure-Informed Machine Learning with Chemoselectivity-Based Optimization

Farough Agin[a], Kostiantyn Prystupa[b], Stephen Newman[b], Jules Thibault[a], and Clémence Fauteux-Lefebvre[a]*

[a]Department of Chemical and Biological Engineering

[b]Department of Chemistry and Biomolecular Sciences

University of Ottawa, Ottawa, Ontario, K1N 6N5, Canada

*Corresponding Author: Clémence Fauteux-Lefebvre

This chapter is a manuscript under preparation to be published.

7.1 Abstract

Selective hydrogenation often involves competing hydrogenation pathways of different functional groups, where catalyst, reactive environment and operating conditions jointly determine whether the desired transformation is favoured. In this study, we develop and benchmark data-driven models to predict conversion and yield for a hydrogenation dataset comprising 10 substrates and 10 catalysts evaluated over a range of temperatures, pressures, and flow rates. Two preprocessing strategies were investigated: (i) categorical representations using one-hot encoding for both substrates and catalysts, and (ii) a structure-informed representation combining RDKit-derived substrate descriptors with one-hot catalyst encoding. In the latter case, descriptor dimensionality was reduced via random forest (RF) feature importance to yield a compact set applied uniformly across all prediction targets. A total of sixteen machine learning (ML) models were trained to predict conversion and yield as separate targets, with hyperparameters optimized via cross-validation and performance assessed using a leave-one-experiment-out protocol that holds out an entire substrate–catalyst pair to evaluate generalization to previously unseen chemistry combinations. Predictive accuracy was found to depend strongly on the choice of chemical representation, with descriptor-based substrate features providing substantial improvements in out-of-sample performance relative to purely categorical encodings. The best-performing trained

predictors were then integrated within a genetic algorithm (GA) formulated to maximize a chemoselectivity objective defined as the difference between the desired yield and undesired conversion across ordered substrate pairs. This integration produced an ordered-pair selectivity landscape together with corresponding catalyst recommendations. Finally, k-means clustering of GA-optimal operating conditions identified a small number of recurring temperature–pressure–flow rate regimes linked to high chemoselectivity, providing an interpretable summary of how optimal operating sets vary across competing substrate types. Collectively, the proposed workflow integrates predictive modelling and optimization to deliver actionable guidance for catalyst and reaction condition selection in competitive hydrogenation systems.

Keywords: selective hydrogenation; chemoselectivity; machine learning; genetic algorithm; operating condition clustering

7.2 Introduction

Transition-metal-catalyzed hydrogenation is a foundational reaction in chemical synthesis, having high importance for the selective reduction of unsaturated functionalities in pharmaceuticals, fine chemicals, polymers, agrochemicals, and bio-derived intermediates [1]. Its widespread applicability stems from the ability of metal catalysts to activate H_2 and mediate diverse reaction pathways with high intrinsic activity. However, as molecular architectures become increasingly complex and polyfunctional, achieving chemoselectivity (preferential reduction of one functional group in the presence of others) has emerged as a persistent and technically demanding challenge. Chemoselective outcomes reflect a subtle interplay among adsorption strength, transition-state energetics, hydrogen availability, and the geometric and electronic structure of catalytic sites [2–4]. In practice, heterogeneous catalysts exhibit substantial structural heterogeneity, presenting terraces, edges, steps, and defect sites that can evolve dynamically under reaction conditions [5]. As a result, selectivity becomes highly sensitive to subtle variations in catalyst structure, support interactions, and reaction environment.

The inherent difficulty of rationalizing chemoselective hydrogenation is well documented across diverse reaction types. In the hydrogenation of halonitroarene, for example, the desired reduction of the nitro group must proceed in the presence of reducible aromatic or halide groups [6]. Conventional Pt, Ru, and Ni catalysts often promote undesired over-hydrogenation, whereas tailored systems such as TiO_x-modified nanoparticles can shift selectivity by altering adsorption motifs [7]. Comparable complexity is observed in heteroarene hydrogenation, where heteroatom coordination and aromatic stabilization exert strong control over pathway preference [8]. Similarly, in the reduction of α,β-unsaturated carbonyl compounds, the competition between C=C and C=O hydrogenation is governed by a coupled dependence of metal identity, support acidity, and solvent effects, each of which modulates selectivity [1]. Biomass-derived substrates such as levulinic acid introduce additional layers of complexity due to solvent-mediated transformations and intramolecular rearrangements [9]. Collectively, these examples illustrate that chemoselectivity is highly system-specific and, as a result, broad mechanistic generalizations remain difficult to establish.

Hydrogenation catalysts encompass a broad spectrum of material classes across both homogeneous and heterogeneous systems, each offering distinct advantages and limitations [10–12].

Homogeneous catalysts provide well-defined molecular active sites and can deliver exceptional chemo-, regio-, and enantioselectivity through precise ligand control. However, they also generate vast combinatorial design spaces that are difficult to explore experimentally and may require more demanding purification to minimize residual metal and ligand content [13,14]. In contrast, heterogeneous catalysts are attractive due to their ease of separation and reuse, compatibility with continuous processing, and the breadth of tunable parameters, including particle size, support interactions, alloying, and interfacial design, that enable fine control over performance. These features have made supported nanoparticles, alloys, intermetallic phases, and single-atom catalysts as central platforms in selective hydrogenation research [1,15]. Industrial hydrogenation processes introduce additional layers of complexity due to multiphase transport limitations, high-pressure effects, and catalyst restructuring under operational conditions [16–18]. Collectively, the multidimensional nature of catalyst architecture, substrate structure, and process variables renders the discovery of chemoselective hydrogenation catalysts highly challenging when relying largely on intuition-driven or trial-and-error approaches. As a result, data-driven methodologies based on machine learning (ML) have emerged as powerful tools for linking catalyst composition, electronic structure, and reaction conditions to hydrogenation performance across transition-metal systems [19].

Growing recognition of these limitations has motivated increasing interest in ML, which enables the extraction of structure-reactivity relationships from experimental or computational datasets [20,21]. In homogeneous asymmetric hydrogenation, ML models have successfully correlated steric and electronic ligand descriptors with enantioselectivity, thereby accelerating ligand optimization and reducing reliance on exhaustive screening [14]. The emergence of large, curated datasets for asymmetric hydrogenation reactions has supported hierarchical learning strategies capable of generalizing across substrate and catalyst families [22]. ML has also been widely applied to heterogeneous hydrogenation, where models trained on nanoparticle size, morphology, alloy composition, and support effects have provided insight into active-site motifs and selectivity trends [1,5]. In the context of biomass hydrogenation, ML has been integrated with the genetic algorithm (GA) to optimize catalyst-condition combinations for reactions such as the conversion of levulinic acid to γ-valerolactone [9]. Additional studies have used artificial neural networks (ANNs), random forests (RF), and boosted models to demonstrate strong predictive performance for ester, ketone, carbonyl, and nitro reductions, capturing nonlinear effects of

molecular structure and process variables [14,17,23,24]. Beyond these specific applications, broader ML frameworks have examined descriptor selection, learning under sparse data regimes, and the integration of ML with high-throughput experimentation [10,11,25].

Together, these studies demonstrate that ML can successfully model diverse aspects of hydrogenation chemistry, including reaction yields, rates, and stereoselectivities, while also elucidating the molecular and operational features that govern catalytic performance. Despite this progress, most existing ML works remain confined to narrow reaction classes or catalyst families. As a result, they do not address the broader challenge of chemoselective hydrogenation of multifunctional substrates, where multiple reducible groups compete under identical conditions. The present work presents the development of a ML framework to inform chemoselective hydrogenation across diverse catalysts, substrates, and reaction conditions using an experimental hydrogenation dataset. Data were generated using the H-Cube continuous-flow hydrogenation platform [26]. Using this platform, 500 experimental hydrogenation data points were generated and curated for subsequent modelling. Two complementary representative strategies are constructed, combining one-hot encoding with cheminformatics-derived substrate descriptors. These representations are systematically evaluated across sixteen ML models, and the optimal model was selected based on predictive accuracy and generalization performance. The final model was subsequently integrated with a GA to identify catalyst-condition combinations that maximize chemoselectivity, thereby enabling data-driven exploration and optimization of hydrogenation reactions. To further interpret the optimized solutions, principal component analysis (PCA) and k-means clustering were applied to the GA-derived optima, providing an interpretable summary of how high-chemoselectivity solutions are organized in the temperature–pressure–flow-rate space. Overall, the framework is designed not only to estimate reaction outcomes but also to support catalyst selection, guide experimental planning, and extract interpretable meaningful patterns from optimized reaction conditions.

7.3 Methodology

7.3.1 Problem statement, experimental setup, and dataset

Several previous studies on catalyst selection have relied on large, publicly available reaction databases, such as Reaxys and Scifinder, while others have analyzed proprietary reaction data generated within the pharmaceutical industry [27–30]. In both cases, the resulting datasets are

strongly biased toward successful, high-yielding reactions and "best-in-class" reaction conditions. In contrast, achieving chemoselective hydrogenation requires identifying a catalyst that is highly reactive toward the desired functional group while exhibiting minimal or no reactivity toward other functional groups that must be preserved. However, public datasets largely lack the low-yielding or unproductive reactions that are essential for understanding and modelling chemoselectivity. This limitation motivated us to design an unbiased dataset that captures diverse chemical space rather than focusing exclusively on optimal reaction conditions.

We employed a continuous-flow hydrogenation platform (H-Cube) to obtain experimental data allowing for catalyst-substrate performance modelling. This flow system provides control over key reaction parameters, including temperature, pressure and residence time, owing to its enhanced heat and mass transfer characteristics relative to batch reactors [31]. The use of sealed, pre-packed catalyst cartridges further enables continuous introduction and collection of reaction mixtures while avoiding direct handling of pyrophoric heterogeneous catalysts. The functional groups selected for this study reflect their prevalence in medicinal and process chemistry, such as nitro and alkene reductions and debenzylation reactions, as well as their importance as diversification handles, including ketones, aryl chlorides and bromides [32]. Aromatic ring reductions (pyridine, pyrrole, naphthalene) were also included to capture structural motifs commonly found in bioactive molecules, which often contain both the aromatic and functional moieties [33]. Investigating model substrates bearing a single functional group enables extension of the results to more complex molecules with multiple functionalities. This strategy parallels fragment-based screening approaches widely used in drug discovery and reaction development [34,35]. Hydrogenation reactions were carried out using a diverse set of metal catalysts on various supports, spanning highly reactive systems (Pd-based catalysts), moderately reactive catalysts (Pt/C, Ru/C, Raney-Ni), and catalysts reported to promote chemoselective reduction of nitro groups and alkenes (Pt(V)C, Au/TiO_2, $Ir/CaCO_3$). Five operating-condition levels were selected across the experimental campaign: temperature was varied from 25 to 85 °C, pressure from 25 to 85 bar, and flow rate from 3 to 1 mL/min (Figure 7.1). To isolate catalyst-functional group interactions, other variables, including solvent type (EtOH), substrate concentration (0.01 M), and reactor configuration, were kept constant.

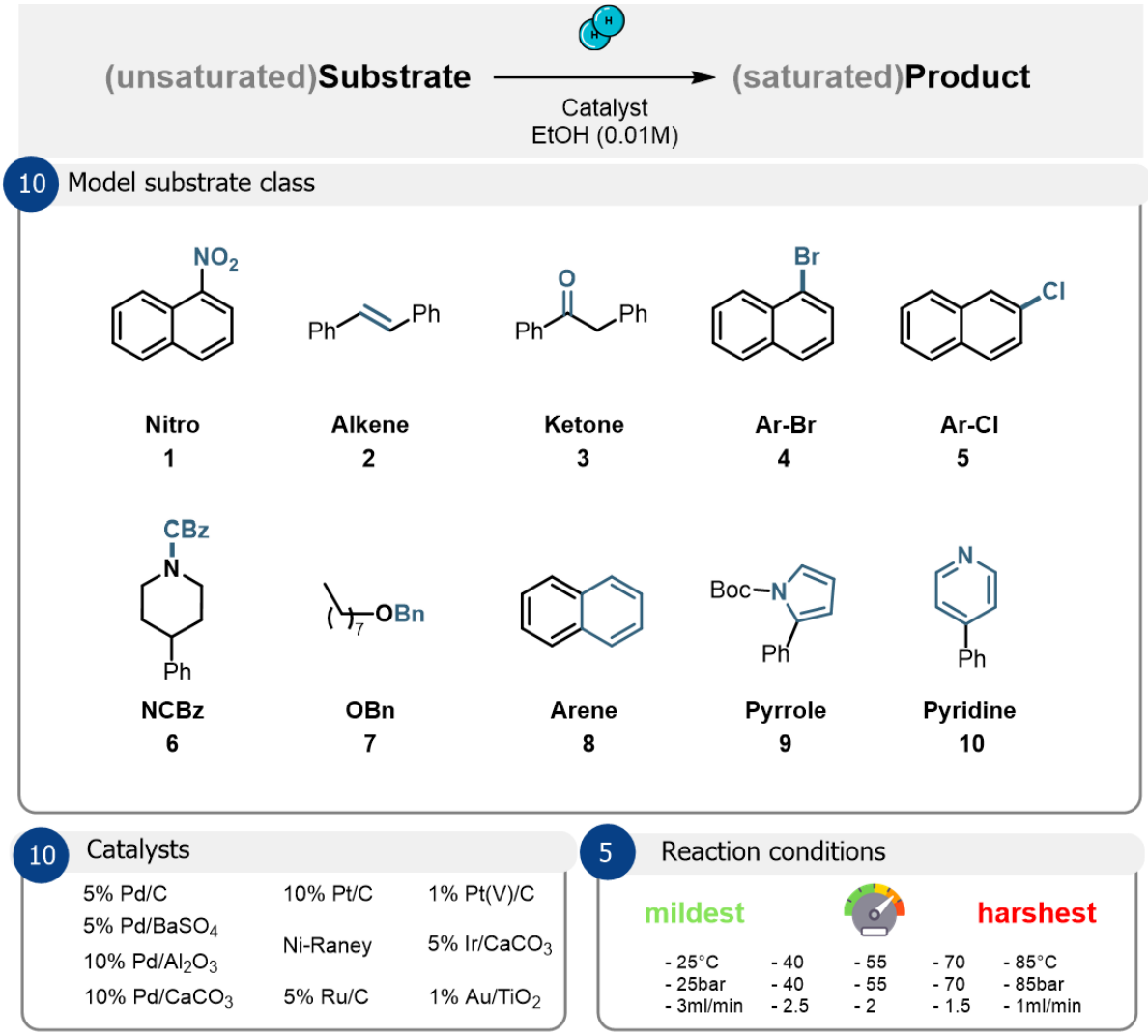


Figure 7.1. Chemical space of hydrogenation reactions covered in this work. Relevant functional group classes were represented by model substrates and reduced under a diverse set of catalytic reaction conditions.

7.3.2 Machine learning models

The model development workflow is summarized in Figure 7.2, which outlines the input variables, preprocessing procedures, the evaluated learning algorithms and the resulting outputs. As outlined in the previous section, several experimental factors, such as solvent type, substrate concentration, and H_2 flow rate, were held constant across all runs. Because these variables exhibited no variation, they cannot explain observed variability in reaction outcomes and therefore provide no statistical significance for supervised learning. Consequently, the ML input space was restricted to variables that were intentionally varied in the dataset and that possess expected mechanistic relevance to

catalytic performance. Accordingly, the final set of predictors comprised substrate type, catalyst type, and the operating conditions (temperature, pressure, and flow rate). Two preprocessing strategies were used for these input variables, as described in detail in Section 7.3.3.

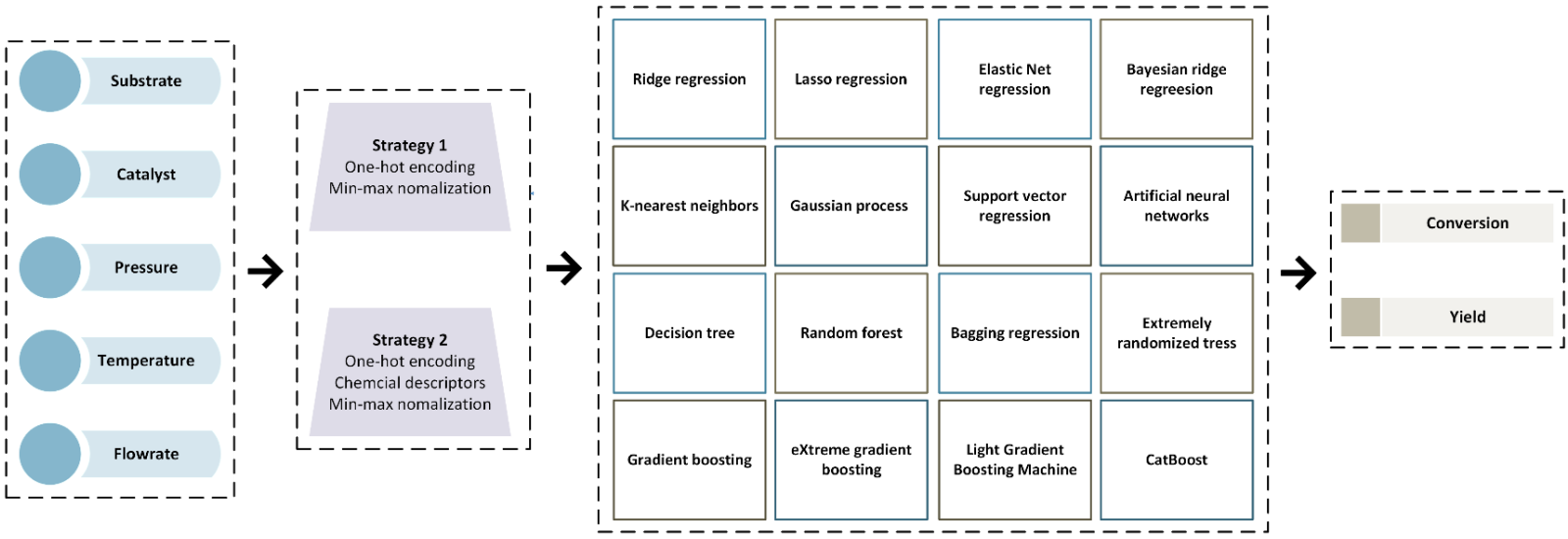


Figure 7.2. Overview of the ML framework used in this study. Selected experimental inputs are provided to a suite of regression models, which are trained and compared to predict key reaction performance metrics.

Using this input set, sixteen ML models were trained and benchmarked to predict two key reaction performance metrics: conversion and yield. Evaluating a diverse collection of algorithms enables coverage of a broad spectrum of functional relationships, from strictly linear dependencies to highly nonlinear, interaction-rich mappings. This diversity is essential because catalytic systems often exhibit threshold behaviour, synergistic interactions between catalyst and substrate identity, and non-additive responses to operating conditions [20,36]. The evaluated models included four linear and regularized-linear models: Ridge regression [37], Lasso regression [38], Elastic Net regression (EN) [39], and Bayesian Ridge regression (BRR) [40]. These approaches provide strong baselines, particularly when the underlying structure is approximately additive or when interpretability of coefficient estimates is desirable. Regularization via L1 and L2 penalties stabilizes parameter estimation in the presence of correlated predictors and mitigates overfitting, while Bayesian Ridge adds a probabilistic prior structure that yields shrinkage and enhances the robustness of inferred coefficients.

To capture nonlinear relationships and higher-order interactions, we also trained eight tree-based models, including both a single-tree baseline and multiple ensemble variants: decision tree (DT) [41], RF [42], bagging regressor [43], extremely randomized trees (ET) [44], gradient boosting (GB) [45], eXtreme gradient boosting (XGBoost) [46], light gradient boosting machine (LightGBM) [47], and CatBoost [48]. DTs construct a hierarchical set of if–then rules by recursively partitioning the feature space into regions with relatively homogeneous target values. This structure naturally captures threshold behaviour (e.g., performance changes above a certain temperature) and interaction effects (e.g., substrate-specific sensitivity to pressure). Ensemble methods further enhance robustness and predictive accuracy by aggregating the predictions of multiple trees. Bagging-based approaches, such as RF and Bagging, reduce variance by averaging predictions from trees trained on bootstrap-resampled data. ETs introduce additional randomness in split selection, which enhances generalization by decorrelating individual trees. Boosting-based approaches (GB, XGBoost, LightGBM, CatBoost) build models sequentially, with each new tree trained to correct the residual errors of the current ensemble, yielding strong performance on complex tabular datasets. In particular, XGBoost and LightGBM implement optimized boosting frameworks that incorporate regularization and computational efficiencies, while CatBoost is specifically designed to handle categorical variables effectively and to mitigate common sources of bias and leakage associated with categorical encoding.

Finally, we included four additional "standard" ML models that represent distinct modelling paradigms: k-nearest neighbours (KNN) [49], Gaussian process regression (GPR) [50], support vector regression (SVR) [51], and artificial neural networks (ANNs) [52]. KNN is an instance-based method that predicts outcomes by averaging the responses of the most similar experiments in feature space, making it effective when local similarity is meaningful. GPR is a kernel-based probabilistic approach that places a prior distribution over functions and generates predictions as posterior means accompanied by uncertainty estimates, which is advantageous for small datasets where quantifying uncertainty is particularly valuable. SVR fits a margin-based regression function that is robust to noise within an ε-insensitive band and can capture nonlinear relationships through kernel transformations. ANNs learn flexible nonlinear mappings through layered transformations of the input variables, allowing approximation of complex functional relationships when sufficient data and appropriate regularization are available.

All sixteen models were trained to learn the mapping from the selected experimental inputs (substrate, catalyst, temperature, pressure, and flow rate) to the two performance targets (conversion and yield). Conversion represents the percentage of the starting substrate consumed during the reaction, whereas yield quantifies the amount of desired product formed relative to the theoretical maximum obtainable from the initial substrate. This systematic benchmarking framework was designed to identify the best-performing model, defined as the model achieving the highest predictive accuracy for both outputs. At the same time, the comparison provides insight into the nature of the underlying input–output relationships: whether they are sufficiently captured by linear effects, require nonlinear models capable of representing interaction terms, or benefit from more flexible function approximators capable of modelling complex, high-dimensional dependencies.

7.3.3 Data preprocessing

In the data preprocessing stage, which is one of the most consequential steps in building reliable ML models [53,54], all continuous variables were min–max normalized to the [0,1] interval. This normalization places predictors on a common scale and promotes numerically stable training across diverse model classes. For the categorical input variables (substrate type and catalyst type; 10 classes each), two complementary feature-representation strategies were implemented to evaluate the trade-off between purely categorical encoding and chemically informed descriptor-based representations.

In the first representation strategy, substrate and catalyst identities were both encoded via one-hot encoding, yielding a sparse binary set that treats each class as distinct and non-ordinal. This approach offers a transparent baseline and avoids introducing external chemical assumptions into the feature space. In the second strategy, catalyst identity was again represented by one-hot encoding, while the substrate was described using molecular descriptors. Because comprehensive descriptors for organic molecules can be computed reproducibly from molecular structure using the RDKit Python toolkit [55], a broad set of substrate descriptors (exceeding 200 features per substrate) was extracted to capture physicochemical and topological information for each substrate. However, using the full descriptor set would substantially increase the dimensionality of the feature space relative to the dataset size, potentially amplifying noise, increasing model complexity, and degrading generalization performance.

To construct a compact descriptor subset that preserves predictive signal while controlling dimensionality, RF feature-importance analysis [56] was conducted separately for conversion and yield (Figure 7.3). Performing this screening on a per-target basis is important, as the molecular features governing overall reactant consumption (conversion) may differ from those influencing product-forming pathways and selectivity losses that ultimately determine yield. For each target, the ten highest-ranking descriptors were retained. As shown in Figure 7.3, four descriptors were consistently identified as highly influential for both outputs, while the remaining top-ranked descriptors differed between conversion and yield. Taking the union of both sets resulted in sixteen unique substrate descriptors for subsequent modelling.

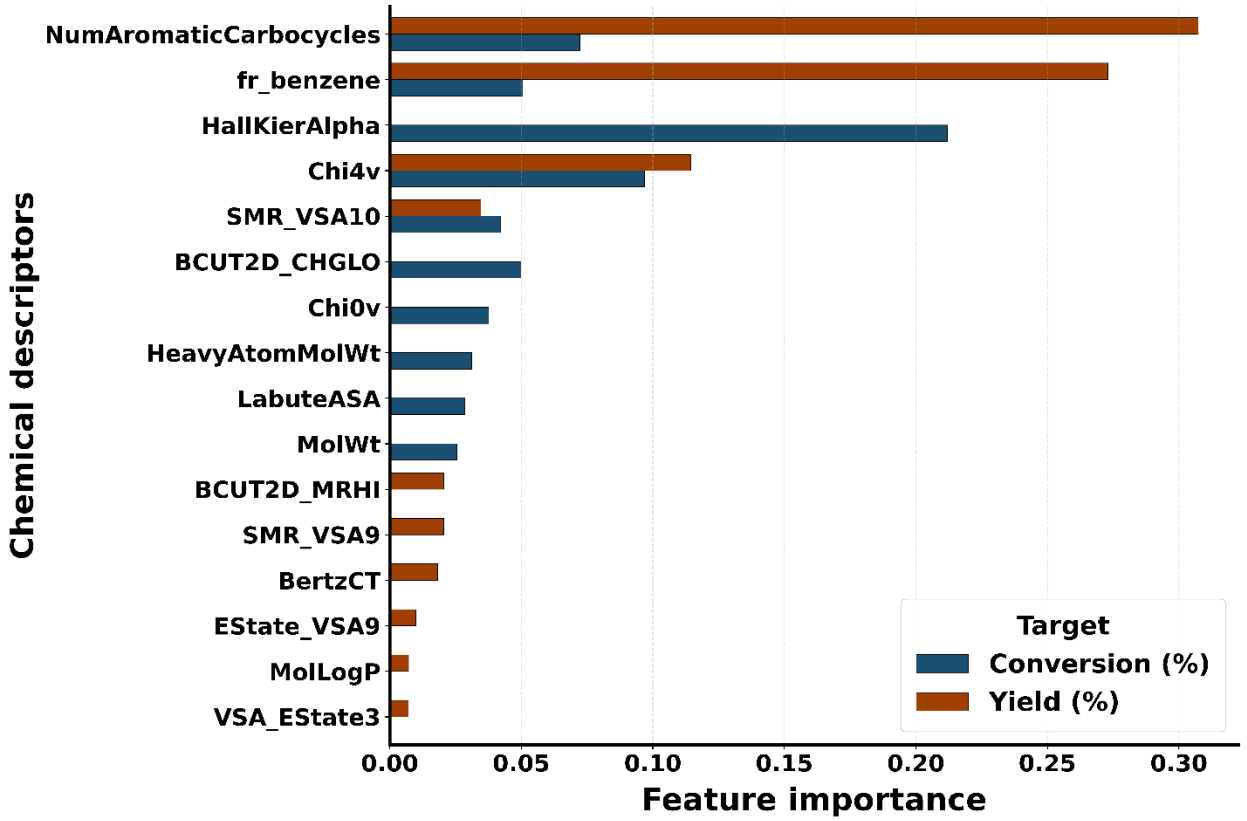


Figure 7.3. RF–based feature-importance ranking of substrate molecular descriptors for predicting conversion and yield. Importance scores were computed separately for each target to account for target-specific descriptor relevance. The top ten descriptors per target were selected; four descriptors were common to both lists, and the remaining descriptors were target-specific, yielding sixteen unique descriptors used as the shared substrate descriptor set for model training.

To ensure methodological consistency and enable direct comparison across outputs, the final modelling stage used a shared feature space: models for both conversion and yield were trained using the same set of sixteen substrate descriptors, together with the encoded catalyst identity and normalized continuous operating variables. This unified representation balances chemical expressiveness with model tractability and supports a fair, aligned evaluation of predictive performance across the two targets.

Table 7.1 summarizes the final set of substrate descriptors retained for model development and groups them into the corresponding RDKit descriptor families. The selected features span topological and fragment-count descriptors (capturing ring content and key substructures), connectivity, shape and structural complexity indices (quantifying branching patterns and higher-order molecular topology), BCUT eigenvalue descriptors (encoding connectivity weighted by atomic properties such as charge and refractivity), and several surface-area–binned descriptors that aggregate molecular van der Waals surface contributions across bins of partial charge, molar refractivity, and electrotopological state. In addition, a small set of physicochemical property descriptors characterizes global molecular attributes such as size, mass, polarizability, and hydrophobicity [55]. It should be noted, however, that descriptor selection was based solely on the RF feature-importance analysis rather than by any assumed or established mechanistic relationship between these descriptors and hydrogenation behaviour.

Table 7.1. Summary of the selected RDKit substrate descriptors used for model development, including a brief definition of each descriptor and its corresponding descriptor category [55].

Descriptor Name	Description	Categories
NumAromaticCarbocycles	Counts the number of aromatic carbon-only ring systems in the molecule.	Topological / fragment count
BCUT2D_MRHI	A BCUT (Burden eigenvalue) descriptor capturing an upper-end value derived from atomic connectivity weighted by molar refractivity-related properties.	2D eigenvalue / topological (BCUT)
PEOE_VSA2	Total van der Waals surface area contribution of atoms falling in a specific partial-charge bin from PEOE charges.	Charge–surface area (VSA)
BCUT2D_CHGLO	A lower-end BCUT (Burden eigenvalue) descriptor derived from connectivity weighted by atomic partial charge.	2D eigenvalue / topological (BCUT)
Chi0v	The valence connectivity index of order 0 summarizing overall branching based on valence states.	Topological connectivity (Chi index)

MolWt	The molecular weight calculated as the sum of atomic masses of the molecule.	Physicochemical (size/mass)
BertzCT	A graph-theoretic complexity measure reflecting structural complexity and branching in the molecular graph.	Topological complexity
SMR_VSA9	Total van der Waals surface area of atoms whose fragment-based molar refractivity (SMR) falls within a defined bin.	Fragment-property–surface area (VSA)
Chi4v	The valence connectivity index of order 4 capturing more distant (higher-order) connectivity patterns in the molecular graph.	Topological connectivity (Chi index)
EState_VSA9	Total van der Waals surface area of atoms whose electrotopological state (EState) values fall within a defined bin.	EState–surface area (VSA)
HeavyAtomMolWt	The molecular weight considering only non-hydrogen (heavy) atoms.	Physicochemical (size/mass)
fr_benzene	Indicates (counts) the presence of benzene ring fragments in the structure.	Fragment count (substructure)
MolMR	The molar refractivity, a proxy for molecular polarizability derived from fragments/structure.	Physicochemical (polarizability)
HallKierAlpha	A shape/size correction factor related to molecular flexibility and deviation from idealized carbon frameworks.	Topological / shape (Hall–Kier)
SMR_VSA10	Total van der Waals surface area of atoms whose SMR values fall within another (higher) predefined bin.	Fragment-property–surface area (VSA)
MolLogP	An estimated octanol/water partition coefficient reflecting overall hydrophobicity (lipophilicity).	Physicochemical (lipophilicity)

7.3.4 Data splitting and hyperparameter tuning

To evaluate model generalization and more effectively capture interactions among the input variables, we did not use a conventional random partitioning of the dataset into training and test subsets. Instead, we adopted a leave-one-experiment-out evaluation strategy, which provides a more rigorous assessment of a model's ability to extrapolate to previously unseen chemical systems [16]. In this approach, an entire experiment corresponding to a randomly selected substrate–catalyst combination was withheld as the test set. Each held-out experiment comprised five measurements collected under distinct operating conditions; thus, the test set consisted of five data points representing one substrate–catalyst pair. The remaining 495 observations were used for model training. This protocol avoids information leakage across closely related data points and provides a more stringent assessment of predictive performance for new substrate–catalyst combinations. For ANN models, an additional 10% of the remaining training data was reserved as

a validation set during training. Model performance was quantified using mean squared error (MSE) and the coefficient of determination (R^2). MSE quantifies the average squared prediction error, with lower values indicating better accuracy. R^2 measures the proportion of variance explained relative to a mean-only baseline, with higher values indicating stronger predictive agreement.

Hyperparameter tuning [57] was performed for each of the sixteen ML models under every preprocessing strategy to ensure a fair and unbiased comparison across algorithms. For each model, a predefined grid of candidate hyperparameter values was evaluated using GridSearchCV with five-fold cross-validation applied to the training set. In this procedure, the training data set was partitioned into five folds; models were iteratively trained on four folds and validated on the remaining fold, cycling through all folds so that each subset served once as the validation set. Cross-validated performance was then aggregated across folds for each hyperparameter combination, and the configuration yielding the highest mean performance was selected as the final setting for that model. The selected hyperparameters were then used to refit the model on the full training dataset, after which performance was evaluated on the held-out test set.

7.3.5 Optimization

After identifying the best-performing ML models for predicting conversion and yield, the final stage of this work applies GA [58] to search the catalyst space and operating-condition domain for combinations that maximize chemoselective performance. In this optimization framework, the operating conditions were allowed to vary within their prescribed decision-space bounds, enabling simultaneous optimization of both catalyst choice and reaction conditions. Rather than optimizing each substrate independently, we define a unified chemoselectivity score that reflects the desired preferential hydrogenation of one functional group in the presence of another. Specifically, the chemoselectivity metric rewards conditions that deliver a high yield for the desired reaction while penalizing conversion associated with an undesired, competing reaction. The resulting objective function therefore maximizes the difference between the predicted yield of the desired reaction and the predicted conversion of the undesired reaction. In doing so, it favours catalyst-condition combinations that selectively hydrogenate the target functional group while suppressing reaction at the competing functional group.

Given the ten available substrates, we evaluated all ordered substrate pairs, resulting in 90 distinct desired–undesired combinations. For each combination, the GA was applied to identify the catalyst and operating conditions that maximize the chemoselectivity score. In addition, to enable catalyst-level performance assessment, the GA was executed separately for each of the ten catalysts on every ordered pair (10 × 90 = 900 optimizations), with catalyst identity held fixed while temperature, pressure, and flow rate were optimized. The chemoselectivity objective is defined in Equation (7.1):

$$CS = Y_{desired} - X_{protected} \tag{7.1}$$

where CS is the chemoselectivity score, Y_{desired} is the predicted yield of the desired product from the target functional group, and $X_{\text{protected}}$ is the predicted conversion of the functional group intended to remain unreacted.

Based on Equation (7.1), the optimal catalyst and corresponding operating conditions were identified and reported for each pair of functional groups exhibiting competing reactivity. To further interpret the GA-derived optima and to elucidate operating-condition regimes associated with high chemoselectivity, we applied k-means clustering [59] to the set of optimal solutions using temperature, pressure, and flow rate as clustering features. Clustering was performed both (i) in the original three-dimensional operating-condition space and (ii) after applying principal component analysis (PCA) [60] to facilitate visualization and assess whether comparable groupings emerge in a reduced coordinate system. The number of clusters k was determined using two complementary criteria: the elbow method, based on within-cluster sum of squares (inertia), to identify diminishing returns in cluster compactness, and the silhouette coefficient, which was maximized to select the value of k that provides the best balance between intra-cluster cohesion and inter-cluster separation.

7.4 Results and discussion

7.4.1 Model development and predictive performance

Under Strategy 1, both substrate and catalyst identities are represented using one-hot encoding, while the operating conditions are min–max scaled. As shown in Figure 7.4, this representation leads to poor generalization performance for conversion prediction under the leave-one-experiment-out protocol. Although most model families achieve strong apparent fit on the training

data, as indicated by high training R^2 values, their predictive accuracy deteriorates substantially on held-out substrate–catalyst pairs. This behaviour is reflected in pronounced train–test gaps, large test-set MSE values, typically on the order of 10^3, and predominantly negative test R^2 values, all of which indicate severe overfitting and limited extrapolation capability when chemical identity is represented only through categorical encodings. The ANN is a notable exception, exhibiting markedly better generalization than the other models. Unlike the other algorithms, the ANN was trained using an internal validation set for model selection and early stopping, which effectively mitigated overfitting. Under Strategy 1, it achieved MSE values of 191 (train), 205 (validation), and 235 (test), with corresponding R^2 values of 0.90 (train), 0.87 (validation), and 0.54 (test). The close agreement between training and validation performance suggests that the validation-based training procedure was effective, although the decline in the held-out experiment confirms that Strategy 1 still imposes substantial limitations on conversion prediction for unseen substrate-catalyst combinations.

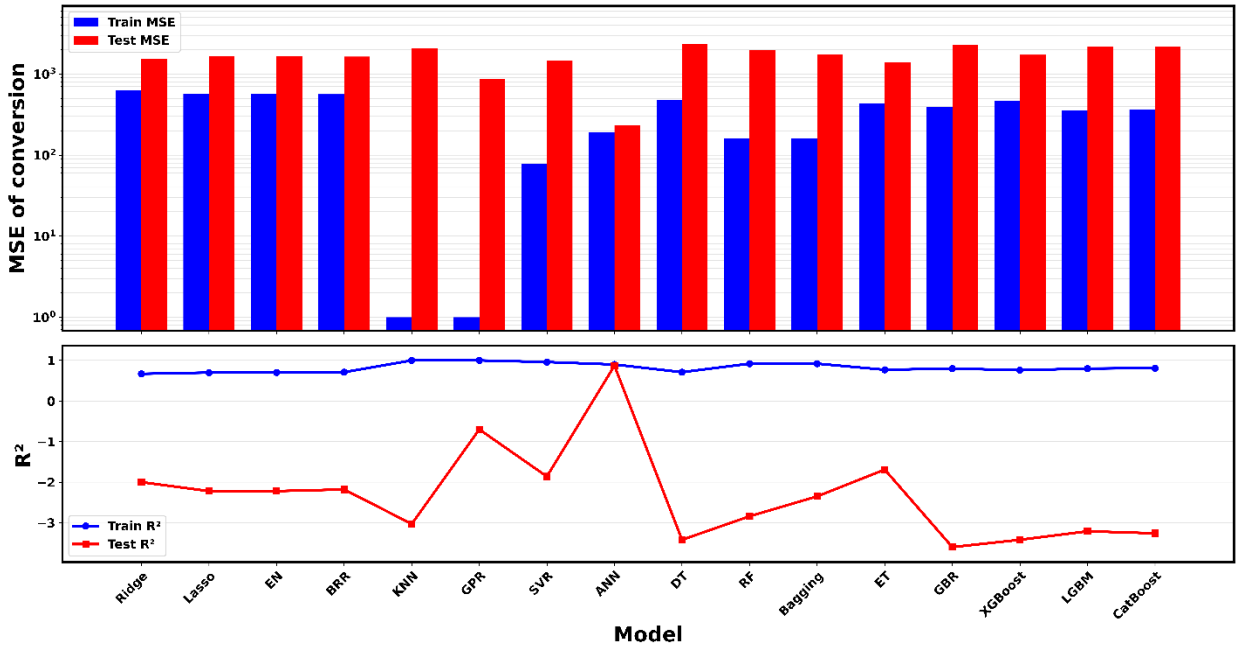


Figure 7.4. Conversion model performance under Strategy 1 (one-hot substrate and catalyst): training vs. test MSE and R^2 across sixteen models using leave-one-experiment-out evaluation.

A similar pattern is observed for yield prediction under Strategy 1, as presented in Figure 7.5. Most models again exhibit pronounced train–test gaps and negative test R^2 values, indicating out-of-sample performance inferior to that of a mean-only baseline. Among the evaluated algorithms, ET

delivers the strongest overall performance in this setting, achieving the best test metrics, with MSE of 394 and R^2 of 0.23, while maintaining good training accuracy, with MSE of 307 and R^2 of 0.80. By contrast, several models that appear highly accurate on the training data, such as KNN and GPR, both of which achieve training R^2 values near 1.0, fail to transfer this apparent fit to held-out experiments. This outcome reinforces that in-sample accuracy is not a reliable indicator of predictive generalization under the leave-one-experiment-out framework, where extrapolation to unseen substrate–catalyst combinations is required. Overall, ET can be regarded as the most robust baseline model for yield prediction when only categorical inputs are used. Nevertheless, the broader pattern suggests that richer structural information is needed to obtain more reliable and transferable yield predictions across previously unobserved chemical systems.

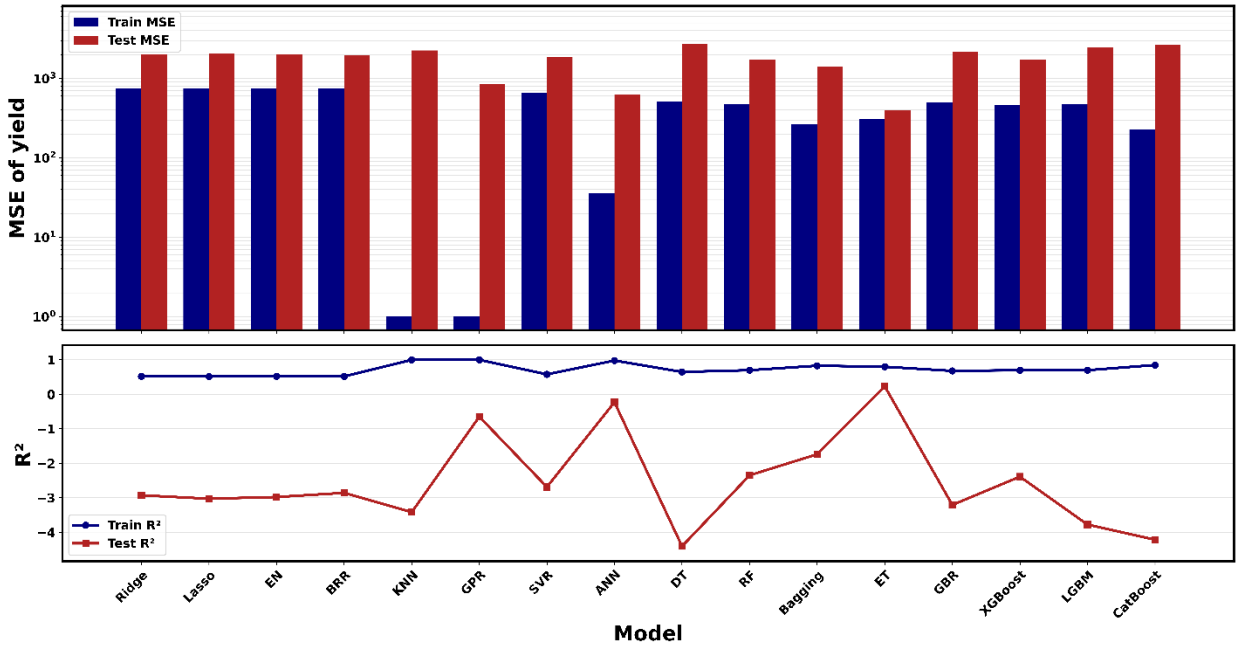


Figure 7.5. Yield model performance under Strategy 1 (one-hot substrate and catalyst): training vs. test MSE and R^2 across sixteen models using leave-one-experiment-out evaluation.

Under Strategy 2, catalyst identities remain one-hot encoded, whereas substrates are represented using the selected RDKit descriptor set, with operating conditions again min–max scaled. Relative to Strategy 1, Figure 7.6 shows a clear improvement in conversion-prediction generalization. Several non-neural models exhibit less severe degradation from training to testing, and their test R^2 values shift closer to zero, with some becoming slightly positive. This indicates that introducing substrate molecular descriptors provides a more informative basis for extrapolating to unseen

substrate–catalyst pairs than categorical identifiers alone. However, accurately representing the molecular attributes that govern hydrogenation behaviour remains challenging. Conversion is strongly influenced by the identity and local chemical environment of the functional group undergoing hydrogenation, and these chemically decisive effects may not be fully captured by the selected descriptor subset. Consequently, substantial train–test gaps persist for many model families, and test performance remains weak for most algorithms, reflecting the intrinsic difficulty of predicting conversion for entirely unseen chemistry combinations. The ANN again achieves the best performance and benefits most strongly from the descriptor-enhanced representation. Using an internal validation set for model selection and early stopping, the ANN attained MSE values of 1 (training), 15 (validation), and 96 (test), with corresponding R^2 values of 0.99 (training), 0.99 (validation), and 0.81 (test). Compared with its performance under Strategy 1, where the test R^2 and test MSE were 0.54 and 235, respectively, Strategy 2 substantially improves held-out predictive performance, increasing test R^2 from 0.54 to 0.81 and reducing test MSE from 235 to 96, an approximately 59% decrease in test error. These results demonstrate that substrate descriptors substantially improve conversion predictability and that the ANN is particularly effective at leveraging this richer molecular representation to generalize across previously unseen substrate–catalyst pairs.

Yield prediction also improves substantially under Strategy 2, as shown in Figure 7.7, although the extent of improvement remains model dependent. Several algorithms display less pronounced deterioration on the test folds, with test R^2 values moving closer to zero relative to Strategy 1. Notably, the identity of the best-performing model shifts from ET under Strategy 1 to the ANN under Strategy 2. The ANN achieves strong held-out performance, with a test MSE of 108 and a test R^2 of 0.78, while maintaining near-perfect training performance, with MSE of 3 and R^2 of 0.99. This behaviour suggests that the ANN is able to leverage the descriptor-based substrate representation to learn more transferable relationships for yield across unseen substrate–catalyst pairs. Because the leave-one-experiment-out protocol withholds an entire substrate–catalyst pair, this improvement is more significant than a gain observed under a conventional random split. It suggests that the descriptor-based representation enables the model to capture relationships among molecular structure, catalyst identity, and operating conditions that can be partially transferred to new substrate–catalyst pairs. Rather than relying only on categorical labels, the model can use chemically meaningful substrate information to relate unseen substrates to previously observed

ones, thereby improving its ability to estimate yield for combinations not present in the training data. Among the remaining models, ETs remain comparatively robust, with a test MSE of 487 and R^2 of 0.05, although they are clearly outperformed by the ANN under Strategy 2. Relative to the best-performing Strategy 1 model for yield, namely ET with a test MSE of 394 and R^2 of 0.23, the ANN under Strategy 2 increases test R^2 from 0.23 to 0.78 and reduces test MSE from 394 to 108, corresponding to an approximately 73% decrease in test error. These results provide two important conclusions. First, the choice of chemical representation strongly controls whether transferable information can be extracted from the dataset. Second, the model family determines how effectively this information is used. Under the descriptor-based strategy, the ANN appears particularly well-suited to capturing nonlinear interactions among substrate descriptors, catalyst identity, and operating conditions, whereas ET provides a comparatively robust but less expressive alternative. By contrast, several models that achieve excellent in-sample fit fail to generalize to held-out substrate–catalyst pairs, indicating that apparent training accuracy is not a reliable criterion for model selection in this application. Overall, these results indicate that Strategy 2 yields a markedly more predictive model for out-of-sample yield estimation.

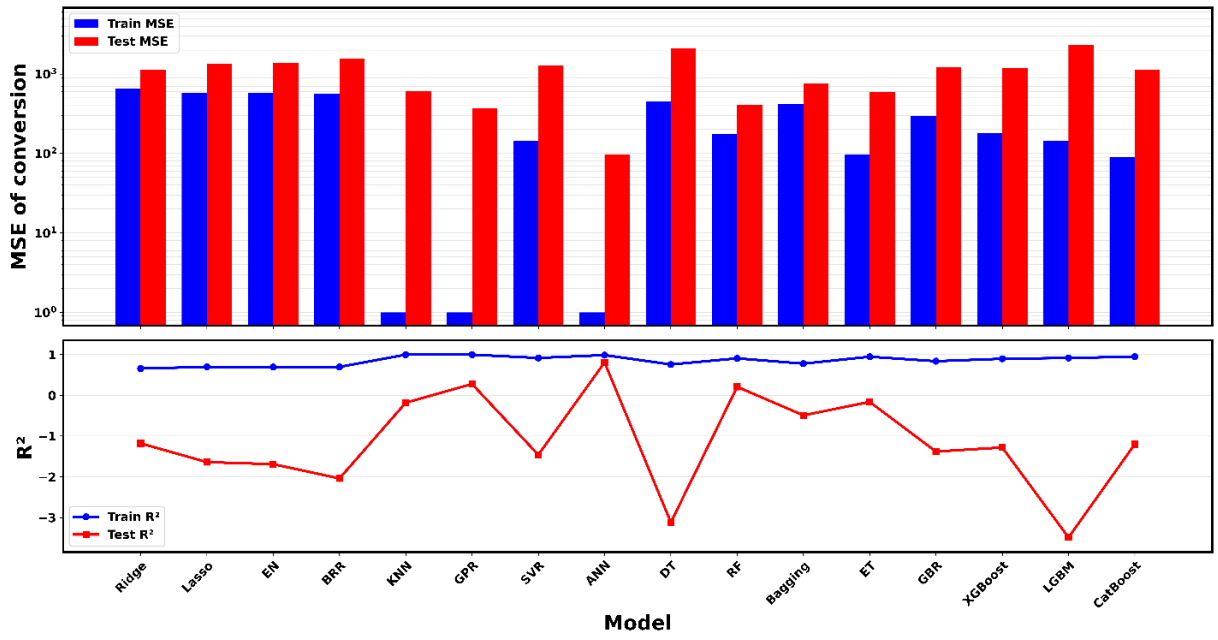


Figure 7.6. Conversion model performance under Strategy 2 (RDKit substrate descriptors + one-hot catalyst): training vs. test MSE and R^2 across sixteen models using leave-one-experiment-out evaluation.

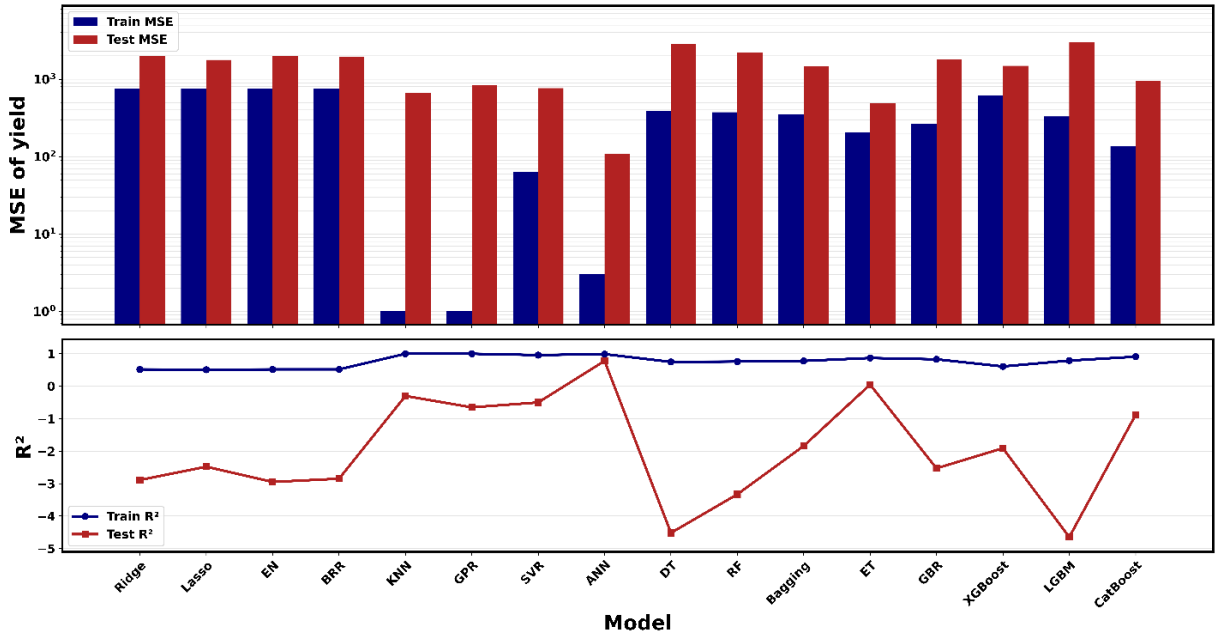


Figure 7.7. Yield model performance under Strategy 2 (RDKit substrate descriptors + one-hot catalyst): training vs. test MSE and R^2 across sixteen models using leave-one-experiment-out evaluation.

Overall, the benchmarking results show that predictive generalization to unseen substrate–catalyst pairs remain challenging and is highly sensitive to the underlying chemical representation. Relative to categorical encoding alone, structure-informed substrate representations provide a substantially stronger foundation for learning transferable relationships, particularly when paired with flexible nonlinear models such as ANNs.

7.4.2 Chemoselectivity optimization and catalyst recommendation

Following model development and evaluation, the ANN models identified as the best-performing predictors under Strategy 2 were selected for deployment. Specifically, the final ANN models for conversion and yield were integrated into the GA-based optimization framework to evaluate catalyst–condition combinations and maximize the chemoselectivity objective defined in Equation (1). Thus, this stage focuses on optimization using the previously trained and selected predictive models, with Figure 7.8 summarizing the resulting deployment-stage optimization. The heatmap is organized as an ordered-pair selectivity landscape: the y-axis lists the substrate designated as the desired hydrogenation target (Substrate A), and the x-axis lists the substrate treated as the competing or undesired reactant (Substrate B). Each cell, therefore, corresponds to a distinct

optimization problem of the form “maximize the yield of desired product from Substrate A while minimizing the conversion of Substrate B”, and the colour scale reports the highest chemoselectivity score achievable for that specific ordered pair.

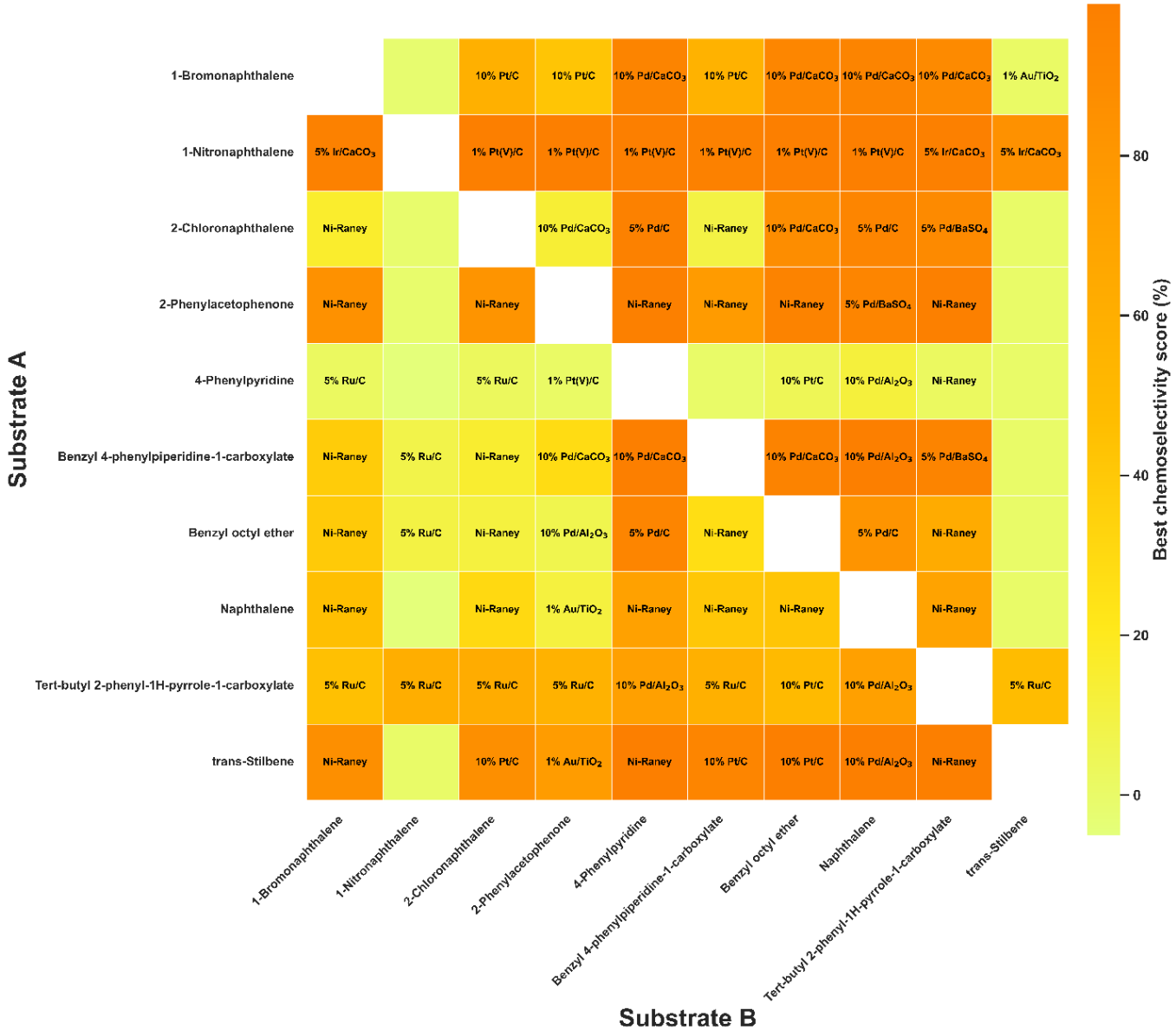


Figure 7.8. GA-optimised chemoselectivity landscape for ordered substrate pairs using the selected best ML predictors from Strategy 2. Each cell represents a substrate pair in which Substrate A is targeted for hydrogenation and Substrate B is intended to remain unreacted. Cell colour indicates the maximum chemoselectivity score identified by the GA, while the cell label indicates the corresponding optimal catalyst.

The catalyst label printed within each cell identifies the catalyst that attained the maximum chemoselectivity score. Empty cells denote substrate competitions with chemoselectivity scores ≤ 0%, for which the GA–ML optimization failed to deliver a positive yield–conversion gap. In such cases, the heatmap intentionally omits a catalyst label, indicating that no catalyst recommendation is made for that ordered pair. Therefore, although 90 ordered substrate competitions were evaluated, the heatmap reports optimal catalyst recommendations for only 77 pairs (13 pairs with chemoselectivity ≤ 0% are left unlabeled).

Since the chemoselectivity score is directional, as defined in Equation (7.1), the matrix must be interpreted strictly from left to right: swapping A and B changes the optimization objective and can lead to a different optimum. Consequently, the map is not intended to be read from bottom to top. Diagonal cells (A = B) have no meaningful interpretation under this formulation and are therefore omitted.

From an interpretation perspective, the orange cells indicate substrate competition pairs for which the model predicts chemoselectivity magnitude that is both substantial and practically meaningful within the explored catalyst and operating-condition space. Specifically, these correspond to pairs in which at least one catalyst-condition region is identified where the desired product from a substrate is predicted to achieve relatively high yield while the competing substrate exhibits comparatively lower conversion. By contrast, the yellow cells correspond to substrate pairs for which the model predicts only limited separation between the desired and competing responses across the tested hydrogenation conditions. For these pairs, improving the yield of the desired reaction is more frequently accompanied by a concomitant increase in conversion of the competing substrate, thereby limiting the achievable chemoselective differentiation within the studied design space. Accordingly, these results suggest that catalyst selection is pair-specific rather than universal. Effective catalyst choices depend on the extent to which the desired pathway can be selectively promoted relative to the competing one.

To extract the maximum interpretive value from Figure 7.8, it is instructive to consider the data not solely as a pairwise chemoselectivity map, but also as a catalyst-selection summary. Across all ordered substrate pairs, the GA most frequently identified Ni-Raney as the optimal catalyst (24/77), followed by 5% Ru/C (10/77), 10% Pd/$CaCO_3$ (9/77), and 10% Pt/C (8/77). This distribution is also chemically meaningful when interpreted in relation to catalyst activity. Highly

active hydrogenation catalysts, such as Pd-based systems, are not universally selected, as their strong activity may also promote conversion of the substrate or functional group intended to remain unreacted, thereby reducing the chemoselectivity score. Conversely, catalysts with very low intrinsic activity are likewise disfavored; although they may suppress undesired conversion, they often fail to provide sufficient yield of the desired transformation. The catalysts most frequently identified by the GA, therefore, reflect a balance between activity and discrimination. They are active enough to promote the target hydrogenation, but selective enough to limit the competing reaction. This behaviour suggests that the optimization framework is not merely ranking catalysts by overall hydrogenation activity. Instead, it is identifying catalyst-condition combinations that best satisfy the specific selectivity trade-off defined for each ordered substrate pair. Interpreted row-wise, the heatmap serves as a catalyst recommendation map for each target (Substrate A) under competitive hydrogenation constraints. Some substrates exhibit strong associations with a small subset of catalysts, whereas others show a more diffuse distribution across competitors. For instance, 1-bromonaphthalene is more often linked to 10% Pd/$CaCO_3$ and 10% Pt/C, while 1-nitronaphthalene is more frequently associated with 1% Pt(V)/C, with 5% Ir/$CaCO_3$ also appearing repeatedly. Likewise, Ni-Raney recurs often for 2-phenylacetophenone and naphthalene, and 5% Ru/C appears frequently for tert-butyl 2-phenyl-1H-pyrrole-1-carboxylate. In contrast, substrates such as 2-chloronaphthalene display a more heterogeneous pattern, with no single catalyst emerging as consistently dominant.

While Figure 7.8 identifies the preferred catalyst for each ordered substrate competition, a complementary catalyst-level view of activity is required to establish whether favourable outcomes arise from high conversion of the desired substrate, suppression of the competing substrate, or a combination of both. Figure 7.9 therefore reports the distributions of GA-optimized Conversion of A and Conversion of B for each of the ten catalysts across all 90 directional substrate pairs. For every pair (Substrate A, Substrate B), temperature, pressure, and flow rate were optimized independently for each catalyst, and the recorded values correspond to the predicted conversions at those catalyst-specific optima. Each panel thus summarizes n = 90 independent optimization outcomes, providing an unbiased comparison of catalytic activity toward the desired and competing substrates across the full competitive design space. The overlays show that catalyst performance cannot be described by overall activity alone, but rather by how strongly Conversion of A and Conversion of B diverge under optimized conditions. Several Pd-based catalysts,

including 5% Pd/C, 10% Pd/$CaCO_3$, and 10% Pd/Al_2O_3, more frequently accumulate Conversion of A near high values while Conversion of B remains comparatively lower or more dispersed, consistent with preferential conversion of the desired substrate. In contrast, catalysts such as 1% Au/TiO_2 and 5% Ir/$CaCO_3$ concentrate both conversions near low values, indicating limited activity across much of the competitive matrix even after condition optimization. Materials including Ni-Raney and 10% Pt/C exhibit broader or bimodal profiles for both conversions, reflecting pronounced pair dependence. Viewed together with Figure 7.8, these distributions separate selection frequency from the underlying A–B activity balance: frequent appearance as an optimal catalyst does not imply uniformly high Conversion of A and low Conversion of B for every competition, but rather favourable discrimination between A and B in specific competitive contexts.

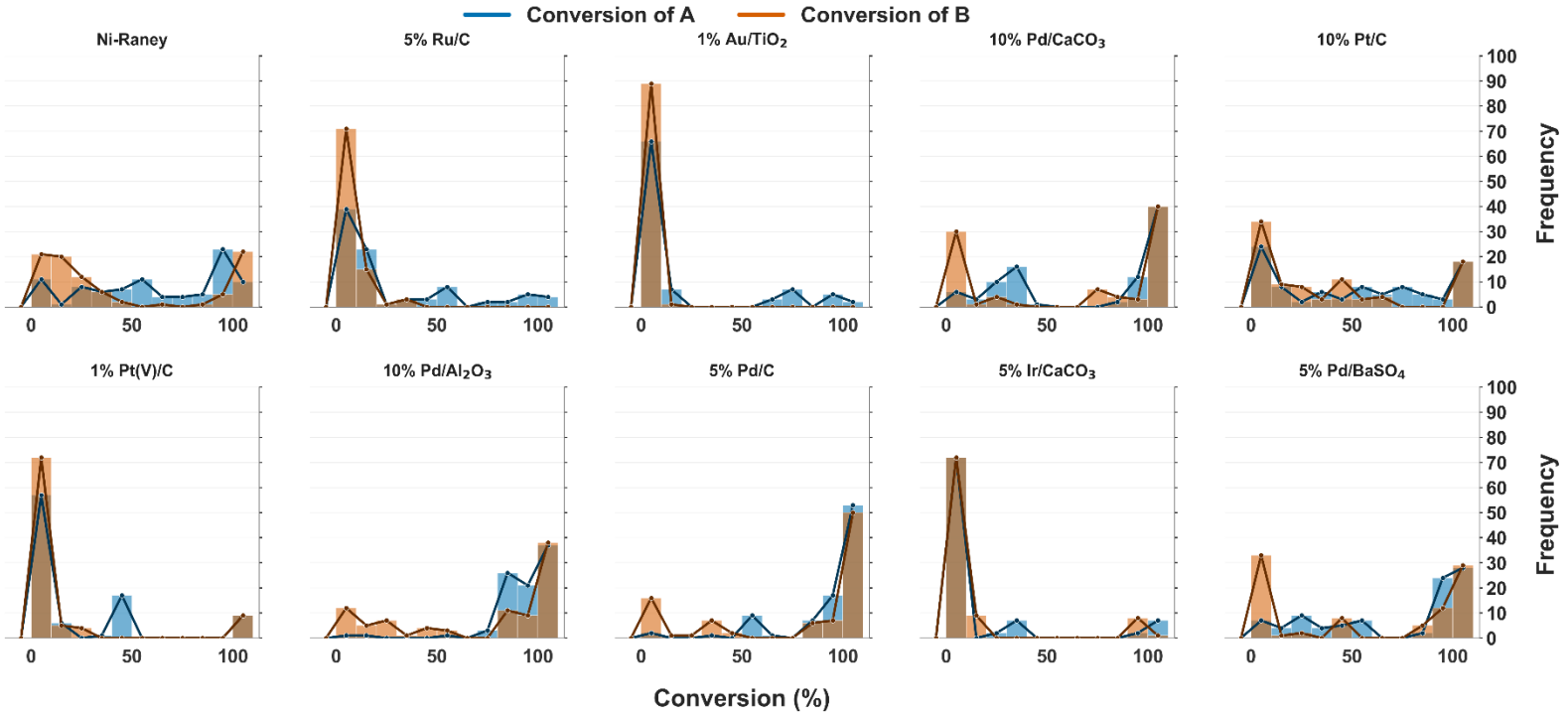


Figure 7.9. Distributions of genetically algorithm–optimized Conversion of A (blue) and Conversion of B (orange) for each catalyst across all 90 ordered substrate pairs (Substrate A versus Substrate B; n = 90 per catalyst). For every catalyst–pair combination, temperature, pressure, and flow rate were optimized independently, and the histograms summarize the resulting conversion values (10% bins). The dashed line indicates 50% conversion. Catalyst panels are arranged according to their frequency of selection as the optimal catalyst.

7.4.3 Interpretable analysis of optimized operating-condition regimes

Figure 7.10 depicts the clustered GA-derived optima projected onto the first two principal components (PC1 = 38.8% variance; PC2 = 29.4% variance), with points colored by cluster membership and scaled according to chemoselectivity score. The projection reveals that the solutions occupy well-defined regions rather than forming a diffuse continuum cloud, indicating that the optimization procedure repeatedly converges toward a limited set of recurring operating-conditions regimes. High-performing solutions are concentrated within a small subset of clusters, most notably Cluster 2 (n = 24; mean = 92.5%; range = 77–99%) and Cluster 9 (n = 5; mean = 96.0%; range = 85–99%), while low-performing solutions occupy separate regions (e.g., Cluster 6: n = 11; mean = 2.9%; range = -5–28%; Cluster 10: n = 3; mean = 2.0%; range = 1–3%). Intermediate clusters exhibit broader score dispersion, reflecting substrate-pair-dependent variability in achievable chemoselectivity.

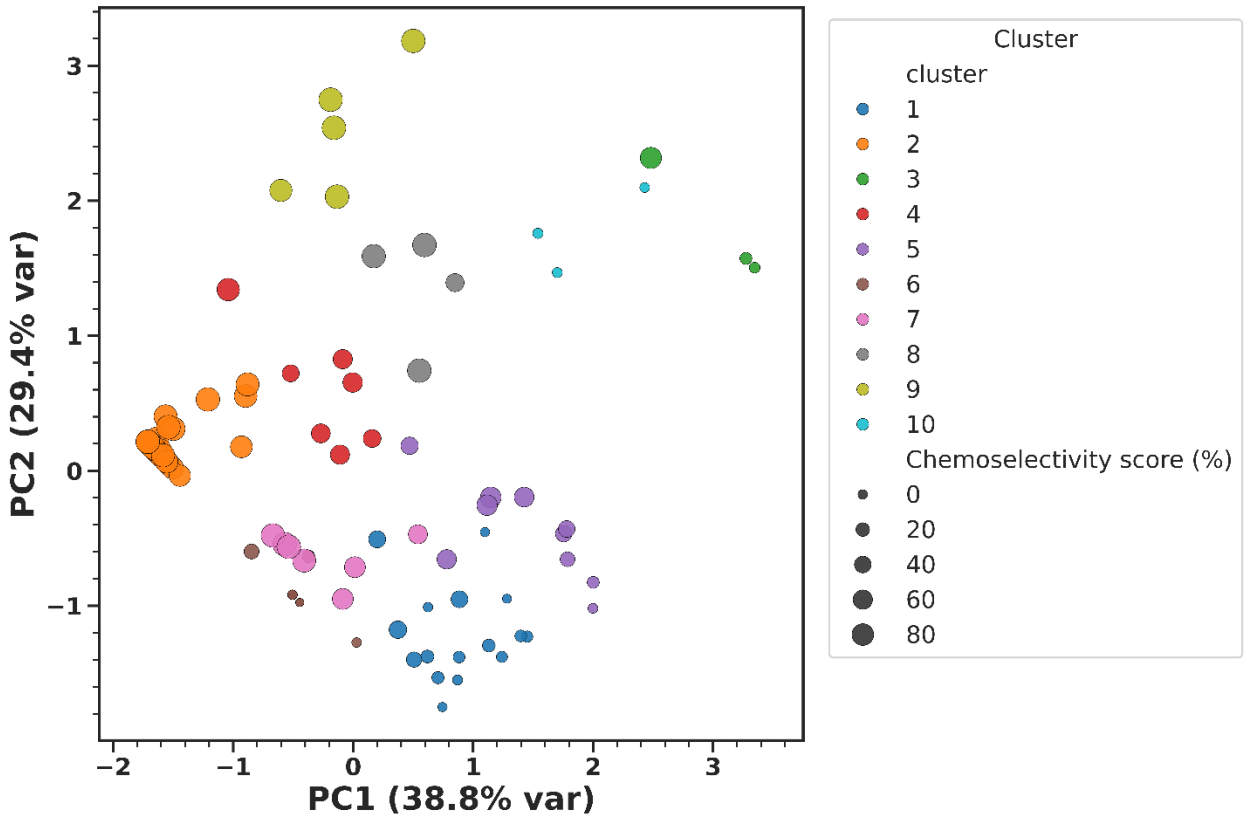


Figure 7.10. PCA visualization of k-means clustering (k = 10) for GA-optimal operating conditions (temperature, pressure, flow rate), with point size proportional to the chemoselectivity score.

Additionally, Figure 7.11 provides the operational interpretation of the clustering results by linking chemoselectivity directly to each decision variable and, crucially, to their combined effects. The dominant high-chemoselectivity regime (Cluster 2; n = 24) corresponds to a highly consistent joint configuration of operating conditions: solutions concentrate at the lowest temperature and pressure values explored (approximately 25 °C and 25 bar) while simultaneously favouring the upper end of the flow rate range (typically near 3 mL/min). This recurring low–low–high pattern (low temperature, low pressure, high flow rate) suggests that many of the best chemoselectivity solutions emerge when the GA drives the system toward milder hydrogenation severity (via reduced temperature and pressure). This combination appears to sustain the desired transformation while limiting the conversion of the undesired substrate. In contrast, the second high-performing regime (Cluster 9; n = 5) exhibits a distinct combination of operating conditions. Although it also favours relatively high flow rates (>2 mL/min), it departs from the strict low-temperature/low-pressure corner, indicating that some substrate-competition scenarios require additional driving force in at least one of these variables to maintain high yield of the desired product without sacrificing selectivity. This regime is narrower and more substrate-pair-specific, yet it achieves chemoselectivity values approaching the upper performance limit.

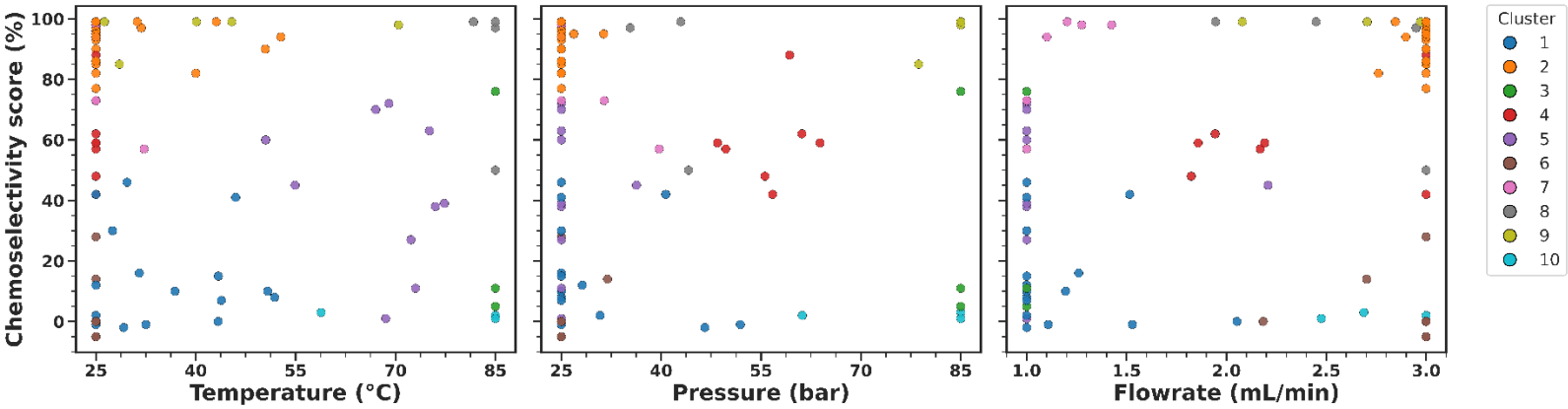


Figure 7.11. k-means clustering results (k = 10) in the original operating-variable space, showing chemoselectivity score versus temperature, pressure, and flow rate for the GA-optimal solutions, coloured by cluster.

Clusters exhibiting moderate-to-high mean scores but larger dispersion (e.g., Cluster 7: n = 7; mean = 84.6%; Cluster 8: n = 4; mean = 86.3%) reflect mixed joint-condition settings, in which high chemoselectivity can be achieved under multiple temperature–pressure combinations, but

only for specific substrate pairs. In these clusters, the GA-identified optimal solutions do not converge toward a single corner of the operating space, implying that selectivity is governed more strongly by substrate identity and how the competing reaction pathways respond to incremental changes in operating severity. By comparison, low-score clusters exhibit operating-condition combinations that repeatedly fail to create separation between desired and undesired pathways. For example, Cluster 6 (n = 11; mean = 2.9%) and Cluster 10 (n = 3; mean = 2.0%) largely occupy regions where one or more variables deviate from the favourable low–low–high pattern, producing conditions under which both substrates respond similarly. As a result, these regimes offer limited capacity to widen the yield–conversion gap defined by the chemoselectivity metric in (Equation (7.1)).

Taken together, Figure 7.10 and Figure 7.11 show that the GA-derived optimal solutions are far from arbitrary; rather, they cluster within a small number of recurring temperature–pressure–flow rate regimes. The most frequently observed high-performance outcomes concentrate in a low-temperature/low-pressure/high-flow rate region (Cluster 2), whereas a smaller subset achieves nearly-maximal chemoselectivity within a distinct regime characterized by high flow rates but shifted temperature and pressure requirements (Cluster 9). This structure provides a compact and operationally meaningful interpretation of the heatmap results by indicating where high-chemoselectivity solutions tend to occur in the operating-condition space and how the optimal balance of temperature, pressure, and flow rate changes across the most influential clusters. Importantly, the combined heatmap and clustering analyses also suggest that chemoselectivity is not governed by a single dominant factor across all cases. For some ordered substrate pairs, the achievable chemoselectivity appears to be primarily substrate–catalyst dependent, with particular catalysts repeatedly favoured regardless of moderate changes in operating conditions. For other pairs, the separation between desired and undesired hydrogenation pathways is more strongly condition-dependent, requiring specific combinations of temperature, pressure, and flow rate to achieve high chemoselectivity. The proposed framework, therefore, provides a way not only to recommend catalysts and operating conditions but also to assess whether a given selective hydrogenation problem is likely to be governed predominantly by catalyst identity, operating-condition control, or their interaction.

To further demonstrate the practical applicability of the developed predictive and optimization framework, the best-performing ML models were also implemented in an interactive web-based application. This application was developed to provide a user-friendly interface for catalyst evaluation, selectivity-based comparison, and operating-condition analysis without requiring users to directly interact with the underlying modelling code. The app includes two main workflows: one for evaluating catalyst performance under predefined operating conditions and another for coupling the trained ML models with a genetic algorithm to identify favourable catalyst–condition combinations. Since the web application serves as a deployment-oriented extension of the modelling framework rather than a core methodological component of the chapter, its detailed structure and workflow are presented separately in Appendix C.

7.5 Conclusion

This work developed a practical workflow that connects predictive modelling of hydrogenation outcomes to chemoselectivity-focused decision support for catalyst and operating-condition selection. Sixteen ML models were trained to predict conversion and yield using a leave-one-experiment-out protocol in which an entire substrate–catalyst pair withheld, enabling the evaluation of model generalization to previously unseen chemistry combinations. The results show that chemical representation supports the model transferability. With categorical-only inputs (Strategy 1; one-hot substrate and catalyst identifiers), most models generalized poorly, often yielding negative test R^2 values. Incorporating RDKit-derived substrate descriptors (Strategy 2; reduced to a sixteen-descriptor set) substantially improved predictive performance: the best conversion model improved from test $R^2 = 0.54$ (Strategy 1) to $R^2 = 0.81$ (Strategy 2), while the best yield model improved from $R^2 = 0.23$ (Strategy 1; ET) to $R^2 = 0.78$ (Strategy 2; ANN). Although the descriptor set does not fully capture all molecular factors governing hydrogenation behaviour, these gains nevertheless demonstrate the promise of structure-informed representations for improving predictive transferability to unseen substrate–catalyst combinations.

The selected best-performing predictors were then deployed within a GA framework to maximize the chemoselectivity objective across 90 ordered substrate competitions, producing a directional chemoselectivity landscape that reports, for each (A - B) pair, the maximum achievable score and the corresponding best catalyst. Across all pairs, the most frequently selected catalysts were Ni-Raney (24/77), 5% Ru/C (10/77), 10% Pd/$CaCO_3$ (9/77), and 10% Pt/C (8/77), with clear

substrate-specific shortlists emerging on a row-wise basis. Catalyst-level score distributions further show that frequent heatmap selection does not imply uniformly high or narrow chemoselectivity across all ordered pairs.

Clustering of GA-optimal operating conditions showed that high-selectivity solutions concentrate into a small number of recurring regimes rather than being scattered across the design space. The dominant high-performance cluster (Cluster 2; n = 24) exhibited a mean chemoselectivity score of approximately 92.5% (range 77–99%) and was characterized by a consistent combination of low temperature and pressure with high flow rate. A smaller cluster (Cluster 9; n = 5) achieved near-ceiling performance, with a mean chemoselectivity of approximately 96.0%, under a narrower and distinct set of operating conditions.

Overall, the study demonstrates that both model choice and chemical representation strongly influence predictive performance for conversion and yield. Although generalization to entirely unseen substrate–catalyst combinations remains challenging, the benchmarking results show that certain model classes, especially when combined with structure-informed substrate representations, capture more transferable relationships than others. At the same time, the value of the modelling framework is not limited to extrapolative performance alone: even within the studied chemistry space, the trained models help organize the catalyst landscape by identifying and sorting promising catalyst-condition combinations under competitive hydrogenation constraints. The results also indicate that the framework can learn useful catalyst-activity trends from the experimental data. Catalysts that are generally too reactive are not always favoured because they may also promote undesired conversion, whereas weakly active catalysts are less frequently selected because they may not provide sufficient yield for the desired transformation. Thus, the model-supported optimization does not simply identify the most active catalysts, but helps distinguish catalysts that offer an appropriate balance between activity and chemoselectivity for each substrate pair. Embedding these predictors within a chemoselectivity-focused optimization framework yields two practical outputs: (i) a directional catalyst-recommendation map across competing substrate pairs and (ii) interpretable operating-condition regimes that indicate where high-chemoselectivity solutions are more likely to arise. Together, these outputs can directly support experimental planning by narrowing catalyst screening efforts and focusing validation on operating windows that are repeatedly associated with high chemoselectivity.

7.6 References

1. Zhang, L., Zhou, M., Wang, A., & Zhang, T. (2020). Selective Hydrogenation over Supported Metal Catalysts: From Nanoparticles to Single Atoms. Chemical Reviews, 120(2), 683–733. https://doi.org/10.1021/acs.chemrev.9b00230
2. Lyons, T. W., Leibler, I. N.-M., He, C. Q., Gadamsetty, S., Estrada, G. J., & Doyle, A. G. (2024). Broad Survey of Selectivity in the Heterogeneous Hydrogenation of Heterocycles. The Journal of Organic Chemistry, 89(3), 1438–1445. https://doi.org/10.1021/acs.joc.3c02028
3. Younas, M., Shafique, S., Hafeez, A., Javed, F., & Rehman, F. (2022). An Overview of Hydrogen Production: Current Status, Potential, and Challenges. Fuel, 316, 123317. https://doi.org/10.1016/j.fuel.2022.123317
4. Zhang, F., Sasmal, H. S., Rana, D., & Glorius, F. (2024). Switchable and Chemoselective Arene Hydrogenation for Efficient Late Stage Applications. Journal of the American Chemical Society, 146(27), 18682–18688. https://doi.org/10.1021/jacs.4c05883
5. Fang, R., Zhang, Q., Yao, C., Wu, H., Xie, S., Zhang, X., Wang, Q., Lyu, J., Feng, F., Lu, C., Zhang, Q., & Li, X. (2024). Prediction of catalytic performance of metal oxide catalysts for alkyne hydrogenation reaction based on machine learning. Applied Catalysis A: General, 687, 119969. https://doi.org/10.1016/j.apcata.2024.119969
6. Loos, P., Alex, H., Hassfeld, J., Lovis, K., Platzek, J., Steinfeldt, N., & Hübner, S. (2016). Selective Hydrogenation of Halogenated Nitroaromatics to Haloanilines in Batch and Flow. Organic Process Research & Development, 20(2), 452–464. https://doi.org/10.1021/acs.oprd.5b00170
7. Corma, A., Serna, P., Concepción, P., & Calvino, J. J. (2008). Transforming Nonselective into Chemoselective Metal Catalysts for the Hydrogenation of Substituted Nitroaromatics. Journal of the American Chemical Society, 130(27), 8748–8753. https://doi.org/10.1021/ja800959g
8. Wang, D.-S., Chen, Q.-A., Lu, S.-M., & Zhou, Y.-G. (2012). Asymmetric Hydrogenation of Heteroarenes and Arenes. Chemical Reviews, 112(4), 2557–2590. https://doi.org/10.1021/cr200328h
9. Liu, D., Jia, Z., Shen, L., Liu, W., Pang, R., Yu, S., Liu, S., Li, L., Liu, Y., & Yu, L. (2024). Machine Learning Assisted Selection of Catalyst for γ-Valerolactone Hydrogenation from

Levulinic Acid. ACS Sustainable Chemistry & Engineering, 12(44), 16340–16353. https://doi.org/10.1021/acssuschemeng.4c05941

10. Dotson, J. J., van Dijk, L., Timmerman, J. C., Grosslight, S., Walroth, R. C., Gosselin, F., Püntener, K., Mack, K. A., & Sigman, M. S. (2023). Data-Driven Multi-Objective Optimization Tactics for Catalytic Asymmetric Reactions Using Bisphosphine Ligands. Journal of the American Chemical Society, 145(1), 110–121. https://doi.org/10.1021/jacs.2c08513
11. Hisata, Y., Washio, T., Takizawa, S., Ogoshi, S., & Hoshimoto, Y. (2024). In-silico-assisted derivatization of triarylboranes for the catalytic reductive functionalization of aniline-derived amino acids and peptides with H_2. Nature Communications, 15(1), 3708. https://doi.org/10.1038/s41467-024-47984-0
12. Ivanytsya, M. O., Subotin, V. V., Gavrilenko, K. S., Ryabukhin, S. V., Volochnyuk, D. M., & Kolotilov, S. V. (2024). Advances and challenges in development of transition metal catalysts for heterogeneous hydrogenation of organic compounds. The Chemical Record, 24(2), e202300300. https://doi.org/10.1002/tcr.202300300
13. Marianov, A. N., Jiang, Y., Baiker, A., & Huang, J. (2023). Homogeneous and heterogeneous strategies of enantioselective hydrogenation: Critical evaluation and future prospects. Chem Catalysis, 3(7). https://doi.org/10.1016/j.checat.2023.100631
14. Singh, S., Pareek, M., Changotra, A., Banerjee, S., Bhaskararao, B., Balamurugan, P., & Sunoj, R. B. (2020). A unified machine-learning protocol for asymmetric catalysis as a proof of concept demonstration using asymmetric hydrogenation. Proceedings of the National Academy of Sciences, 117(3), 1339–1345. https://doi.org/10.1073/pnas.1916392117
15. Vilé, G., Albani, D., Almora-Barrios, N., López, N., & Pérez-Ramírez, J. (2016). Advances in the Design of Nanostructured Catalysts for Selective Hydrogenation. ChemCatChem, 8(1), 21–33. https://doi.org/10.1002/cctc.201501269
16. Jablonka, K. M., Ongari, D., Moosavi, S. M., & Smit, B. (2020). Big-Data Science in Porous Materials: Materials Genomics and Machine Learning. Chemical Reviews, 120(16), 8066–8129. https://doi.org/10.1021/acs.chemrev.0c00004
17. Liles, J. P., Rouget-Virbel, C., Wahlman, J. L. H., Rahimoff, R., Crawford, J. M., Medlin, A., O'Connor, V. S., Li, J., Roytman, V. A., Toste, F. D., & Sigman, M. S. (2023). Data science

enables the development of a new class of chiral phosphoric acid catalysts. Chem, 9(6), 1518–1537. https://doi.org/10.1016/j.chempr.2023.02.020

18. Smith, S. C., Horbaczewskyj, C. S., Tanner, T. F. N., Walder, J. J., & Fairlamb, I. J. S. (2024). Automated approaches, reaction parameterisation, and data science in organometallic chemistry and catalysis: Towards improving synthetic chemistry and accelerating mechanistic understanding. Digital Discovery, 3(8), 1467–1495. https://doi.org/10.1039/D3DD00249G
19. Mishra, A. S., Lade, V. G., Barmar, J. R., Bindwal, A. B., & Birmod, R. P. (2025). A mini review on revolutionizing hydrogenation catalysis: Unleashing transformative power of artificial intelligence. Journal of Molecular Modeling, 31(5), 152. https://doi.org/10.1007/s00894-025-06376-x
20. Guan, Y., Chaffart, D., Liu, G., Tan, Z., Zhang, D., Wang, Y., Li, J., & Ricardez-Sandoval, L. (2022). Machine learning in solid heterogeneous catalysis: Recent developments, challenges and perspectives. Chemical Engineering Science, 248, 117224. https://doi.org/10.1016/j.ces.2021.117224
21. Singh, S., & Sunoj, R. B. (2023). Molecular Machine Learning for Chemical Catalysis: Prospects and Challenges. Accounts of Chemical Research, 56(3), 402–412. https://doi.org/10.1021/acs.accounts.2c00801
22. Xu, L.-C., Zhang, S.-Q., Li, X., Tang, M.-J., Xie, P.-P., & Hong, X. (2021). Towards Data-Driven Design of Asymmetric Hydrogenation of Olefins: Database and Hierarchical Learning. Angewandte Chemie International Edition, 60(42), 22804–22811. https://doi.org/10.1002/anie.202106880
23. Mishra, C., von Wolff, N., Tripathi, A., Brodie, C. N., Lawrence, N. D., Ravuri, A., Brémond, É., Preiss, A., & Kumar, A. (2023). Predicting ruthenium catalysed hydrogenation of esters using machine learning. Digital Discovery, 2(3), 819–827. https://doi.org/10.1039/D3DD00029J
24. Nakajima, H., Murata, C., Noto, N., & Saito, S. (2025). Database Construction for the Virtual Screening of the Ruthenium-Catalyzed Hydrogenation of Ketones. The Journal of Organic Chemistry, 90(2), 1054–1060. https://doi.org/10.1021/acs.joc.4c02347
25. Bozal-Ginesta, C., Pablo-García, S., Choi, C., Tarancón, A., & Aspuru-Guzik, A. (2025). Developing machine learning for heterogeneous catalysis with experimental and

computational data. Nature Reviews Chemistry, 9(9), 601–616. https://doi.org/10.1038/s41570-025-00740-4

26. Irfan, M., Glasnov, T. N., & Kappe, C. O. (2011). Heterogeneous Catalytic Hydrogenation Reactions in Continuous-Flow Reactors. ChemSusChem, 4(3), 300–316. https://doi.org/10.1002/cssc.201000354
27. Ahlbrecht, J., Lutz, M. D. R., Jost, V., Färber, M., Bräse, S., & Wuitschik, G. (2026). Which Reaction Conditions Work on Drug-Like Molecules? Lessons from 66,000 High-Throughput Experiments. ACS Central Science, 12(2), 222–232. https://doi.org/10.1021/acscentsci.5c02031
28. Fitzner, M., Wuitschik, G., Koller, R. J., Adam, J.-M., Schindler, T., & Reymond, J.-L. (2020). What can reaction databases teach us about Buchwald–Hartwig cross-couplings? Chemical Science, 11(48), 13085–13093. https://doi.org/10.1039/D0SC04074F
29. Kalikadien, A. V., Valsecchi, C., van Putten, R., Maes, T., Muuronen, M., Dyubankova, N., Lefort, L., & Pidko, E. A. (2024). Probing machine learning models based on high throughput experimentation data for the discovery of asymmetric hydrogenation catalysts. Chemical Science, 15(34), 13618–13630. https://doi.org/10.1039/D4SC03647F
30. King-Smith, E., Berritt, S., Bernier, L., Hou, X., Klug-McLeod, J. L., Mustakis, J., Sach, N. W., Tucker, J. W., Yang, Q., Howard, R. M., & Lee, A. A. (2024). Probing the chemical 'reactome' with high-throughput experimentation data. Nature Chemistry, 16(4), 633–643. https://doi.org/10.1038/s41557-023-01393-w
31. Desai, B., & Kappe, C. O. (2005). Heterogeneous Hydrogenation Reactions Using a Continuous Flow High Pressure Device. Journal of Combinatorial Chemistry, 7(5), 641–643. https://doi.org/10.1021/cc050076x
32. Schneider, N., Lowe, D. M., Sayle, R. A., Tarselli, M. A., & Landrum, G. A. (2016). Big Data from Pharmaceutical Patents: A Computational Analysis of Medicinal Chemists' Bread and Butter. Journal of Medicinal Chemistry, 59(9), 4385–4402. https://doi.org/10.1021/acs.jmedchem.6b00153
33. Shearer, J., Castro, J. L., Lawson, A. D. G., MacCoss, M., & Taylor, R. D. (2022). Rings in Clinical Trials and Drugs: Present and Future. Journal of Medicinal Chemistry, 65(13), 8699–8712. https://doi.org/10.1021/acs.jmedchem.2c00473

34. Collins, K. D., Rühling, A., & Glorius, F. (2014). Application of a robustness screen for the evaluation of synthetic organic methodology. Nature Protocols, 9(6), 1348–1353. https://doi.org/10.1038/nprot.2014.076
35. Grosjean, H., & Biggin, P. C. (2026). Developments and challenges in hit progression within fragment-based drug discovery. Nature Communications, 17(1), 2226. https://doi.org/10.1038/s41467-026-68941-z
36. Erdem Günay, M., & Yıldırım, R. (2021). Recent advances in knowledge discovery for heterogeneous catalysis using machine learning. Catalysis Reviews, 63(1), 120–164. https://doi.org/10.1080/01614940.2020.1770402
37. McDonald, G. C. (2009). Ridge regression. Wiley Interdisciplinary Reviews: Computational Statistics, 1(1), 93–100. https://doi.org/10.1002/wics.14
38. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B: Statistical Methodology, 58(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x
39. Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society Series B: Statistical Methodology, 67(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x
40. Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research, 1(Jun), 211–244.
41. Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (2017). Classification and regression trees (First). Chapman and Hall/CRC.
42. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
43. Breiman, L. (1996). Bagging predictors. Machine Learning, 24, 123–140. https://doi.org/https://doi.org/10.1007/BF00058655
44. Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1
45. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 39(5), 1189–1232.

46. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785
47. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30.
48. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. Advances in Neural Information Processing Systems, 31.
49. Ortiz-Villaseñor, D., Trujillo-Hernández, G., Real-Moreno, O., Castro-Toscano, M. J., Medina-Madrazo, L. D., & Barrera-Román, D. (2025). K-Nearest Neighbors Regression and Applications. In Exploring Psychology, Social Innovation and Advanced Applications of Machine Learning (pp. 295–316). IGI Global Scientific Publishing.
50. Gibbs, M. N. (1998). Bayesian Gaussian processes for regression and classification [PhD Thesis, University of Cambridge].
51. Drucker, H., Burges, C. J., Kaufman, L., Smola, A., & Vapnik, V. (1996). Support vector regression machines. Advances in Neural Information Processing Systems, 9.
52. Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. Heliyon, 4(11), e00938. https://doi.org/10.1016/j.heliyon.2018.e00938
53. García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: Methods and prospects. Big Data Analytics, 1(1), 1–22. https://doi.org/10.1186/s41044-016-0014-0
54. Luengo, J., García-Gil, D., Ramírez-Gallego, S., García, S., & Herrera, F. (2020). Big data preprocessing. Cham: Springer.
55. Landrum, G. (2013). Rdkit documentation. Release, 1(1–79), 4.
56. Cano, G., Garcia-Rodriguez, J., Garcia-Garcia, A., Perez-Sanchez, H., Benediktsson, J. A., Thapa, A., & Barr, A. (2017). Automatic selection of molecular descriptors using random forest: Application to drug discovery. Expert Systems with Applications, 72, 151–159. https://doi.org/10.1016/j.eswa.2016.12.008

57. Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A., Deng, D., & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. WIREs Data Mining and Knowledge Discovery, 13(2), e1484. https://doi.org/10.1002/widm.1484
58. Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. Multimedia Tools and Applications, 80(5), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6
59. Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100–108. JSTOR. https://doi.org/10.2307/2346830
60. Ding, C., & He, X. (2004). K-means clustering via principal component analysis. Proceedings of the Twenty-First International Conference on Machine Learning, ICML '04, 29. https://doi.org/10.1145/1015330.1015408

Chapter 8

Conclusion

8.1 Overview of the thesis

This thesis investigated the application of data-driven methodologies within chemical engineering, with a particular emphasis on predictive modelling, optimization, and engineering decision support. The work was motivated by the increasing complexity of modern chemical engineering systems, which are often characterized by nonlinear behaviour, strong variable interactions among variables, heterogeneous data sources, and competing technical objectives. In such contexts, conventional modelling approaches remain indispensable; however, there is an increasing need for complementary approaches that can efficiently extract knowledge from available data, facilitate exploration of complex design spaces, and assist in the identification of practically meaningful engineering solutions.

Within this context, the thesis examined how ML-based predictive frameworks can be integrated with optimization and decision-support strategies to address a range of chemical engineering problems. Rather than treating prediction as an isolated task, the thesis adopted a holistic perspective in which predictive models were embedded within complete workflows that explicitly accounted for data preparation, trade-off analysis, and result interpretation. This overall structure connected the different studies presented in the thesis, despite their application to different systems and scales.

The thesis was organized around two main groups of studies. The first group focused on process-level applications, with the styrene production process as a representative case study. The second group focused on catalytic and reaction-oriented applications, including SO_2 oxidation and chemoselective hydrogenation. Across both groups, the central objective remained consistent: to investigate how data-driven methods can be used not only to improve predictive accuracy, but also to strengthen engineering analysis, optimization, and informed decision-making.

8.2 Summary of main findings

The studies on the styrene production process showed that predictive modelling can effectively support process analysis and optimization when embedded within a structured engineering

framework. The comparative modelling analysis demonstrated that both dataset size and data quality exert a strong influence on optimization outcomes. When complete, higher-quality datasets were used, the resulting Pareto fronts were smoother, more densely populated, and more consistent across models, indicating stronger generalization and more reliable optimization behaviour. In contrast, dataset reduction and the introduction of noise led to fragmentated and unstable solution spaces, with some models exhibiting greater sensitivity than others. These results showed that the usefulness of predictive models in chemical engineering cannot be evaluated solely using traditional predictive metrics; it must also be assessed in terms of robustness and engineering relevance of the optimization results they produce. The consensus-based analysis further strengthened this conclusion, showing that combining information across multiple models can provide more reliable guidance than relying on any single modelling approach.

The missing data imputation study addressed a fundamental challenge in data-driven engineering workflows: ensuring the integrity and completeness of the underlying data. Using AANN, this work demonstrated that missing process variables can be reconstructed with good accuracy across multiple dataset sizes and under noisy conditions. The results showed that imputation accuracy improves with increasing dataset size, while also depending strongly on the number of missing variables and the degree of correlation among process variables. This study reinforced the broader conclusion that data quality is not merely a preliminary technical concern, but a critical prerequisite for reliable downstream modelling, optimization, and decision support applications.

The SO_2 oxidation studies extended the thesis toward catalytic process design, demonstrating that data-driven frameworks can provide valuable support for identifying favourable trade-offs among conversion, productivity, and cost. These studies demonstrated that catalyst composition and operating conditions must be selected through careful balancing of conflicting objectives rather than isolated optimization of a single performance indicator. The results highlighted the critical influence of operating temperature, catalyst composition, and residence-time-related variables in shaping the overall process performance. Moreover, the analysis revealed that systematic optimization can uncover meaningful operating windows that may not be readily apparent from direct inspection of the data. At the same time, these chapters emphasized that the reliability of such conclusions depends strongly on data preprocessing, variable representation,

and model-development strategy, especially when working with literature-derived datasets that contain inconsistencies and heterogeneity.

The selective hydrogenation study further broadened the thesis by applying data-driven methods to reaction-specific catalyst recommendation. This work showed that the predictive performance of conversion and yield models depends strongly on the way chemical information is represented. Models relying solely on categorical encodings showed limited ability to generalize to previously unseen substrate-catalyst combinations, whereas structure-informed molecular representations significantly improved predictive transferability. Beyond predictive benchmarking, the study demonstrated how trained models can be embedded in a practical workflow for catalyst and operating-condition selection under chemoselectivity constraints. The results also showed that the framework can reveal catalyst-performance patterns, including which catalysts tend to be broadly active, insufficiently active, or better suited to selective hydrogenation under specific substrate-competition scenarios. The results revealed a directional catalyst-recommendation landscape across competing substrate pairs and showed that high-performance operating solutions tend to cluster into a limited number of recurring regimes rather than being randomly distributed throughout the design space. Overall, the study illustrates how predictive modelling can be translated into actionable support for reaction planning.

Taken together, the findings across all chapters support a consistent conclusion: data-driven methodologies can play a meaningful role in chemical engineering when they are developed within rigorous and application-aware workflows that integrate data preparation, predictive modelling, optimization, and interpretation.

8.3 Main contributions of the thesis

This thesis makes contributions at both the methodological and application levels. From a methodological perspective, the thesis demonstrates that ML should not be treated as a standalone predictive tool, but rather as the predictive core of a broader engineering workflow. Across the different case studies, the work showed that the greatest value of data-driven methods emerges when prediction is tightly linked to optimization and decision support. This integrated framework enables a shift from merely estimating system behaviour to systematically identifying preferred

operating regions, catalyst formulations, or reaction pathways in a more structured and transparent manner.

A second methodological contribution is the emphasis placed on data quality, dataset size, preprocessing strategies, and feature representation. The findings consistently showed that these factors are not merely secondary implementation choices; rather, they are fundamental determinants of model reliability and of the engineering insights that can be drawn from a data-driven study.

A third contribution is the comparative evaluation of multiple ML model families across different chemical engineering applications. By examining linear models, tree-based methods, ensemble models, kernel-based approaches, Gaussian process regression, and neural networks across process and reaction datasets, the thesis provides practical insight into the applicability, strengths, and limitations of different algorithms. These comparative studies show that no single model is universally superior; instead, model suitability depends on the dataset size, degree of nonlinearity, feature representation, noise level, and intended downstream use in optimization or decision support.

A fourth contribution is the emphasis placed on engineering-oriented evaluation of predictive models. Rather than assessing model performance only through statistical indicators, this thesis showed that models should also be judged according to the quality, stability, and interpretability of the solutions they generate when used for optimization or design support. This perspective is particularly important in chemical engineering, where predictive models are rarely the final objective and are instead used to inform decisions.

The thesis also provides a practical guide for researchers and new users seeking to apply ML-based methods in chemical engineering. Through step-by-step workflows covering data preparation, model selection, hyperparameter tuning, validation, optimization, and decision-making, the work illustrates how data-driven frameworks can be implemented in a systematic and reproducible manner. This guidance is especially relevant for users who wish to move beyond isolated model fitting and develop complete engineering workflows that connect prediction with actionable process or catalyst-design decisions.

From an application standpoint, the thesis makes several concrete contributions. In the styrene case study, an integrated workflow was established that combines predictive modelling with trade-off analysis and consensus-based decision support. The missing-data study demonstrated both the capability and limitations of AANNs in reconstructing missing process information across diverse scenarios. In the SO_2 oxidation studies, the work developed data-driven strategies for catalyst and process optimization, while also identifying practical challenges associated with building reliable models from literature-derived data. Finally, the selective hydrogenation study presented a workflow that links predictive modelling with catalyst recommendation, catalyst-performance interpretation, operating-condition categorization, and practical support for reaction planning.

Together, these contributions offer a coherent and transferable perspective on the application of data-driven methods in chemical engineering, illustrating how such approaches can support analysis, design, and decision-making across a wide range of problem classes.

8.4 Limitations and future research directions

Although the results of this thesis are encouraging, several limitations remain and point to important directions for future work.

One key limitation concerns data quality, coverage, and representativeness, particularly for catalytic and reaction-oriented systems. In several studies, the data were obtained from diverse literature sources or from relatively constrained experimental spaces. As a result, the predictive models and optimization results are necessarily conditioned by the consistency, completeness, and diversity of the available data. Future work would benefit from larger and more systematically curated datasets, together with more careful treatment of data quality, in order to improve both predictive robustness and generalizability [1,2].

A second limitation relates to extrapolation and transferability. While some studies showed strong predictive performance within the investigated spaces, generalization to new process conditions, catalyst systems, or chemical structures remains challenging, especially when models remain purely data-driven [3,4]. This was especially evident in the selective hydrogenation work, where generalization improved substantially when more informative chemical descriptors were introduced [3]. Future studies should therefore continue to explore richer and more chemically or

physically meaningful representations, as well as hybrid and physics-informed modelling approaches that combine data-driven learning with domain knowledge [3–5].

A third important direction concerns uncertainty and interpretability. Although this thesis emphasized engineering usefulness and structured workflows, more explicit incorporation of uncertainty quantification would strengthen confidence in model-based optimization and decision support [4,6]. This is particularly important when such methods are intended for high-impact engineering choices, where opaque predictions are difficult to trust. Similarly, additional work on model interpretability would help improve transparency and facilitate adoption by practitioners [1,4,7].

Experimental validation also remains an important future step. While the thesis developed predictive and optimization-based frameworks that provide strong guidance, further validation through targeted experiments would help confirm the practical value of the identified solutions and strengthen the connection between computational recommendations and real-world implementation [8].

Finally, there is significant potential to further develop human-centred, user-oriented tools that make advanced data-driven methods more accessible in practice [9]. The reaction-planning application developed in the hydrogenation study illustrates one promising direction. Similar tools could be extended to other processes and catalytic systems, helping to translate complex predictive and optimization results into formats that are more readily usable by engineers and experimental researchers.

8.5 Final remarks

In conclusion, this thesis has shown that data-driven methodologies can make meaningful and versatile contributions to chemical engineering when they are developed within a rigorous, integrated, and problem-oriented framework. Across process modelling, missing-data reconstruction, catalytic process design, and reaction planning, the different studies demonstrated that predictive models can serve as a strong analytical foundation for broader engineering workflows. Their true value extends beyond output estimation to enabling systematic exploration of alternatives, revealing critical trade-offs, and supporting more transparent and informed decision-making.

The thesis thus contributes to the growing role of data-driven methods in chemical engineering by showing how the approaches can be applied across different domains while adhering to a common methodological logic. This logic begins with careful treatment of data, continues through predictive model development, and extends to optimization and practical decision support. By demonstrating the effectiveness of this progression across multiple case studies, the thesis reinforces the view that data-driven approaches can function as practical engineering tools rather than merely computational exercises.

Overall, the work presented here shows that ML and related data-driven strategies are most powerful when embedded within broader engineering workflows that respect the realities of data quality constraints, competing objectives, and the need for interpretable and actionable outcomes. In this sense, the thesis advances both the application and the methodological understanding of data-driven chemical engineering, offering a foundation for future work aimed at more adaptive, efficient, and informed engineering design and decision-making.

8.6 References

1. Dritsas, E., & Trigka, M. (2025). Exploring the Intersection of Machine Learning and Big Data: A Survey. Machine Learning and Knowledge Extraction, 7(1), 13. https://doi.org/10.3390/make7010013
2. Takahashi, K., Ohyama, J., Nishimura, S., Fujima, J., Takahashi, L., Uno, T., & Taniike, T. (2023). Catalysts informatics: Paradigm shift towards data-driven catalyst design. Chemical Communications, 59(16), 2222–2238. https://doi.org/10.1039/d2cc05938j
3. Singh, S., & Sunoj, R. B. (2023). Molecular Machine Learning for Chemical Catalysis: Prospects and Challenges. Accounts of Chemical Research, 56(3), 402–412. https://doi.org/10.1021/acs.accounts.2c00801
4. Holzinger, A., Longo, L., Cangelosi, A., & Ser, J. D. (2026). Research Frontiers in Machine Learning & Knowledge Extraction. Machine Learning and Knowledge Extraction, 8(1), 6. https://doi.org/10.3390/make8010006
5. Sharma, N., & Liu, Y. A. (2022). A hybrid science-guided machine learning approach for modeling chemical processes: A review. AIChE Journal, 68(5), e17609. https://doi.org/10.1002/aic.17609

6. Dai, J., Adhikari, S., & Wen, M. (2025). Uncertainty quantification and propagation in atomistic machine learning. Reviews in Chemical Engineering, 41(4), 333–357. https://doi.org/10.1515/revce-2024-0028
7. Esterhuizen, J. A., Goldsmith, B. R., & Linic, S. (2022). Interpretable machine learning for knowledge generation in heterogeneous catalysis. Nature Catalysis, 5(3), 175–184. https://doi.org/10.1038/s41929-022-00744-z
8. Coley, C. W., Eyke, N. S., & Jensen, K. F. (2020). Autonomous Discovery in the Chemical Sciences Part II: Outlook. Angewandte Chemie International Edition, 59(52), 23414–23436. https://doi.org/10.1002/anie.201909989
9. Tu, Z., Stuyver, T., & Coley, C. W. (2023). Predictive chemistry: Machine learning for reaction deployment, reaction development, and reaction discovery. Chemical Science, 14(2), 226–244. https://doi.org/10.1039/d2sc05089g

Appendix A: Supporting Information of Chapter 3

Harnessing Machine Learning and Optimization for Informed Chemical Engineering Decisions: A Styrene Reactor Analysis

1. Introduction

Table A.1. Searched keywords in Scopus and Web of Science.

Scopus	(TITLE-ABS-KEY (("machine learning" OR "ML" OR "AI" OR "artificial intelligence" OR "artificial neural network*" OR "neural network*" OR "deep learning") AND ("optimi*" OR "multi-objective optimi*" OR "multiobjective optimi*" OR "multi-objective optimi*" OR "multiobjective optimi*" OR "pareto front*" OR "pareto" OR "pareto domain*") AND ("multi-criteria decision-making" OR "multicriteria decision making" OR "multi criteria decision making" OR "MCDM" OR "decision analysis" OR "decision making" OR "decision-making")) AND PUBYEAR > 1999 AND PUBYEAR < 2025) AND PUBYEAR > 2003 AND PUBYEAR < 2025 AND (LIMIT-TO (DOCTYPE , "ar"))
Web of Science	TS=(("machine learning" OR "ML" OR "AI" OR "artificial intelligence" OR "artificial neural network*" OR "neural network*" OR "deep learning") AND ("optimi*" OR "multi-objective optimi*" OR "multiobjective optimi*" OR "pareto front*" OR "pareto" OR "pareto domain*" OR "evolutionary algorithm*") AND ("multi-criteria decision-making" OR "multicriteria decision making" OR "multi criteria decision making" OR "MCDM" OR "decision analysis" OR "decision making" OR "decision-making")) AND PY=(2000-2024) AND (DT=("Article" OR "Proceedings Paper"))

To further explore the thematic structure of the literature involving ML, MOO, and MCDM, a keyword co-occurrence analysis was conducted. The resulting visualization identifies distinct clusters of frequently co-occurring keywords, highlighting the interdisciplinary nature of the field. Prominent clusters include themes centred on machine learning algorithms (e.g., random forest, support vector machine, neural networks), optimization techniques (e.g., genetic algorithm, surrogate models, Pareto front), and application contexts such as energy systems, process scheduling, and uncertainty analysis. Keywords such as "intelligence," "uncertainty," and "objective function" appear at the intersections of these clusters, indicating a convergence of data-driven modelling and decision-making under complex conditions. Overall, the network map highlights the diversity of research themes and the growing integration of data science, optimization, and decision analysis in chemical engineering contexts.

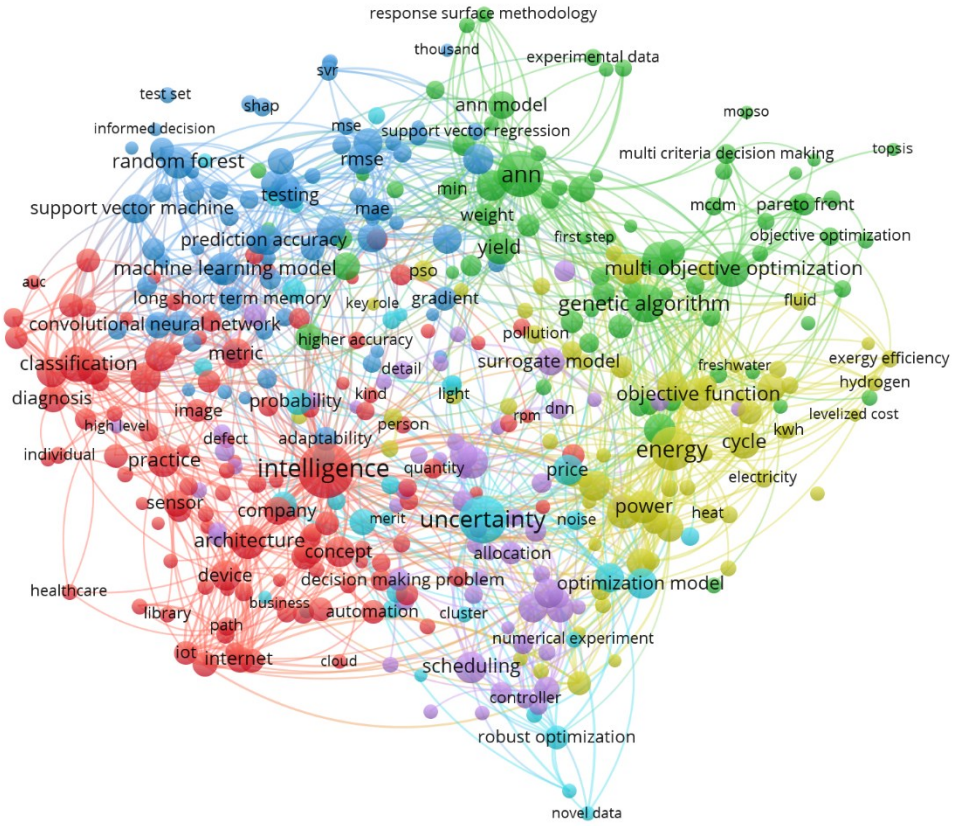


Figure A.1. Keyword co-occurrence network showing dominant research themes and interconnections in studies combining ML, MOO, and MCDM. Node size indicates frequency, and colours represent thematic clusters.

To examine the evolution of research priorities over time, a temporal keyword trend analysis was performed. The resulting area chart displays the annual frequency of selected keywords from 2010 to 2021, providing insights into both emerging and dominant themes within the literature. Broad terms such as machine learning, optimization, and artificial intelligence exhibit a pronounced and sustained increase in usage, particularly after 2018, reflecting the growing interest in data-driven and intelligent systems. More specialized keywords such as multi-objective optimization, genetic algorithm, and reinforcement learning also show notable growth, suggesting a shift toward advanced and application-oriented methodological developments. The increasing prevalence of deep learning and neural networks further indicates an increasing reliance on complex computational architectures for modelling and optimization in engineering research.

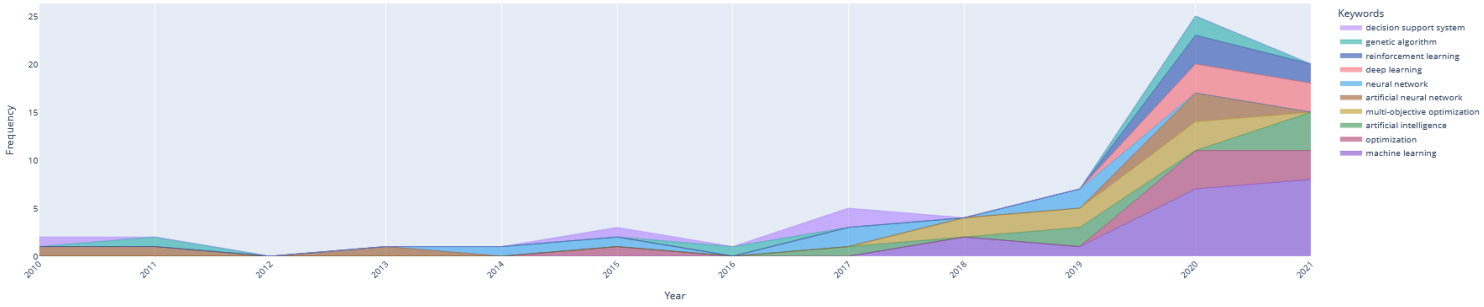


Figure A.2. Annual keyword frequency trends (2010–2021) showing the rise of key concepts such as machine learning, optimization, and artificial intelligence in the literature combining ML, MOO, and MCDM.

To capture the most frequently used terms in the literature combining ML, MOO, and MCDM, a word cloud analysis was generated. This visualization highlights the dominant concepts by proportionally scaling the font size of each word according to its frequency. Prominent terms such as *decision*, *model*, *system*, *data*, and *optimization* reflect the central focus of the reviewed studies on data-driven decision-making and system modelling. Additional frequently occurring words, including *method*, *performance*, *energy*, *network*, and *algorithm*, suggest a strong emphasis on computational techniques and their practical applications in energy and process systems. The prevalence of these keywords further reinforces the interdisciplinary nature of research situated at the intersection of machine learning, optimization, and engineering decision support.

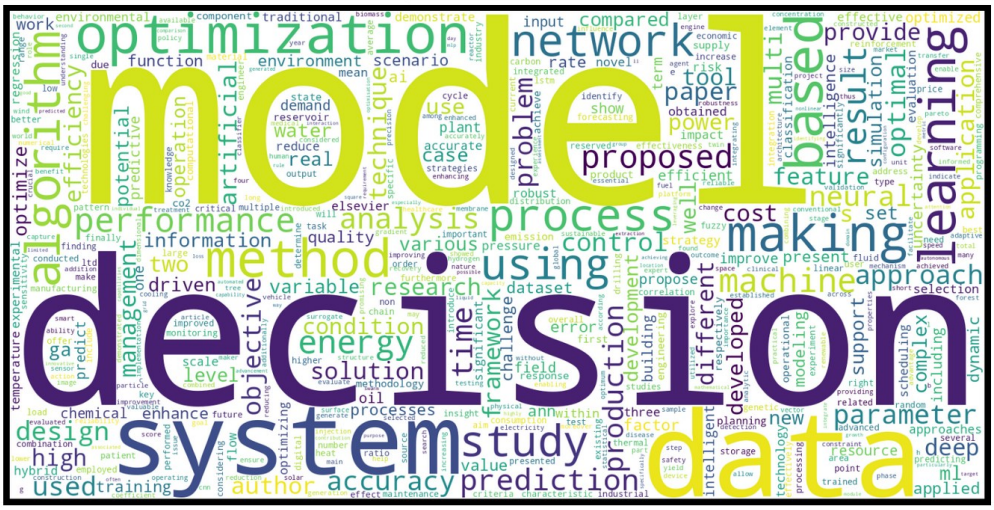


Figure A.3. Word cloud illustrating the most frequent keywords in the reviewed literature on ML, MOO, and MCDM. Larger font size indicates higher frequency of occurrence.

Table A.2. provides a summary of some selected previous studies, each assigned a unique identifier for reference. To facilitate analysis, the reviewed papers were first organized chronologically by year of publication. Each study was then classified into an application domain based on the nature of its case study. This domain-specific classification enables clearer interpretation of how ML, MOO, and MCDM techniques are integrated across different research areas. The energy systems domain which includes applications such as power generation, hydrogen production, renewable energy systems, and biofuel optimization, was the most widely represented, with studies including [3, 4, 5, 8, 20–23, 29–31, 33, 34, 40, 42–44]. Heat transfer and thermal management constituted another major category, covering investigations of cooling technologies and finned systems [1, 6, 7, 12, 14, 24, 36, 39]. A substantial number of papers also focused on process simulation and optimization, including chemical reactors, separation processes, and process control systems [2, 9, 10, 16, 25, 26, 28, 32, 35, 37, 38, 41]. In the materials and construction domain, studies dealt with topics such as hybrid nanofluids, concrete mix design, and thermal coatings [11, 13, 15, 27, 39]. This structured categorization provides a comprehensive overview of the diverse application areas in which ML, MOO, and MCDM are being applied across various fields and lays the groundwork for comparative analysis across application areas.

Abbreviations used in Table A.2. are listed as follows: Artificial neural network (ANN), Adaptive neuro-fuzzy inference system (ANFIS), Non-dominated sorting grey wolf optimizer algorithm (NSGWOA), Non-dominated sorting genetic algorithm (NSGA-II), Technique for order preference by similarity to ideal solution (TOPSIS), Response surface methodology (RSM), Multi-objective bat algorithm (MOBA), Višekriterijumsko kompromisno rangiranje (VIKOR), Multi-objective grey wolf optimizer (MOGWO), Group method of data handling-type (GMDH), Preference ranking organization method for enrichment evaluation (PROMETHEE), Multi-objective thermal exchange optimization (MOTEO), Linear programming technique for multidimensional analysis of preference (LINMAP), Multi-objective particle swarm optimization (MOPSO). Deep neural network (DNN), combinatorial (COMBI), Physics-informed neural network (PINN), Generalized regression neural network (GRNN), Approximation-guided evolution (AGE), Multi-objective evolutionary algorithm (MOEA), Genetic algorithm (GA), Genetic algorithm backpropagation (GABP), Multilayer perceptron (MLP), Multi-attributive border approximation area comparison (MABAC), Preference selection index (PSI), Feedforward neural network (FNN), Recurrent neural network (RNN), Convolutional neural network (CNN), Gated recurrent unit (GRU), Long short-term memory (LSTM), Criteria importance through intercriteria correlation (CRITIC), Preference ranking on the basis of ideal-average distance (PROBID), Weighted sum technique (WST), Simple additive

weighting (SAW), Gaussian process regression (GPR), Support vector regression (SVR), Random forest (RF), Differential evolution (DE), Grey relational analysis (GRA), Least squares (LS), Fuzzy C-means clustering (FCM), Multi-objective genetic algorithm (MOGA), Mixed-integer nonlinear programming (MINLP), Simulated Annealing (SA).

Table A.2. Summary of previous studies.

ID	Reference	Case study	Data origin	Inputs	Outputs	ML models	Hyperparameter tuning	Selected ML	Optimization algorithm	MCDM
1	[1]	Maxwell fluid in microchannel	Simulated data from analytical and numerical models	Relaxation time, hall current, ion slip, nanofluid volume fraction, Hartmann number	Electroviscous heat transfer rate (maximized), entropy generation (minimized)	ANN, ANFIS	HP tuning with trial and error	ANFIS	NSGWOA, NSGA-II	TOPSIS
2	[2]	CO_2-WAG enhanced oil recovery (EOR)	Synthetic reservoir simulation data	CO_2 injection rate, water injection rate, injection timing, reservoir controls	Cumulative oil production (maximized), CO_2 storage (maximized), water production (minimized), net present value (maximized)	Ensemble of ML models	HP tuning with trial and error	Stacked Learning	NSGA-II	Clustering-based classification
3	[3]	Photovoltaic Thermal System with Solar Collector	Simulation-based data	Solar radiation, mass flow rate, inlet temperature, wind speed	Electrical efficiency (maximized), thermal efficiency (maximized), entropy generation (minimized)	RSM	RSM coefficients statistically validated	RSM	MOBA	VIKOR
4	[4]	Solar-assisted ocean thermal energy conversion for hydrogen production	Simulation data via Engineering equation solver (EES)	Working fluid type, mass flow rate, solar irradiation, collector area	Exergy efficiency (maximized), hydrogen production rate (maximized), total investment rate (minimized)	ANN	HP tuning with trial and error	ANN	MOGWO	TOPSIS
5	[5]	PCM-based Photovoltaic Thermal (PVT) System	Simulated dataset for PCM-based system	PCM melting temperature, PCM thickness, solar radiation, ambient temperature	Electrical power (maximized), Thermal power (maximized), entropy generation (minimized)	GMDH-type ANN	Whole search strategy	GMDH-ANN	MOTEO	PROMETHEE
6	[6]	T-shaped Porous Fin for Heat Dissipation	Simulation results (ANSYS Workbench, specific number not stated)	Air inlet temperature, Air inlet velocity, Porosity, Fin length, Height-width ratio	Pumping-power consumption (minimized), temperature difference (minimized)	ANN	HP tuning with trial and error	ANN	NSGA-II	LINMAP

7	[7]	Hybrid heat sink in 3D electronic device	Simulation data (77 samples from ANSYS Workbench)	Aspect ratio of microchannel (γ_c), number of elements (N)	Dimensionless temperature difference (ΔT) (maximize), dimensionless pump power (W_P) (maximized)	ANN	Three-layer ANN (40-20-10), performance validated against CFD	ANN	NSGA-II	LINMAP, TOPSIS, Shannon Entropy
8	[8]	Hydrogen liquefaction process optimization	Simulation data from ASPEN HYSYS & SimaPro (Dataset size is not specified)	Suction pressure, H_2 mol fraction, discharge pressure, flow rate of precooling	CO_2 emissions, max Exergy efficiency (maximized), rate of return (maximized)	GA-GMDH	GMDH model coefficients tuned using GA	GA-GMDH	NSGA-II, MOPSO, MOGWO	TOPSIS, LINMAP, Bellman-Zadeh fuzzy
9	[9]	CO_2 capture in PSA unit with modified Skarstrom cycle	Simulation data (3000 samples generated via LHS from PDE-based virtual plant)	Step durations (pressurization, adsorption, heavy/light reflux, depressurization)	CO_2 purity (maximized), CO_2 recovery (maximized)	DNN	Hyperband	DNN	constraints-based sliding PSO	Not used (feasible operational region evaluated instead)
10	[10]	Naphtha cracking furnace (industrial plant using SPYRO simulation)	Industrial data (783 data points from 9 years, used in SPYRO simulation)	Coil outlet temperature (COT), molar fractions of NP5, IP4, Aro6, NP4, NP6, N7, IP5, N6, and N5	Ethylene yield (maximized), Propylene yield (maximized), Heat duty (minimized)	DNN	Heuristically tuned	DNN	NSGA-II	Selected with human experts
11	[11]	Water-based hybrid nanofluids with MWCNTs and oxide nanoparticles (Al_2O_3, ZnO, CeO_2, TiO_2)	Experimental data (192 samples, 48 per nanofluid type)	System temperature, volume fraction of nanoparticles, density of oxide nanoparticles, thermal conductivity of oxide nanoparticles, specific heat capacity of oxide nanoparticles	Density ratio (minimized), viscosity ratio (minimized), specific heat capacity ratio (maximized), thermal conductivity ratio (maximized)	GMDH-NN, COMBI algorithm	HP tuning with trial and error	COMBI	NSGA-II	TOPSIS, VIKOR
12	[12]	3D finned heat sink system (forced air cooling)	Simulation data via CFD	Fin height, fin thickness, central fin length, side fin length, inlet air velocity	Mean surface temperature of the heat source (minimized), pressure drop (minimized)	PINN, hybrid PINN	6 hidden layers with 256 neurons per layer, trained over 1 million steps using Monte Carlo integration and SDF-weighted loss functions	Hybrid PINN	NSGA-II	TOPSIS

13	[13]	Methanol synthesis process with recycle stream	Aspen Plus simulation (100 samples generated using Latin Hypercube Sampling)	Reactor temperature, reactor pressure, reflux ratio, column pressure, feed flow rate, recycle gas rate	Total energy cost (minimized), resource utilization (maximized), CO_2-equivalent emissions per energy input (minimized)	GRNN	HP tuning with trial and error	GRNN	NSGA-II	TOPSIS
14	[14]	Perforated-finned heat sink design	Simulation data (CFD modelling in ANSYS Fluent, 147 CFD samples)	Perforation shape, perforation volume ratio, Reynolds number	Thermal performance (maximized), thermohydraulic performance (maximized), thermovolumetric performance (maximized)	GMDH-NN, COMBI	HP tuning with trial and error	COMBI	NSGA-II	TOPSIS, VIKOR
15	[15]	Rice husk ash concrete (RHC) mixture design	Experimental dataset compiled from literature (1150 samples after outlier removal)	Cement content, rice husk ash content, coarse aggregate content, fine aggregate content, water content, superplasticizer content, curing age	Compressive strength (maximized), materials cost (minimized), embodied carbon (minimized), embodied energy (minimized)	Gradient boosting (CatBoost, XGBoost, LightGBM)	Bayesian optimization using Optuna for all boosting models	CatBoost	MOEA II (AGE-MOEA II), compared with NSGA-II, U-NSGA-III, and AGE-MOEA	Pseudo-weight vector approach
16	[16]	Stirred tank mixing optimization	CFD simulation + GA-GABP ANN model + 20 Latin Hypercube samples	Impeller speed, Impeller diameter, baffle width, clearance	Power consumption (minimized), mixing degree (maximized), suspension uniformity (maximized)	GA-GABP ANN	GA	GA-GABP ANN	NSGA-II	TOPSIS
17	[17]	Catalytic ozonation of acid red 88 (AR88) with Fe_3O_4	Experimental (78 samples)	Initial AR88 concentration, catalyst dosage, pH, inlet air flow rate, batch time	Degradation efficiency (maximized), catalyst dosage (minimized)	MLP	Grid search	MLP	MOPSO	TOPSIS, MABAC
18	[18]	Radially stirred tank	CFD simulation data	Impeller diameter, blade width, blade length, number of blades	Power consumption (minimized), mixing time (minimized)	XGBoost, GBDT, LGBM, MLP	HP tuning with trial and error	XGBoost	NSGA-II	TOPSIS, LINMAP

19	[19]	Turning process optimization (90MnCrV7)	Experimental machining dataset (exact sample size not specified)	Cutting speed, feed rate, depth of cut, insert radius	Surface roughness (minimized), cutting forces (minimized)	ANN	HP tuning with trial and error	ANN	Modified PSI-TOPSIS	PSI-TOPSIS
20	[20]	Biomass gasification + heat recovery for H_2 and Power	Simulation data (no size provided)	Working fluid type, mass flow rate, electrolyzer input, Organic cycle parameters	Energy efficiency (maximized), exergy efficiency (maximized), CO_2 emissions (minimized)	ANN	HP tuning with trial and error	ANN	NSGA-II	LINMAP
21	[21]	Methanol production with CO_2 feed optimization	Simulated methanol production system with triple CO_2 feed	Feed flow rates, reformer configuration, pressure, temperature	CO2 emissions (minimized), methanol production (maximized), Energy use (minimized), CH_4 conversion (maximized)	FNN, RNN, CNN	Optuna	CNN	NSGA-III	TOPSIS
22	[22]	Marine engine system with liquefied hydrogen	Simulated integrated marine engine system	Heat integration parameters, engine output, cycle temperatures and pressures	Exergy efficiency (maximized), CO_2 emissions (minimized)	ANN	HP tuning with trial and error	ANN	NSGA-II	TOPSIS
23	[23]	Plate battery cooling with MEPCM and chaotic flow	Simulation data (3 cases, 27 samples per case)	Reynolds number, MEPCM volume fraction	Nusselt number (maximized), pressure drop (minimized)	ANN	HP tuning with trial and error	ANN	NSGA-II	TOPSIS, LINMAP, Shannon Entropy
24	[24]	Slot jet impingement/microchannel heat sink	Simulation data (75 CFD samples)	Channel unit width (W1), channel width ratio (φ), channel height (H_3)	Average heated surface temperature (T_{heat}) (minimized), total pressure drop (ΔP) (minimized)	ANN	HP tuning with trial and error	ANN	MOGA, MOPSO	TOPSIS
25	[25]	Model predictive control of continuous stirred tank reactor (CSTR)	Simulation data (non-isothermal CSTR using first-principles and RNN-based surrogate models)	Inlet concentration of component A, heat input rate, initial reactor temperature, feed flow rate, control horizon data	Tracking error (minimized), Control effort (minimized), Energy consumption (minimized)	RNN, GRU, LSTM	HP tuning with trial and error	LSTM	NSGA-II	CRITIC, PROBID, WST

26	[26]	LPG Thermal cracking Process	Simulation data from solved mathematical model	Steam-to-feed ratio, coil outlet pressure, coil outlet temperature	Olefins yield (maximized), CO_2 emissions (minimized), coke formation (minimized)	MLP	HP tuning with trial and error	MLP	MOPSO	CRITIC, MABAC, PROBID, sPROBID, SAW, TOPSIS
27	[27]	Steel alloy optimization for hardenability	1080 experimental steel samples	Alloy element compositions (e.g., Cr, Mn, Mo, Ni, V)	Jominy hardness (maximized), alloy cost (minimized)	GPR, SVR, RF, ANN	HP tuning with trial and error	GPR	Constrained optimization	None
28	[28]	Wet CO_2-to-methanol process	Simulated process data	Reactor pressure, reactor temperature, reflux ratio, reboiler ratio	Methanol yield (maximized), net CO_2 reduction (maximized), production cost (minimized)	ANN	HP tuning with trial and error	ANN	DE	None
29	[29]	Solar + NG-based cogeneration plant	CFD and simulation models; 5000 samples	Swirling number, pressure ratio, temperature, flow rate	Exergy efficiency (maximized), cost rate (minimized), CO_2 (minimized)	SVR	Grey Wolf Optimization	SVR	Grasshopper Optimization	TOPSIS
30	[30]	Supercritical water gasification process	Data from supercritical water gasification dataset (295 samples)	Feed composition: C, H, N, O, ash percentage operating conditions: solid content, T, P, residence time	Yields of CH_4, CO, CO_2, H_2 H_2 yield (maximized), the rest yields (minimized)	RF, SVM, MLP, and gradient boosting	Particle swarm optimization	Gradient boosting	NSGA-II	GRA, MABAC, PROBID, SAW, and TOPSIS
		Combustion process in a power plant	Data from the gas turbine-based power plan (36733 samples)	temperature (T), ambient pressure (AP), humidity (H), air filter pressure difference (AFDP), gas turbine exhaust pressure (GTEP), turbine inlet temperature (TIT), turbine after temperature (TAT), compressor discharge pressure (CDP)	Turbine energy yield (TEY) (maximized), total emissions of harmful pollutants NO_x and CO (PNC) (minimized)	RF, SVM, MLP, and gradient boosting	Particle swarm optimization	Gradient boosting	NSGA-II	GRA, MABAC, PROBID, SAW, and TOPSIS

31	[31]	CO_2 capture and utilization in urea production at KPIC (Iran)	Simulation data from Aspen HYSYS (100 samples)	Lean monoethanolamine (MEA) temperature, Lean MEA loading (mol CO_2/mol MEA), Absorber column packing height, Stripper column packing height	Exergy efficiency (maximized), Total cost rate (minimized)	ANN	ANN trained on 100 random sets with GA-based optimization	ANN	GA and NSGA-II	None
32	[32]	Acoustic microfluidic system for submicron bioparticle separation	Experimental and numerical data from full factorial DOE (5 level, 3 parameters = 125 fabricated samples)	Electrode length (L1), half distance between electrodes (L2), number of fingers (NF)	Reflection coefficient (minimized), quality factor (minimized)	ANN	HP tuning with trial and error	ANN	NSGA-II	None
33	[33]	Microbial electrolysis cell (MEC-assisted) anaerobic digestion	Literature-based experimental data on alkaline-pretreated waste-activated sludge	Applied voltage, operation time	Net energy output (maximized), net monetary value (maximized)	ANN	ANN topology selection via RSM	ANN	PSO	None
34	[34]	Nuclear Waste Vitrification	Simulation and historical glass formulation data (566 samples)	Glass compositions	Waste loading (maximized)	GPR, SVR, ANN	Random search	GPR with conformal uncertainty	Gradient-based optimization	None
35	[35]	Drying of olive leaves in rotary dryer	Experimental data (9 RSM-based drying trials)	Hot air inlet temperature, air flow rate, drying time, rehydration ratio	Final moisture content (minimized), moisture loss (maximized)	LS	Not applicable	LS	NSGA-II	FCM
36	[36]	Shell-and-tube heat exchanger with porous baffles	CFD simulation data (36 samples)	Number of baffles, Baffle angle, Baffle thickness	Heat transfer rate (kW) (maximized), Pressure drop (kPa) (minimized)	ANN	HP tuning with trial and error (500 trials)	ANN	MOGA	TOPSIS, LINMAP
37	[37]	Ethanol production via syngas fermentation and distillation	Simulation data (Bubble column Reactor + Aspen Plus Distillation, ~2000 samples for each subsystem)	Dilution rate, gas residence time, gas recycle ratio, cell purge fraction, reactor length, reactor volume, S:F ratio (T-01), number of stages (T-02), feed stages (vapor and liquid), reflux ratio	Capital expenditure per ethanol output (minimized), minimum ethanol selling price (minimized), thermodynamic efficiency (maximized)	ANN	HP tuning with trial and error	ANN	MOGA, based on NSGA-II	None

38	[38]	Multi-scale membrane process design	Simulated transport models (1D, 2D ion transport)	Membrane geometry, process layout, feed concentration, salt mixture	Permeate concentration (minimized), process cost (minimized)	ANN	HP tuning with trial and error	ANN	MINLP	None
39	[39]	Double-ceramic-layer thermal barrier coatings (LZ/YSZ, LZ7C3/YSZ, LaPO4/YSZ)	Simulation data (1000 samples)	Total thickness (TH), thickness ratio (TR), Poisson's ratio (ν), elastic modulus (E), coefficient of thermal expansion (α), thermal conductivity (λ) of top coating (TC)	Residual stress in YSZ (maximized), residual stress in top coating (TC) (maximized), Equivalent thermal conductivity (maximized)	ANN	HP tuning with trial and error (20 trials)	ANN	NSGA-II	TOPSIS
40	[40]	Microalgal biofuel production in 120L Photobioreactor	Simulation data (40 CFD-kinetic simulation scenarios, ~360,000 data points)	Incident light intensity, number of sparger holes, diameter of sparger holes, gas inflow rate	Biomass concentration (maximized), bisabolene production (maximized), friction velocity (minimized)	CNN	HP tuning with trial and error	CNN	Hybrid stochastic algorithm (Random Search + PSO + SA)	None
41	[41]	Betaine-enriched spelt flour extrusion	Experimental (central composite rotaTABLE design, 17 experiments)	Screw speed, feed flow rate, feed moisture content	Betaine content (maximized), energy consumption (minimized)	ANN, RSM	Grid search (ANN structure, BFGS training), polynomial fitting (RSM)	ANN	GA	None
42	[42]	SiO_2/water nanofluid flow in circular tube	Experimental data (120 data points from literature)	Reynolds number, volume concentration of nanoparticles, Inlet temperature	Nusselt number (maximized), pressure drop (minimized)	ANN	HP tuning with trial and error	ANN	GA	Compromise programming
43	[43]	Shell-and-tube heat exchanger with helical baffles	Numerical simulation data; 109 samples	Helix angle, baffle pitch, baffle overlap	Heat transfer coefficient (maximized), pressure drop (minimized)	ANN	HP tuning with trial and error	ANN	MOGA	Compromise programming
44	[44]	Nanofluid flow in chaotic channel	Two-phase CFD simulation data; 100 samples	Nanoparticle concentration, particle size	Heat transfer coefficient (maximized), pressure drop (minimized)	ANN	HP tuning with trial and error	ANN	MOGA	Compromise programming

2. Methodology

2.1 Styrene production process and dataset generation

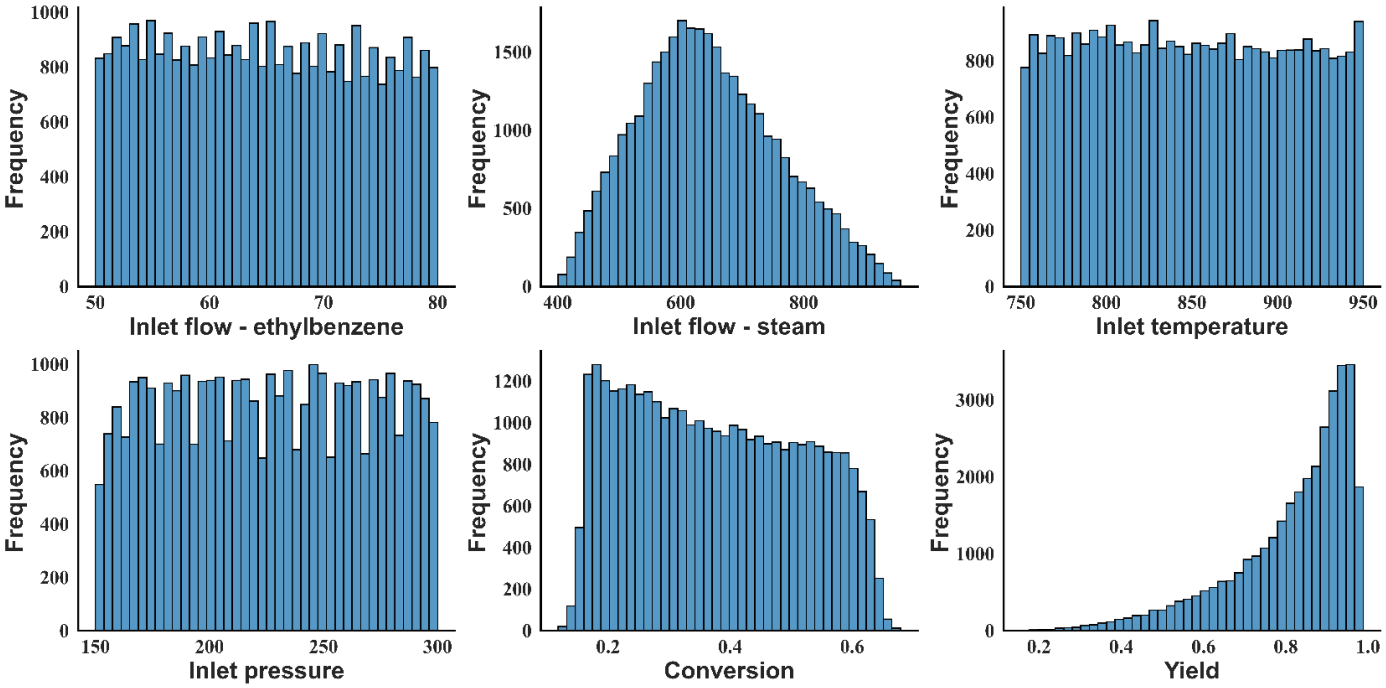


Figure A.4. Histogram of the complete dataset.

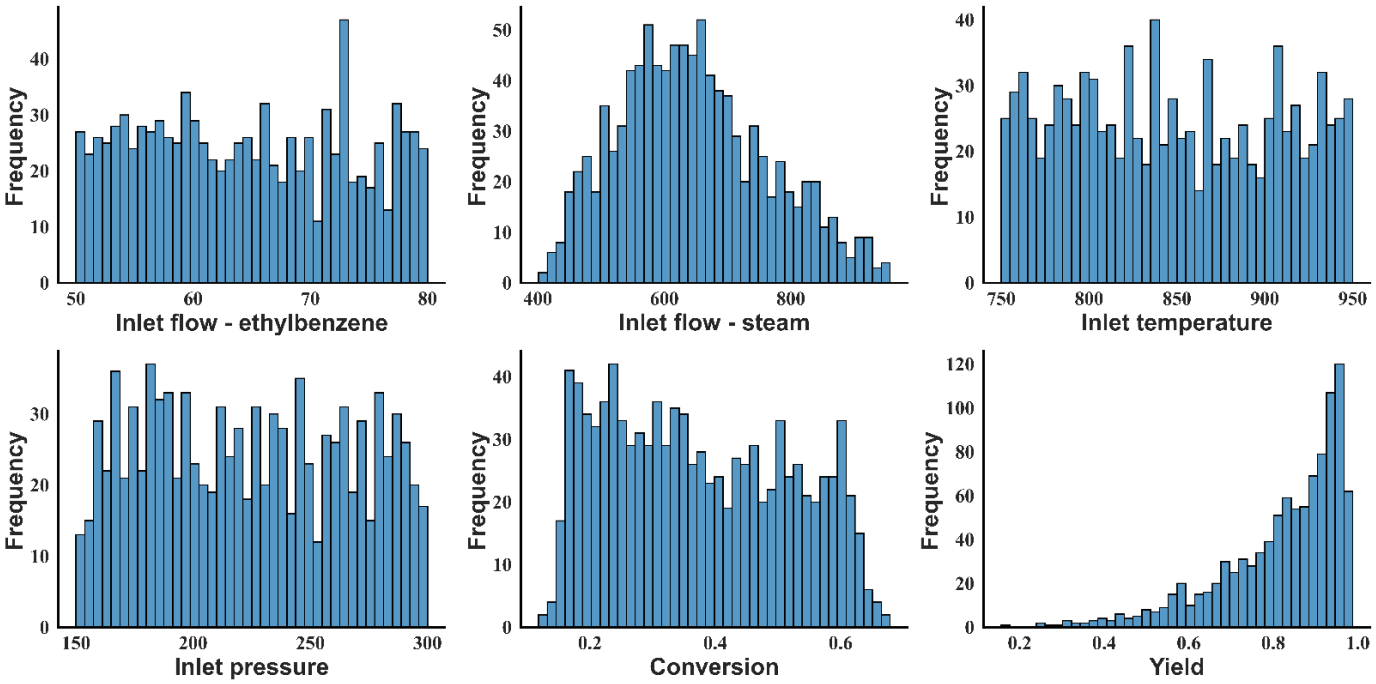


Figure A.5. Histogram of the 1000-record clean dataset.

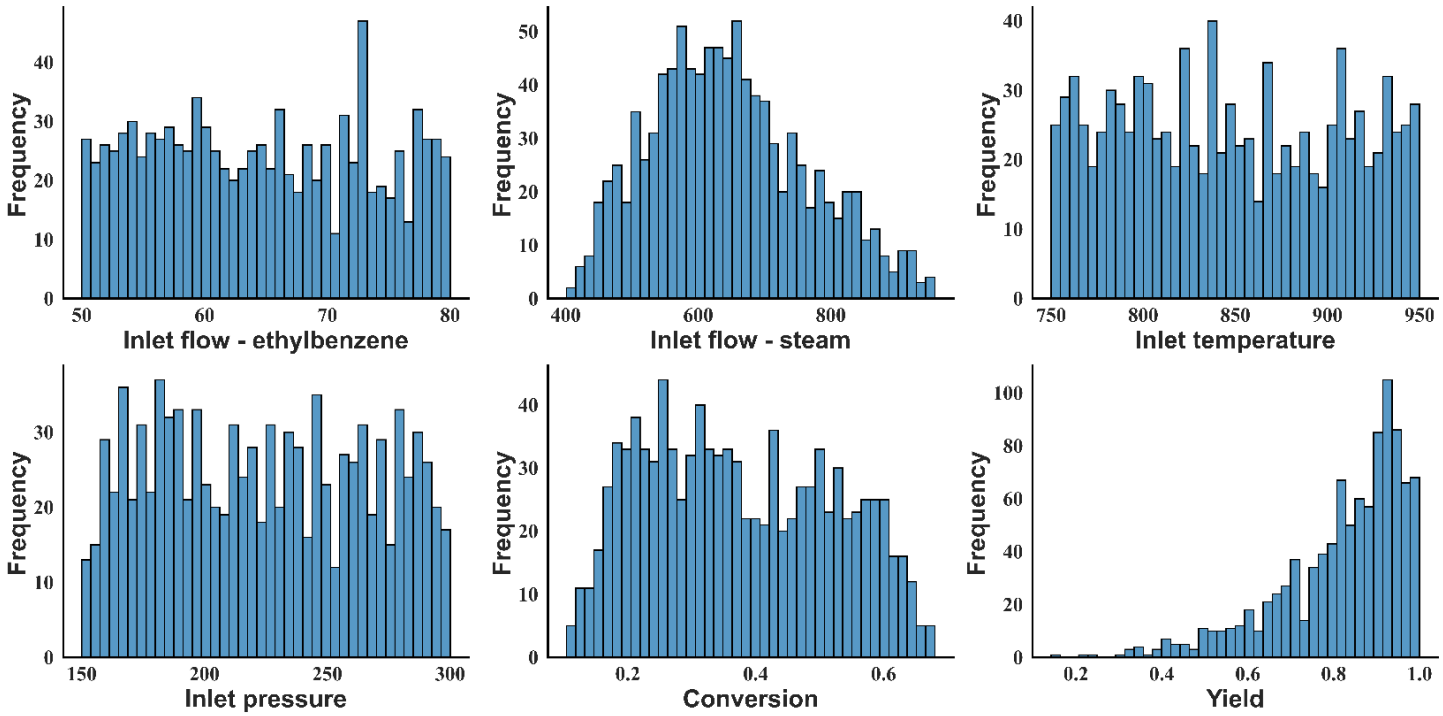


Figure A.6. Histogram of the 1000-record noisy dataset.

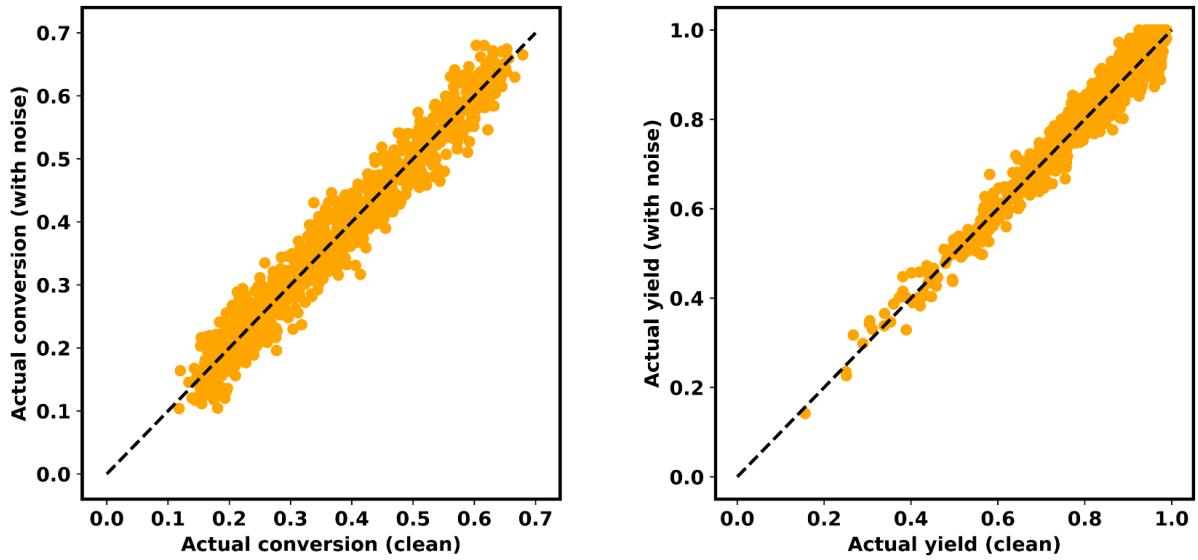


Figure A.7. Amount of noise added to conversion and yield.

2.2 Hyperparameter tuning

Table A.3. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the ANN models.

Output	Dataset	Hidden layers	Neurons	Optimizer	Learning rate	Activation	Dropout rate	Kernel initializer
Conversion	Complete	[1, **2**, 3]	[8, 16, 32, 64, 128] **8, 128, 32**	['adam', 'nadam', **'adagrad'**, 'rmsprop']	Uniform (0.0001, 0.01) **0.0038**	[**'relu'**, 'sigmoid', 'tanh']	Uniform (0.0, 0.5) **0.012**	['glorot_uniform', 'he_normal', **'lecun_normal'**]
	1000-record clean	[1, 2, **3**]	[8, 16, 32, 64, 128] **128, 128, 32**	['adam', **'nadam'**, 'adagrad', 'rmsprop']	Uniform (0.0001, 0.01) **0.0096**	['relu', **'sigmoid'**, 'tanh']	Uniform (0.0, 0.5) **0.285**	['glorot_uniform', **'he_normal'**, 'lecun_normal']
	1000-record noisy	[1, 2, **3**]	[8, 16, 32, 64, 128] **128, 64, 128**	[**'adam'**, 'nadam', 'adagrad', 'rmsprop']	Uniform (0.0001, 0.01) **0.0091**	[**'relu'**, 'sigmoid', 'tanh']	Uniform (0.0, 0.5) **0.116**	['glorot_uniform', **'he_normal'**, 'lecun_normal']
Yield	Complete	[1, **2**, 3]	[8, 16, 32, 64, 128] **16, 64, 64**	['adam', **'nadam'**, 'adagrad', 'rmsprop']	Uniform (0.0001, 0.01) **0.0073**	['relu', **'sigmoid'**, 'tanh']	Uniform (0.0, 0.5) **0.043**	['glorot_uniform', 'he_normal', **'lecun_normal'**]
	1000-record clean	[1, 2, **3**]	[8, 16, 32, 64, 128] **64, 64, 128**	[**'adam'**, 'nadam', 'adagrad', 'rmsprop']	Uniform (0.0001, 0.01) **0.0081**	['relu', **'sigmoid'**, 'tanh']	Uniform (0.0, 0.5) **0.289**	**['glorot_uniform'**, 'he_normal', 'lecun_normal']
	1000-record noisy	[1, 2, **3**]	[8, 16, 32, 64, 128] **64, 128, 128**	['adam', 'nadam', **'adagrad'**, 'rmsprop']	Uniform (0.0001, 0.01) 0.0094	[**'relu'**, 'sigmoid', 'tanh']	Uniform (0.0, 0.5) 0.087	['glorot_uniform', **'he_normal'**, 'lecun_normal']

Table A.4. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the SVR models.

Output	Dataset	C	Epsilon	kernel	degree	gamma
Conversion	Complete	np.logspace(-3, 3, 10) **46.41**	np.logspace(-3, 1, 10) **0.001**	['linear', 'poly', **'rbf'**, 'sigmoid']	[2, 3, 4, 5] **5**	['scale', 'auto'] + list(np.logspace(-3, 2, 10) **0.59**
	1000-record clean	np.logspace(-3, 3, 10) **46.41**	np.logspace(-3, 1, 10) **0.0027**	['linear', 'poly', **'rbf'**, 'sigmoid']	[2, 3, 4, 5] **5**	['scale', 'auto'] + list(np.logspace(-3, 2, 10) **0.16**
	1000-record noisy	np.logspace(-3, 3, 10) **1000**	np.logspace(-3, 1, 10) **0.0027**	['linear', 'poly', **'rbf'**, 'sigmoid']	[2, 3, 4, 5] **5**	['scale', 'auto'] + list(np.logspace(-3, 2, 10) **0.046**
Yield	Complete	np.logspace(-3, 3, 10) **10**	np.logspace(-3, 1, 10) **0.0035**	['linear', 'poly', **'rbf'**, 'sigmoid']	[2, 3, 4, 5] **5**	['scale', 'auto'] + list(np.logspace(-3, 2, 10) **2.15**
	1000-record clean	np.logspace(-3, 3, 10) **2.15**	np.logspace(-3, 1, 10) **1.29**	['linear', 'poly', **'rbf'**, 'sigmoid']	[2, 3, 4, 5] **5**	['scale', 'auto'] + list(np.logspace(-3, 2, 10) **7.74**
	1000-record noisy	np.logspace(-3, 3, 10) **215.44**	np.logspace(-3, 1, 10) **0.0077**	['linear', 'poly', **'rbf'**, 'sigmoid']	[2, 3, 4, 5] **5**	['scale', 'auto'] + list(np.logspace(-3, 2, 10) **0.012**

Table A.5. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the KNN models.

Output	Dataset	Number of neighbours	Weights	Algorithm	Leaf size	p
Conversion	Complete	np.arange (1, 50, 1) **10**	['uniform', **'distance'**]	'auto', 'ball_tree', **'kd_tree'**, 'brute']	np.arange (10, 100, 10) **15**	[1, **2**]
	1000-record clean	np.arange (1, 50, 1) **6**	['uniform', **'distance'**]	**'auto'**, 'ball_tree', 'kd_tree', 'brute']	np.arange (10, 100, 10) **30**	[1, **2**]
	1000-record noisy	np.arange (1, 50, 1) **13**	['uniform', **'distance'**]	'auto', 'ball_tree', **'kd_tree'**, 'brute']	np.arange (10, 100, 10) **60**	[1, **2**]
Yield	Complete	np.arange (1, 50, 1) **11**	['uniform', **'distance'**]	'auto', 'ball_tree', **'kd_tree'**, 'brute']	np.arange (10, 100, 10) **80**	[1, **2**]
	1000-record clean	np.arange (1, 50, 1) **6**	['uniform', **'distance'**]	'auto', 'ball_tree', 'kd_tree', **'brute'**]	np.arange (10, 100, 10) **30**	[**1**, 2]
	1000-record noisy	np.arange (1, 50, 1) **10**	['uniform', **'distance'**]	'auto', 'ball_tree', 'kd_tree', **'brute'**]	np.arange (10, 100, 10) **10**	[1, **2**]

Table A.6. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the GPR models.

Output	Dataset	Kernel function	Alpha	n_restarts_optimizer
Conversion	Complete	**Matern**	np.logspace (-5, 0, 10) **0.00046**	np.arange (0, 10, 1) **3**
	1000-record clean	**Matern**	np.logspace (-5, 0, 10) **0.278**	np.arange (0, 10, 1) **5**
	1000-record noisy	**Matern**	np.logspace (-5, 0, 10) **0.0059**	np.arange (0, 10, 1) **7**
Yield	Complete	**Matern**	np.logspace (-5, 0, 10) **1**	np.arange (0, 10, 1) **6**
	1000-record clean	**Matern**	np.logspace (-5, 0, 10) **1**	np.arange (0, 10, 1) **6**
	1000-record noisy	**Matern**	np.logspace (-5, 0, 10) **0.0016**	np.arange (0, 10, 1) **7**

Table A.7. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the DT models.

Output	Dataset	criterion	splitter	max_depth	min_samples_split	min_samples_leaf	max_features
Conversion	Complete	['squared_error', **'friedman_mse'**, 'absolute_error', 'poisson']	[**'best'**, 'random']	List (np.arange (5, 50, 5)) **30**	np.arange(2, 20, 2) **8**	np.arange(1, 20, 2) **3**	['sqrt', 'log2', **None**]
	1000-record clean	['squared_error', 'friedman_mse', 'absolute_error', **'poisson'**]	[**'best'**, 'random']	List (np.arange (5, 50, 5)) **15**	np.arange(2, 20, 2) **4**	np.arange(1, 20, 2) **5**	['sqrt', 'log2', **None**]
	1000-record noisy	[**'squared_error'**, 'friedman_mse', 'absolute_error', 'poisson']	['best', **'random'**]	List (np.arange (5, 50, 5)) **30**	np.arange(2, 20, 2) **14**	np.arange(1, 20, 2) **11**	['sqrt', 'log2', **None**]
Yield	Complete	['squared_error', **'friedman_mse'**, 'absolute_error', 'poisson']	[**'best'**, 'random']	List (np.arange (5, 50, 5)) **15**	np.arange(2, 20, 2) **12**	np.arange(1, 20, 2) **4**	['sqrt', 'log2', **None**]
	1000-record clean	[**'squared_error'**, 'friedman_mse', 'absolute_error', 'poisson']	[**'best'**, 'random']	List (np.arange (5, 50, 5)) **20**	np.arange(2, 20, 2) **6**	np.arange(1, 20, 2) **5**	['sqrt', 'log2', **None**]
	1000-record noisy	['squared_error', **'friedman_mse'**, 'absolute_error', 'poisson']	[**'best'**, 'random']	List (np.arange (5, 50, 5)) 30	np.arange(2, 20, 2) **8**	np.arange(1, 20, 2) **3**	['sqrt', 'log2', **None**]

Table A.8. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the Bagging models.

Output	Dataset	n_estimators	max_samples	max_features	bootstrap	bootstrap_features
Conversion	Complete	np.arange (10, 1000, 50) **860**	np.linspace (0.1, 1.0, 10) **0.7**	np.linspace (0.1, 1.0, 10) **1**	[**True**, False]	[True, **False**]
	1000-record clean	np.arange (10, 1000, 50) **110**	np.linspace (0.1, 1.0, 10) **0.6**	np.linspace (0.1, 1.0, 10) **1**	[**True**, False]	[True, **False**]
	1000-record noisy	np.arange (10, 1000, 50) **410**	np.linspace (0.1, 1.0, 10) **0.7**	np.linspace (0.1, 1.0, 10) **1**	[**True**, False]	[True, **False**]
Yield	Complete	np.arange (10, 1000, 50) **760**	np.linspace (0.1, 1.0, 10) **0.6**	np.linspace (0.1, 1.0, 10) **1**	[**True**, False]	[True, **False**]
	1000-record clean	np.arange (10, 1000, 50) **210**	np.linspace (0.1, 1.0, 10) **0.6**	np.linspace (0.1, 1.0, 10) **1**	[**True**, False]	[True, **False**]
	1000-record noisy	np.arange (10, 1000, 50) **560**	np.linspace (0.1, 1.0, 10) **0.7**	np.linspace (0.1, 1.0, 10) **1**	[**True**, False]	[True, **False**]

Table A.9. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the ET models.

Output	Dataset	n_estimators	max_depth	min_samples_split	min_samples_leaf	max_features	bootstrap
Conversion	Complete	np.arange (50, 1000, 50) **700**	(np.arange (5, 50, 5)) **25**	np.arange(2, 20, 2) **8**	np.arange(1, 20, 2) **1**	[**'sqrt'**, 'log2', None]	[True, **False**]
	1000-record clean	np.arange (50, 1000, 50) **850**	(np.arange (5, 50, 5)) **15**	np.arange(2, 20, 2) **8**	np.arange(1, 20, 2) **9**	['sqrt', 'log2', **None**]	[True, **False**]
	1000-record noisy	np.arange (50, 1000, 50) **950**	(np.arange (5, 50, 5)) **10**	np.arange(2, 20, 2) **18**	np.arange(1, 20, 2) **7**	['sqrt', 'log2', **None**]	[**True**, False]
Yield	Complete	np.arange (50, 1000, 50) **550**	(np.arange (5, 50, 5)) **15**	np.arange(2, 20, 2) **14**	np.arange(1, 20, 2) **5**	[**'sqrt'**, 'log2', None]	[True, **False**]
	1000-record clean	np.arange (50, 1000, 50) **700**	(np.arange (5, 50, 5)) **25**	np.arange(2, 20, 2) **8**	np.arange(1, 20, 2) **7**	[**'sqrt'**, 'log2', None]	[True, **False**]
	1000-record noisy	np.arange (50, 1000, 50) **750**	(np.arange (5, 50, 5)) **15**	np.arange(2, 20, 2) **12**	np.arange(1, 20, 2) **3**	['sqrt', 'log2', **None**]	[**True**, False]

Table A.10. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the RF models.

Output	Dataset	n_estimators	max_depth	min_samples_split	min_samples_leaf	max_features	bootstrap
Conversion	Complete	np.arange (50, 1000, 50) **600**	(np.arange (5, 50, 5)) **25**	np.arange(2, 20, 2) **8**	np.arange(1, 20, 2) **1**	[**'sqrt'**, 'log2', None]	[True, **False**]
	1000-record clean	np.arange (50, 1000, 50) **950**	(np.arange (5, 50, 5)) **10**	np.arange(2, 20, 2) **18**	np.arange(1, 20, 2) **7**	['sqrt', 'log2', **None**]	[**True**, False]
	1000-record noisy	np.arange (50, 1000, 50) **950**	(np.arange (5, 50, 5)) **10**	np.arange(2, 20, 2) **18**	np.arange(1, 20, 2) **7**	['sqrt', 'log2', **None**]	[**True**, False]
Yield	Complete	np.arange (50, 1000, 50) **700**	(np.arange (5, 50, 5)) **15**	np.arange(2, 20, 2) **6**	np.arange(1, 20, 2) **5**	[**'sqrt'**, 'log2', None]	[True, **False**]
	1000-record clean	np.arange (50, 1000, 50) **500**	(np.arange (5, 50, 5)) **15**	np.arange(2, 20, 2) **8**	np.arange(1, 20, 2) **1**	[**'sqrt'**, 'log2', None]	[True, **False**]
	1000-record noisy	np.arange (50, 1000, 50) **850**	(np.arange (5, 50, 5)) **15**	np.arange(2, 20, 2) **4**	np.arange(1, 20, 2) **3**	['sqrt', 'log2', **None**]	[**True**, False]

Table A.11. List of hyperparameters explored and the selected optimal values for each output variable and dataset in the XGBoost models.

Output	Dataset	n_estimators	Max depth	Learning rate	Min child weight	subsample	gamma	lambda	alpha
Conversion	Complete	np.arange (50, 1000, 50) **650**	np.arange (3, 20, 2) **9**	np.logspace (-3, 0, 10) **0.0999**	np.arange(1, 10, 1) **4**	np.linspace(0.5, 1, 5) **0.875**	np.logspace (-3, 1, 10) **0.0027**	np.logspace (-3, 3, 10) **215.443**	np.logspac e(-3, 3, 10) **0.0046**
	1000-record clean	np.arange (50, 1000, 50) **250**	np.arange (3, 20, 2) **19**	np.logspace (-3, 0, 10) **0.0464**	np.arange(1, 10, 1) **8**	np.linspace(0.5, 1, 5) **0.625**	np.logspace (-3, 1, 10) **0.001**	np.logspace (-3, 3, 10) **10**	np.logspac e(-3, 3, 10) **0.0999**
	1000-record noisy	np.arange (50, 1000, 50) **700**	np.arange (3, 20, 2) **11**	np.logspace (-3, 0, 10) **0.0999**	np.arange(1, 10, 1) **1**	np.linspace(0.5, 1, 5) **0.75**	np.logspace (-3, 1, 10) **0.0027**	np.logspace (-3, 3, 10) **215.443**	np.logspac e(-3, 3, 10) **0.001**
Yield	Complete	np.arange (50, 1000, 50) **500**	np.arange (3, 20, 2) **7**	np.logspace (-3, 0, 10) **0.0215**	np.arange(1, 10, 1) **4**	np.linspace(0.5, 1, 5) **0.625**	np.logspace (-3, 1, 10) **0.0027**	np.logspace (-3, 3, 10) **10**	np.logspac e(-3, 3, 10) **0.0046**
	1000-record clean	np.arange (50, 1000, 50) **250**	np.arange (3, 20, 2) **19**	np.logspace (-3, 0, 10) **0.0464**	np.arange(1, 10, 1) **8**	np.linspace(0.5, 1, 5) **0.625**	np.logspace (-3, 1, 10) **0.001**	np.logspace (-3, 3, 10) **10**	np.logspac e(-3, 3, 10) **0.0999**
	1000-record noisy	np.arange (50, 1000, 50) **350**	np.arange (3, 20, 2) **9**	np.logspace (-3, 0, 10) **0.0464**	np.arange(1, 10, 1) **6**	np.linspace(0.5, 1, 5) **0.75**	np.logspace (-3, 1, 10) **0.0027**	np.logspace (-3, 3, 10) **215.443**	np.logspac e(-3, 3, 10) **0.001**

3. Results

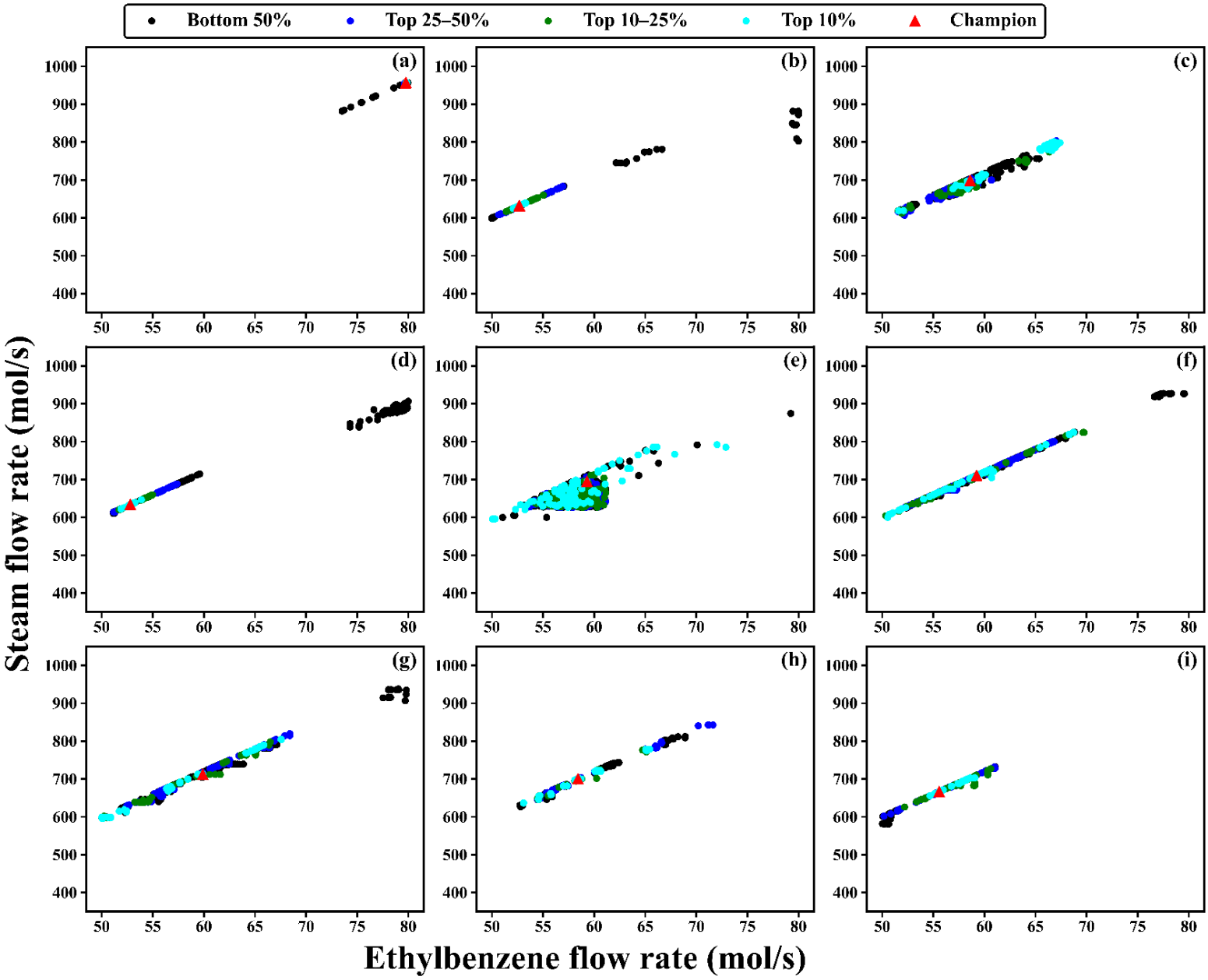


Figure A.8. Decision space plots (steam vs ethylbenzene flow rates) for all models for the complete dataset. Subplots correspond to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost.

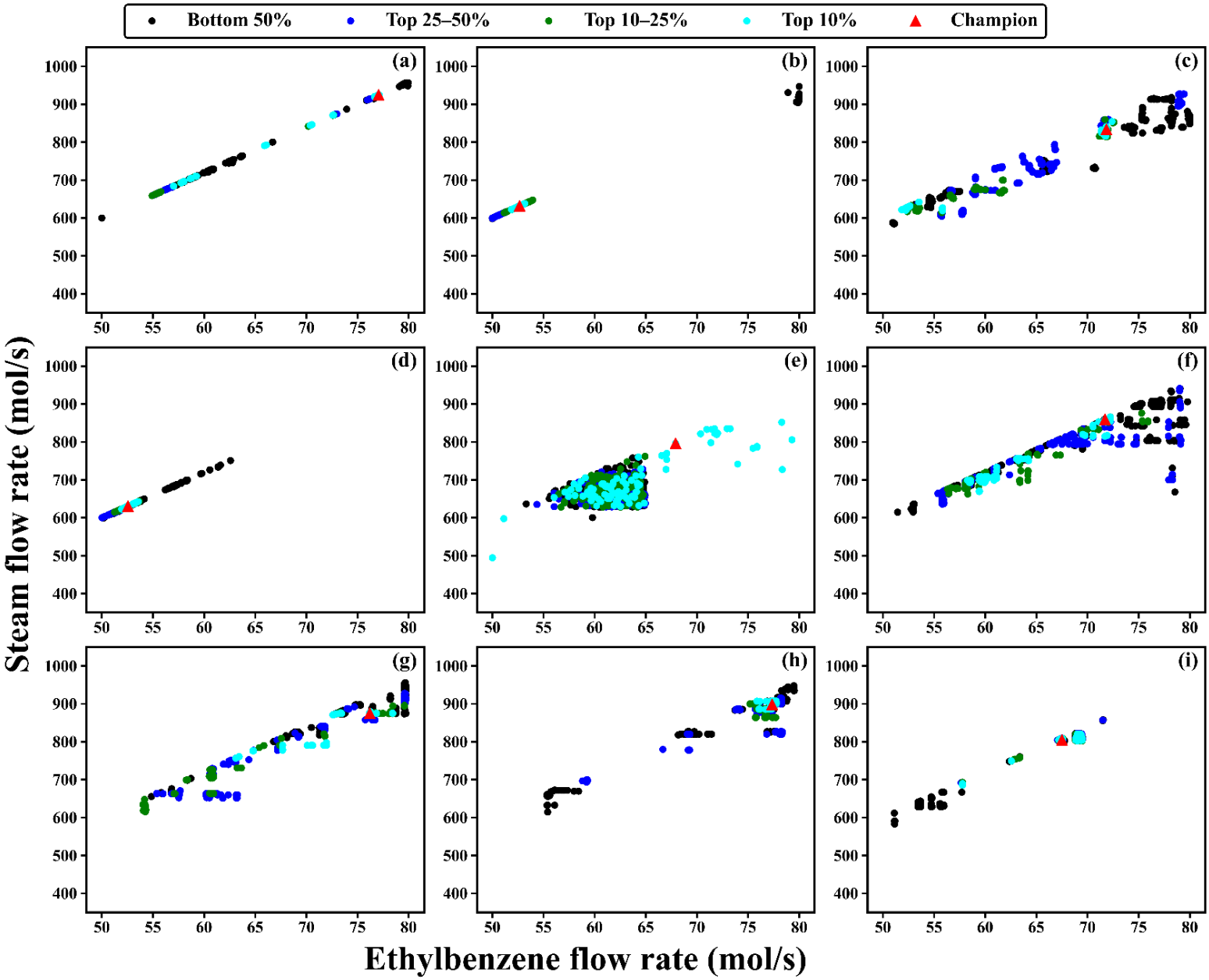


Figure A.9. Decision space plots (steam vs ethylbenzene flow rates) for all models for the 1000-record clean dataset. Subplots correspond to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost.

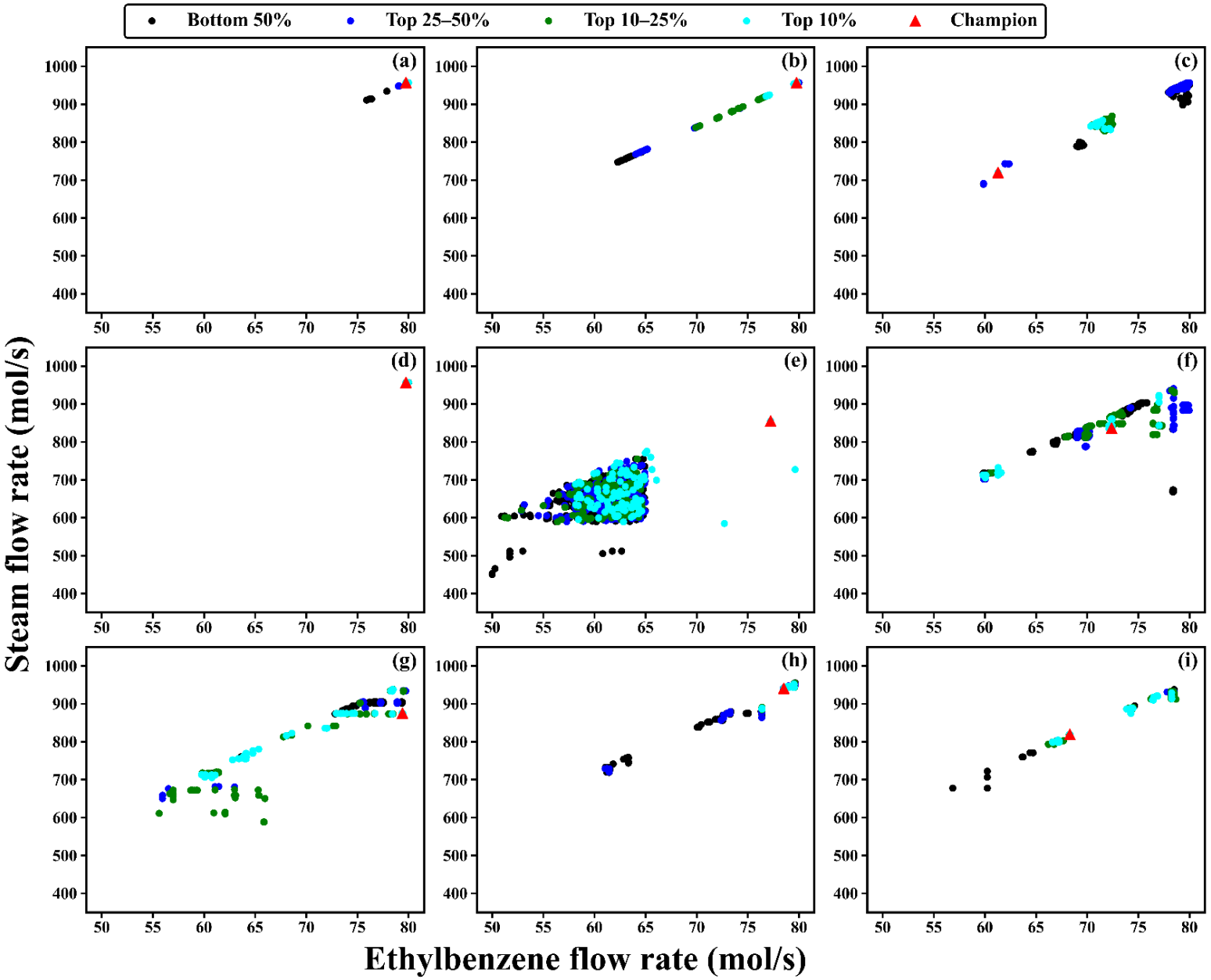


Figure A.10. Decision space plots (steam vs ethylbenzene flow rates) for all models for the 1000-record noisy dataset. Subplots correspond to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost.

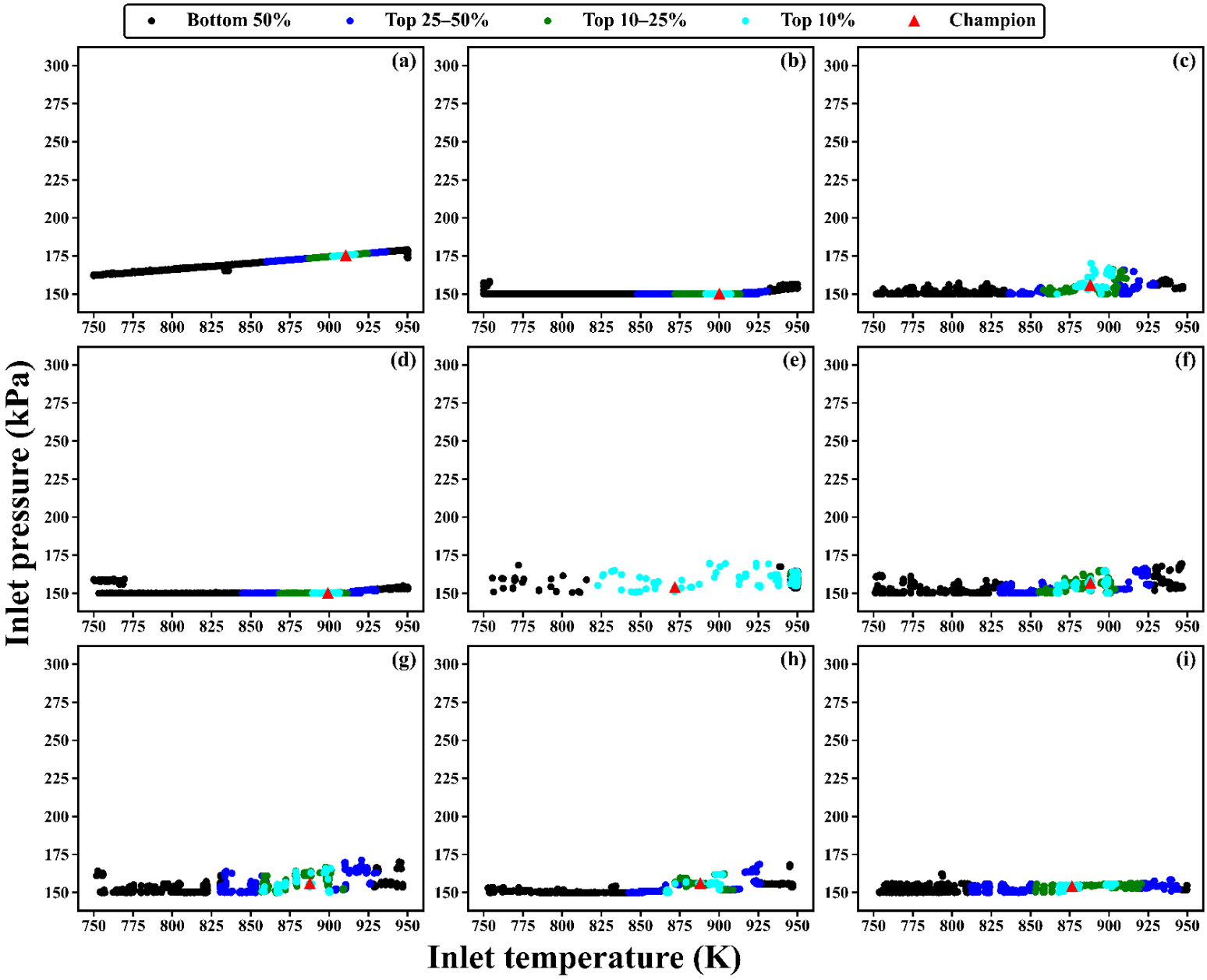


Figure A.11. Decision space plots (inlet pressure vs inlet temperature) for all models for the complete dataset. Subplots correspond to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost.

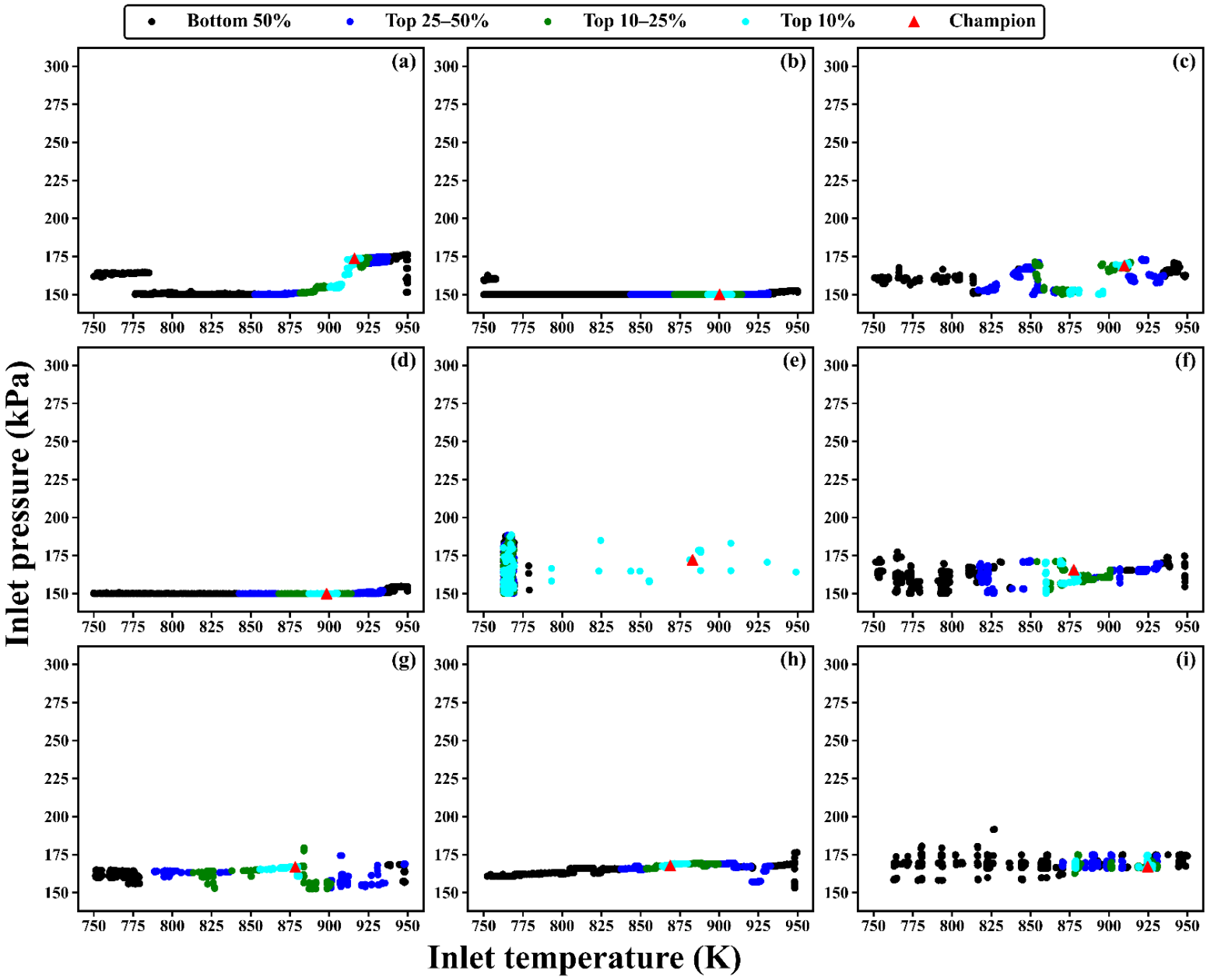


Figure A.12. Decision space plots (inlet pressure vs inlet temperature) for all models for the 1000-record clean dataset. Subplots correspond to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost.

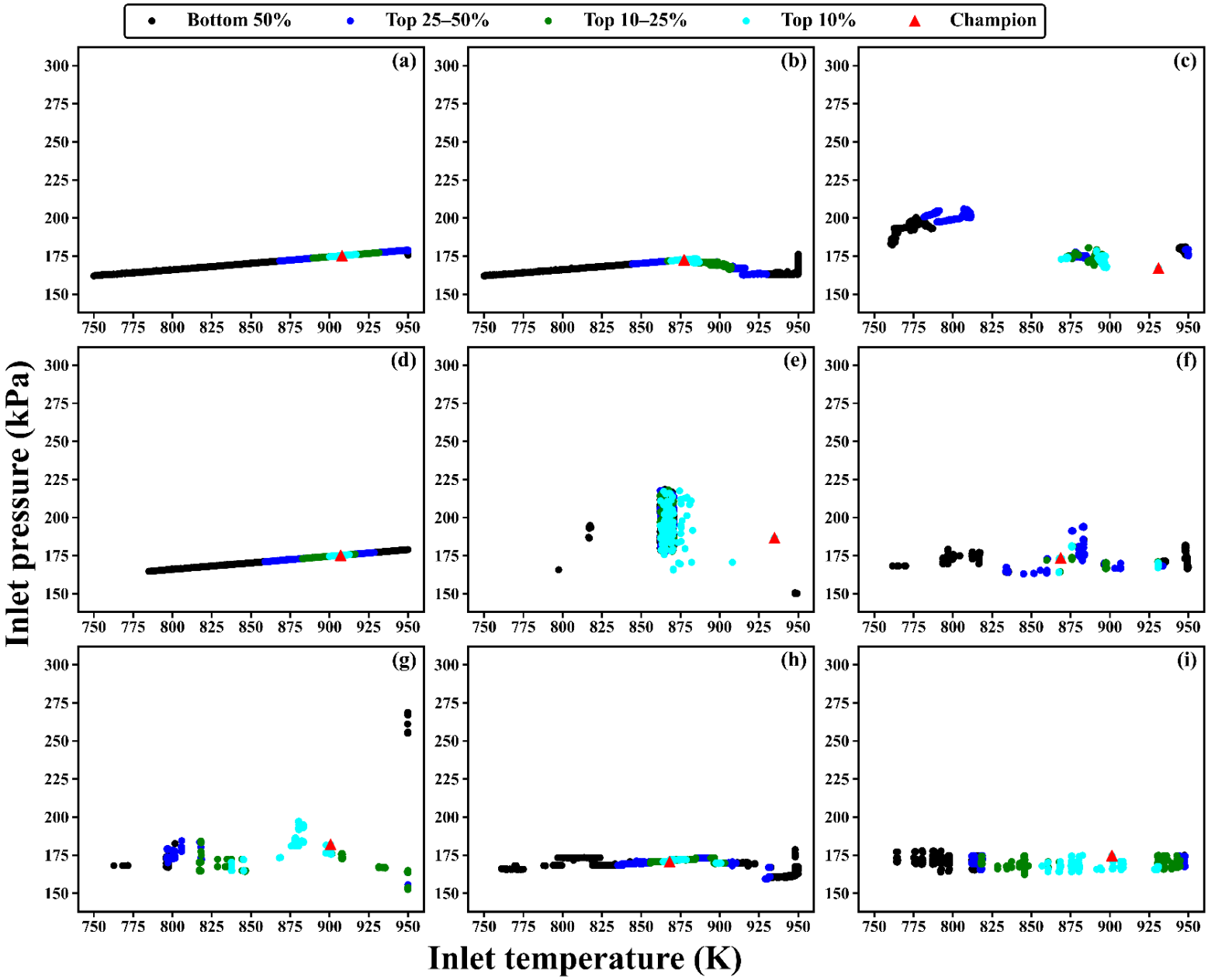


Figure A.13. Decision space plots (inlet pressure vs inlet temperature) for all models for the 1000-record noisy dataset. Subplots correspond to (a) ANN, (b) GPR, (c) KNN, (d) SVR, (e) DT, (f) Bagging, (g) RF, (h) ET, and (i) XGBoost.

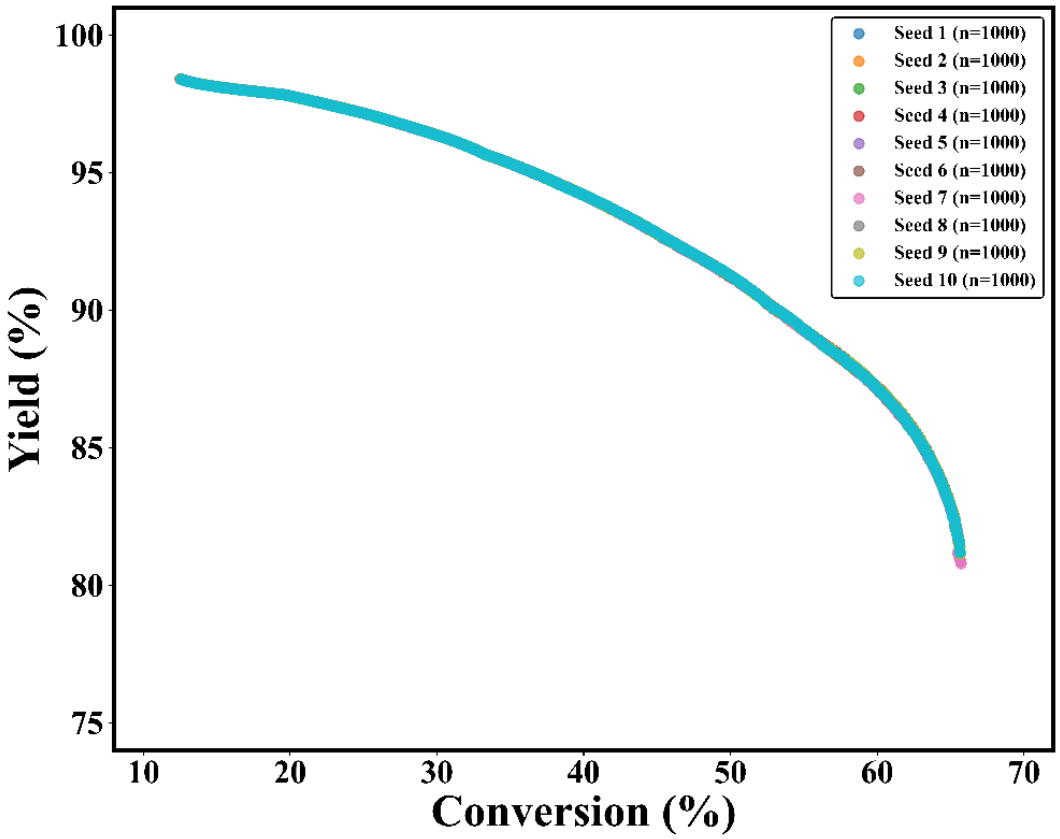


Figure A.14. Pareto fronts generated from 10 repeated NSGA-II runs using different random seeds for the ANN model on the 1000-record clean dataset. The almost indistinguishable fronts demonstrate that the optimization results are highly stable with respect to seed selection under the adopted optimization settings.

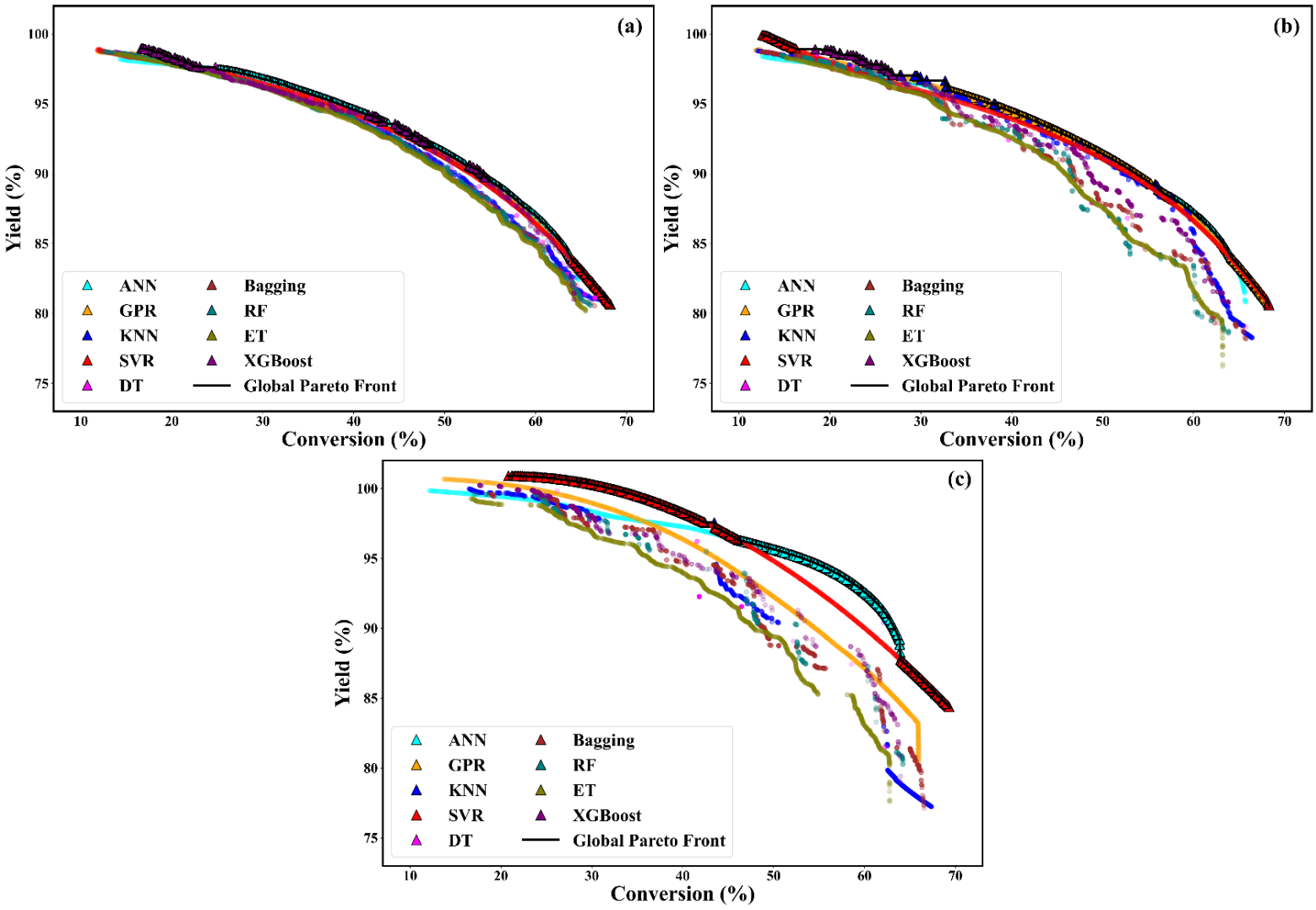


Figure A.15. Full overlay of the Pareto fronts generated by all nine ML models together with the corresponding global Pareto front for the three data scenarios: (a) complete dataset, (b) 1000-record clean dataset, and (c) 1000-record noisy dataset. The black curve represents the global Pareto front obtained from the pooled non-dominated solutions, allowing direct visual comparison of the deviation of each model-specific front from the overall best trade-off surface.

Table A.12. Comparison of champion points (obtained by NFM) of each model for both objectives and decision space for all three datasets.

Dataset	ML model	Ethylbenzene flow rate (mol/s)	Steam flow rate (mol/s)	Inlet temperature (K)	Inlet pressure (kPa)	Conversion (%) (ML model)	Conversion (%) (simulation)	Yield (%) (ML model)	Yield (%) (simulation)
Complete dataset	ANN	79.75	957.00	910.56	175.47	55.82	54.30	89.14	88.10
	GPR	52.68	632.05	900.22	150.00	54.85	54.70	89.27	89.10
	KNN	58.61	698.94	888.04	155.99	50.42	50.40	90.58	90.60
	SVR	52.80	633.51	899.12	150.02	54.53	54.40	89.27	89.30
	DT	59.30	694.66	871.86	153.95	46.14	45.40	92.65	92.30
	Bagging	59.23	710.63	888.21	156.68	50.36	50.40	90.34	90.60
	RF	59.89	712.41	887.65	155.96	50.31	50.00	90.44	90.70
	ET	58.42	700.62	887.98	155.98	49.81	50.50	90.25	90.60
	XGBoost	55.57	666.45	876.37	154.01	47.12	47.50	92.63	91.70
1000-record clean dataset	ANN	77.06	924.72	916.07	173.71	57.99	56.30	88.16	87.30
	GPR	52.67	631.94	900.30	150.00	55.08	54.80	89.39	89.10
	KNN	71.84	833.46	909.80	168.89	55.98	54.90	89.15	87.80
	SVR	52.55	630.54	898.32	150.00	54.57	54.20	89.33	89.40
	DT	67.91	796.14	883.07	172.22	47.25	48.00	91.77	90.40
	Bagging	71.71	859.35	877.61	165.68	46.30	45.40	90.95	92.00
	RF	76.27	873.64	878.11	166.77	45.72	44.80	91.50	91.90
	ET	77.35	898.71	868.90	167.74	42.22	41.90	91.86	92.90
	XGBoost	67.50	804.73	924.62	167.10	58.10	59.80	86.57	85.60
1000-record noisy dataset	ANN	79.75	957.00	907.97	175.25	56.42	53.60	94.11	88.50
	GPR	79.75	956.99	877.37	172.56	44.88	44.30	94.54	92.10
	KNN	61.28	719.91	930.98	167.07	62.09	62.00	90.39	83.40
	SVR	79.75	957.00	907.08	175.17	56.56	53.30	91.83	88.60
	DT	77.22	855.22	934.86	186.67	58.55	60.90	87.43	81.40
	Bagging	72.37	836.14	868.54	173.35	43.77	43.10	94.44	92.10
	RF	79.40	874.96	900.73	181.98	52.42	51.40	90.30	87.80
	ET	78.52	940.98	868.15	170.72	41.88	41.70	93.41	93.00
	XGBoost	68.30	819.51	901.26	174.74	53.37	53.40	90.54	88.20

Table A.13. Local inter-model uncertainty around the majority and minority consensus points for the complete, 1000-record clean, and 1000-record noisy datasets. For each consensus point, the nearest Pareto-optimal solution from each ML model was identified in normalized objective space using conversion and yield, and the resulting dispersion across models was summarized by the mean, standard deviation, and range.

Dataset	Consensus point	Conversion mean ± SD (%)	Conversion range (%)	Yield mean ± SD (%)	Yield range (%)	Ethylbenzene flow rate mean ± SD (mol/s)	Steam flow rate mean ± SD (mol/s)	Inlet temperature mean ± SD (K)	Inlet pressure mean ± SD (kPa)
Complete dataset	Majority point	50.62 ± 0.64	49.43–51.43	90.63 ± 0.22	90.38–90.95	60.07 ± 7.84	718.53 ± 94.13	888.98 ± 3.24	156.59 ± 7.07
	Minority point	54.58 ± 1.03	52.75–55.82	88.81 ± 0.31	88.29–89.14	57.70 ± 9.58	686.47 ± 115.30	901.79 ± 5.35	155.97 ± 8.69
1000-record clean dataset	Majority point	49.58 ± 2.09	46.65–51.97	89.76 ± 0.93	87.93–90.67	65.11 ± 9.61	758.56 ± 96.05	889.20 ± 5.45	162.76 ± 10.62
	Minority point	53.02 ± 1.83	49.97–55.00	88.33 ± 1.09	86.82–89.43	62.69 ± 8.50	729.29 ± 89.13	901.57 ± 3.61	160.37 ± 7.84
1000-record noisy dataset	Majority point	50.74 ± 1.66	47.86–52.38	91.60 ± 1.96	89.07–95.17	74.55 ± 6.90	858.17 ± 93.70	894.69 ± 8.25	173.02 ± 5.53
	Minority point	60.37 ± 1.52	58.55–62.60	88.17 ± 1.76	85.18–90.76	68.23 ± 8.31	802.91 ± 99.12	930.60 ± 5.34	170.25 ± 8.37

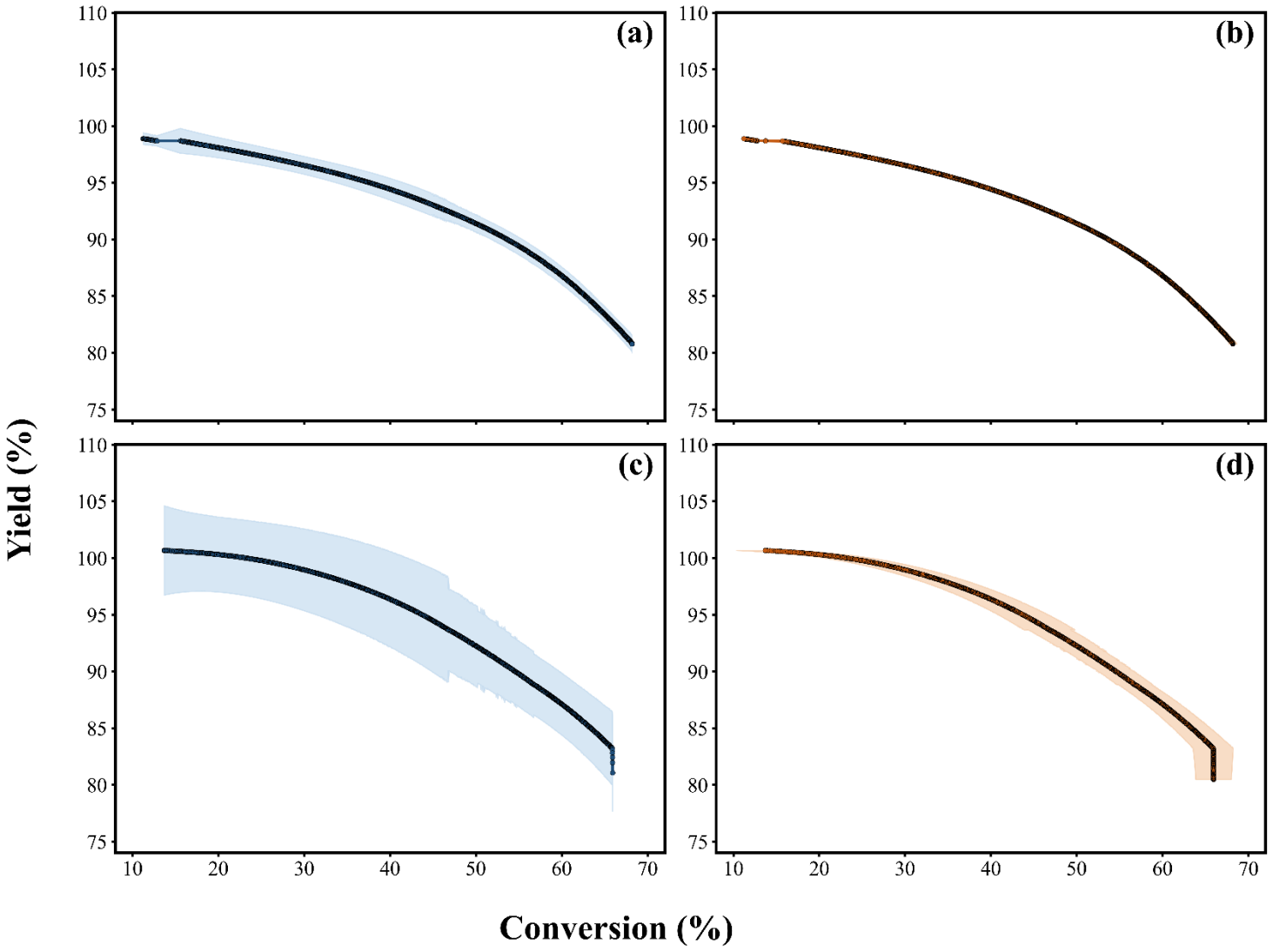


Figure A.16. Predictive uncertainty bands for the GPR-based Pareto fronts. Panels (a) and (b) correspond to the 1000-record clean dataset, and panels (c) and (d) correspond to the 1000-record noisy dataset. Panels (a) and (c) show the 95% confidence bands for yield, while panels (b) and (d) show the 95% confidence bands for conversion, all plotted in the standard Pareto orientation of yield versus conversion.

4. References

1. Saha, S., & Kundu, B. (2025). Highest electro-viscous energy and lowest irreversibility analysis for Maxwell fluid in transient microchannel flow. *Applied Thermal Engineering*, *267*, 125764. https://doi.org/10.1016/j.applthermaleng.2025.125764

2. Kanaani, M., Sedaghat Kameholiya, A., Amarzadeh, A., & Sedaee, B. (2025). Stacking Learning for Smart Proxy Modeling in CO_2 –WAG Optimization: A Techno-Economic Approach to Sustainable Enhanced Oil Recovery. *ACS Omega*, *10*(9), 9563–9582. https://doi.org/10.1021/acsomega.4c10432
3. Hsu, C.-Y., Pallathadka, H., Patel, P., Yogi, K. S., Bokov, D. O., Rao, S. B., Kumar, A. V., Verma, A., Mamory, T. R. A., & Faaq, A. S. (2025). A novel approach for optimizing a photovoltaic thermal system combined with solar thermal collector: Integrating RSM, multi-objective bat algorithm and VIKOR decision maker. *Journal of the Taiwan Institute of Chemical Engineers*, *168*, 105927. https://doi.org/10.1016/j.jtice.2024.105927
4. Wang, L., Alirahmi, S. M., & Rejeb, O. (2025). Solar-Enhanced ocean thermal energy conversion for green liquid hydrogen production. *Applied Thermal Engineering*, *265*, 125615. https://doi.org/10.1016/j.applthermaleng.2025.125615
5. Li, Y., Basem, A., Alizadeh, A., Singh, P. K., Dixit, S., Abdulaali, H. K., Ali, R., Cajla, P., Rajab, H., & Ghachem, K. (2025). Synergizing neural networks with multi-objective thermal exchange optimization and PROMETHEE decision-making to improve PCM-based photovoltaic thermal systems. *Case Studies in Thermal Engineering*, *68*, 105851. https://doi.org/10.1016/j.csite.2025.105851
6. Diao, X., Feng, H., Chen, L., Ge, Y., & Wu, B. (2025). Constructal design of a T-shaped porous fin adopting minimum composite function, ANN and NSGA-II. *International Communications in Heat and Mass Transfer*, *162*, 108581. https://doi.org/10.1016/j.icheatmasstransfer.2025.108581
7. Wu, B., Feng, H., Chen, L., Ge, Y., & Liu, X. (2024). Constructal design of a hybrid heat sink with rectangular microchannel and porous fin in a 3D electronic device with artificial neural-network, NSGA-II and different decision-making methods. *International Communications in Heat and Mass Transfer*, *158*, 107954. https://doi.org/10.1016/j.icheatmasstransfer.2024.107954
8. Ghorbani, B., Zendehboudi, S., Alizadeh Afrouzi, Z., Lohi, A., & Khan, F. (2024). Efficient Design of the Hydrogen Liquefaction System: Thermodynamic, Economic, Environmental, and Uncertainty Perspectives. *Industrial & Engineering Chemistry Research*, *63*(33), 14668–14699. https://doi.org/10.1021/acs.iecr.3c04286

9. Rebello, C. M., & Nogueira, I. B. R. (2024). Optimizing CO_2 capture in pressure swing adsorption units: A deep neural network approach with optimality evaluation and operating maps for decision-making. *Separation and Purification Technology*, *340*. https://doi.org/10.1016/j.seppur.2024.126811
10. Joo, C., Kwon, H., Lim, J., Lee, J., & Kim, J. (2024). Multiobjective Optimization of CO_2 Emission and Net Profit for a Naphtha Cracking Furnace Using a Deep Neural Network with a Nondominated Sorting Genetic Algorithm. *ACS Sustainable Chemistry & Engineering*, *12*(7), 2841–2851. https://doi.org/10.1021/acssuschemeng.3c07939
11. Zhang, T., Manafi Khajeh Pasha, A., Mohammad Sajadi, S., Jasim, D. J., Nasajpour-Esfahani, N., Maleki, H., Salahshour, S., & Baghaei, Sh. (2024). Optimization of thermophysical properties of nanofluids using a hybrid procedure based on machine learning, multi-objective optimization, and multi-criteria decision-making. *Chemical Engineering Journal*, *485*, 150059. https://doi.org/10.1016/j.cej.2024.150059
12. Lu, Z., Li, Y., He, C., Ren, J., Yu, H., Zhang, B., & Chen, Q. (2024). Multi-objective inverse design of finned heat sink system with physics-informed neural networks. *Computers & Chemical Engineering*, *180*, 108500. https://doi.org/10.1016/j.compchemeng.2023.108500
13. Wolday, A. K., & Ramteke, M. (2024). Surrogate model-based optimization of methanol synthesis process for multiple objectives: A pathway towards achieving sustainable development goals. *Chemical Engineering Research and Design*, *204*, 172–182. https://doi.org/10.1016/j.cherd.2024.02.021
14. Abdollahi, S. A., Alenezi, A., Alizadeh, A., Jasim, D. J., Ahmed, M., Fezaa, L. H. A., Aich, W., Said, L. B., Kolsi, L., & Maleki, H. (2024). A novel insight into the design of perforated-finned heat sinks based on a hybrid procedure: Computational fluid dynamics, machine learning, multi-objective optimization, and multi-criteria decision-making. *International Communications in Heat and Mass Transfer*, *155*, 107535. https://doi.org/10.1016/j.icheatmasstransfer.2024.107535
15. Li, Z., Gao, Y., Zhu, Z., & Tian, W. (2024). Data-guided for discovering high-strength, cost-effective, and low-carbon rice husk ash concrete. *Journal of CO_2 Utilization*, *83*, 102786. https://doi.org/10.1016/j.jcou.2024.102786

16. Wu, Y., Li, Z., Zhang, B., Chen, H., & Sun, Y. (2024). Multi-objective optimization of key parameters of stirred tank based on ANN-CFD. *Powder Technology*, *441*, 119832. https://doi.org/10.1016/j.powtec.2024.119832
17. Nabavi, S. R., Ghahri, S., & Rangaiah, G. P. (2024). Data-Based Modeling, Multi-Objective Optimization and Multi-Criteria Decision Making of a Catalytic Ozonation Process for Degradation of a Colored Effluent. *Processes*, *12*(3), 515. https://doi.org/10.3390/pr12030515
18. Zhao, X., Fan, H., Lin, G., Fang, Z., Yang, W., Li, M., Wang, J., Lu, X., Li, B., Wu, K., & Fu, J. (2024). Multi-objective optimization of radially stirred tank based on CFD and machine learning. *AIChE Journal*, *70*(3), e18324. https://doi.org/10.1002/aic.18324
19. Pasic, M., Marinkovic, D., Lukic, D., Begic-Hajdarevic, D., Zivkovic, A., Milosevic, M., & Muhamedagic, K. (2024). Prediction and Optimization of Surface Roughness and Cutting Forces in Turning Process Using ANN, SHAP Analysis, and Hybrid MCDM Method. *Applied Sciences*, *14*(23), 11386. https://doi.org/10.3390/app142311386
20. Chen, R., Qian, H., Khashan, M. K., Abdulameer, K. I., Abed Balla, H. H., Kareem, Z. S., Ghandour, R., & Fouad, Y. (2024). Triple-objective optimization using ANN+NSGA-II for an innovative biomass gasification-heat recovery process, producing electricity, coolant, and liquefied hydrogen. *Case Studies in Thermal Engineering*, *60*. https://doi.org/10.1016/j.csite.2024.104647
21. Cao, H., Li, Y., Chang, C., Zhang, X., Yang, A., & Shen, W. (2024). A Deep Learning Hybrid Framework Combining an Efficient Evolutionary Algorithm for Complex Many-Objective Optimization of Sustainable Triple CO_2 Feed Methanol Production. *ACS Sustainable Chemistry & Engineering*, *12*(17), 6682–6696. https://doi.org/10.1021/acssuschemeng.4c00448
22. Nutakki, T. U. K., Alghassab, M. A., Dutta, A. K., Abdullaeva, B. S., Alkhalaf, S., Alharbi, F. S., Ghandour, R., Al Barakeh, Z., & Knani, S. (2024). Thermo-environmental multi-investigation and ANN-based optimization of a novel heat integration criteria system integrated with a marine engine generating liquefied hydrogen. *Case Studies in Thermal Engineering*, *56*, 104240. https://doi.org/10.1016/j.csite.2024.104240
23. Fini, A. T., Fattahi, A., & Musavi, S. (2023). Machine learning prediction and multiobjective optimization for cooling enhancement of a plate battery using the chaotic water-

microencapsulated PCM fluid flows. *Journal of the Taiwan Institute of Chemical Engineers*, *148*. https://doi.org/10.1016/j.jtice.2023.104680

24. Cui, H. C., Shi, C. Y., Yu, M. J., Zhang, Z. K., Liu, Z. C., & Liu, W. (2023). Optimal parameter design of a slot jet impingement/microchannel heat sink base on multi-objective optimization algorithm. *Applied Thermal Engineering*, *227*, 120452. https://doi.org/10.1016/j.applthermaleng.2023.120452
25. Wang, Z., Tan, W. G. Y., Rangaiah, G. P., & Wu, Z. (2023). Machine learning aided model predictive control with multi-objective optimization and multi-criteria decision making. *Computers & Chemical Engineering*, *179*, 108414. https://doi.org/10.1016/j.compchemeng.2023.108414
26. Nabavi, S. R., Jafari, M. J., & Wang, Z. (2023). Deep learning aided multi-objective optimization and multi-criteria decision making in thermal cracking process for olefines production. *Journal of the Taiwan Institute of Chemical Engineers*, *152*, 105179. https://doi.org/10.1016/j.jtice.2023.105179
27. Allen, L., Gill, A., Smith, A., Hill, D., Moghadam, P. Z., & Cordiner, J. (2023). Development of a machine learning framework to determine optimal alloy composition based on steel hardenability prediction. *Digital Chemical Engineering*, *9*, 100118. https://doi.org/10.1016/j.dche.2023.100118
28. Dat Nguyen, V., Chang, J., Hong, S.-H., & Lee, C.-H. (2023). Performance and ANN-based optimization of an advanced process for wet CO_2-to-Methanol using a catalytic fluidized bed reactor integrated with separators. *Fuel*, *343*. https://doi.org/10.1016/j.fuel.2023.128045
29. Shakibi, H., Shokri, A., Assareh, E., Yari, M., & Lee, M. (2023). Using machine learning approaches to model and optimize a combined solar/natural gas-based power and freshwater cogeneration system. *Applied Energy*, *333*, 120607.
30. Wang, Z., Li, J., Rangaiah, G. P., & Wu, Z. (2022). Machine learning aided multi-objective optimization and multi-criteria decision making: Framework and two applications in chemical engineering. *Computers & Chemical Engineering*, *165*, 107945. https://doi.org/10.1016/j.compchemeng.2022.107945
31. Shirmohammadi, R., Aslani, A., Ghasempour, R., Romeo, L. M., & Petrakopoulou, F. (2022). Exergoenvironmental analysis and thermoeconomic optimization of an industrial post-

combustion CO2 capture and utilization installation. *Journal of CO_2 Utilization*, *59*, 101927. https://doi.org/10.1016/j.jcou.2022.101927

32. Talebjedi, B., Heydari, M., Taatizadeh, E., Tasnim, N., Li, I. T. S., & Hoorfar, M. (2022). Neural Network-Based Optimization of an Acousto Microfluidic System for Submicron Bioparticle Separation. *Frontiers in Bioengineering and Biotechnology*, *10*, 878398. https://doi.org/10.3389/fbioe.2022.878398
33. Nguyen, V. T., Ta, Q. T. H., & Nguyen, P. K. T. (2022). Artificial intelligence-based modeling and optimization of microbial electrolysis cell-assisted anaerobic digestion fed with alkaline-pretreated waste-activated sludge. *Biochemical Engineering Journal*, *187*, 108670. https://doi.org/10.1016/j.bej.2022.108670
34. Gunnell, L. L., Manwaring, K., Lu, X., Reynolds, J., Vienna, J., & Hedengren, J. (2022). Machine Learning with Gradient-Based Optimization of Nuclear Waste Vitrification with Uncertainties and Constraints. *Processes*, *10*(11), 2365. https://doi.org/10.3390/pr10112365
35. Vural, N., Yilmazer Hitit, Z., & Ertunç, S. (2021). Multi-objective optimization of drying conditions for the *Olea europaea* L. leaves with NSGA-II. *Journal of Food Processing and Preservation*, *45*(7), e15625. https://doi.org/10.1111/jfpp.15625
36. Abbasi, H. R., Sharifi Sedeh, E., Pourrahmani, H., & Mohammadi, M. H. (2020). Shape optimization of segmental porous baffles for enhanced thermo-hydraulic performance of shell-and-tube heat exchanger. *Applied Thermal Engineering*, *180*, 115835. https://doi.org/10.1016/j.applthermaleng.2020.115835
37. De Medeiros, E. M., Noorman, H., Maciel Filho, R., & Posada, J. A. (2020). Production of ethanol fuel via syngas fermentation: Optimization of economic performance and energy efficiency. *Chemical Engineering Science: X*, *5*, 100056. https://doi.org/10.1016/j.cesx.2020.100056
38. Rall, D., Schweidtmann, A. M., Kruse, M., Evdochenko, E., Mitsos, A., & Wessling, M. (2020). Multi-scale membrane process optimization with high-fidelity ion transport models through machine learning. *Journal of Membrane Science*, *608*, 118208. https://doi.org/10.1016/j.memsci.2020.118208
39. Chen, Q., Hu, P., Pu, J., & Wang, J. H. (2019). Sensitivity analysis and multi-objective optimization of double-ceramic-layers thermal barrier system. *Ceramics International*, *45*(14), 17224–17235. https://doi.org/10.1016/j.ceramint.2019.05.278

40. Del Rio-Chanona, E. A., Wagner, J. L., Ali, H., Fiorelli, F., Zhang, D., & Hellgardt, K. (2019). Deep learning-based surrogate modeling and optimization for microalgal biofuel production and photobioreactor design. *AIChE Journal*, *65*(3), 915–923. https://doi.org/10.1002/aic.16473
41. Kojić, J. S., Ilić, N. M., Kojić, P. S., Pezo, L. L., Banjac, V. V., Krulj, J. A., & Bodroža Solarov, M. I. (2019). Multiobjective process optimization for betaine enriched spelt flour based extrudates. *Journal of Food Process Engineering*, *42*(1), e12942. https://doi.org/10.1111/jfpe.12942
42. Amani, M., Amani, P., Jumpholkul, C., Mahian, O., & Wongwises, S. (2018). Hydrothermal optimization of SiO2/water nanofluids based on attitudes in decision making. *International Communications in Heat and Mass Transfer*, *90*, 67–72. https://doi.org/10.1016/j.icheatmasstransfer.2017.10.008
43. Saeedan, M., & Bahiraei, M. (2015). Effects of geometrical parameters on hydrothermal characteristics of shell-and-tube heat exchanger with helical baffles: Numerical investigation, modeling and optimization. *Chemical Engineering Research and Design*, *96*, 43–53. https://doi.org/10.1016/j.cherd.2015.02.004
44. Bahiraei, M., & Hangi, M. (2016). Numerical investigation and optimization of flow and thermal characteristics of nanofluid within a chaotic geometry. *Advanced Powder Technology*, *27*(1), 184–192. https://doi.org/10.1016/j.apt.2015.11.016

Appendix B: Supporting Information of Chapter 6

Machine Learning–Driven Design of Catalytic Processes for Sulfur Dioxide Oxidation: Lessons from the Trenches

1. Introduction

Dehydrogenation reaction: One of the pioneering studies in the application of ML to catalysis was conducted by Kite et al. They explored the utilization of artificial neural networks (ANNs) to predict the selectivity of various products of the dehydrogenation reaction of ethylbenzene, including styrene, benzaldehyde, benzene, toluene, CO, and CO_2. The input dataset comprised 18 SnO catalysts, with and without a promoter, and included variables such as surface area, valence, catalyst composition, ionic radius, coordination number, electronegativity, partial charge of oxygen ions, and the standard heat of formation of oxides [1].

CO oxidation reaction: Günay and Yildirim compiled a database of 1 337 samples to develop an ANN model aimed at predicting CO conversion over Cu-based catalysts. The variables considered in their study encompassed catalyst preparation parameters (including Cu loading, secondary metal additives, support type, and preparation method) and operating parameters (such as reaction temperature, feed composition, and feed flow rate to catalyst weight ratio). Additionally, they examined the impact of these variables using an ANN model [2]. In another study, Günay and Yildirim assembled a comprehensive dataset consisting of 5 610 samples derived from 80 scholarly articles, focusing on the catalytic oxidation of CO over noble metals. This extensive database encompassed six noble metals, nine preparation methods, fifteen supports, and twenty promoters. The operating variables included reaction temperature, concentrations of H_2, O_2, CO, H_2O, and CO_2 in the feed stream, time on stream, and the feed flow rate to catalyst mass ratio. These variables were collectively categorized as input variables, with the output variable being CO conversion. Initially, they employed a decision tree methodology to classify the conditions that resulted in high CO conversion. Subsequently, they utilized an ANN to determine the relative significance of each input variable [3]. Günay and Yildirim also explored the feasibility of predicting the kinetics of CO oxidation using Au-based catalysts. To this end, they compiled a dataset comprised of 882 experimental records from 19 different papers. They developed and tested an ANN model that used catalyst properties and operating variables as inputs to predict the reaction rate [4].

Water-gas shift (WGS) reaction: Cavalcanti et al. collected 283 experimental records from the literature to construct an ANN for predicting CO conversion in the WGS reaction. The study utilized inputs encompassing operating conditions (such as temperature, pressure, catalyst mass,

gas hourly space velocity (GHSV), and feed composition) alongside catalyst design and textural properties (including active phase and its composition, support type, promoter/dopant concentration, surface area, and calcination temperature and time) [5]. Odabaşı et al. compiled 4 360 experimental records from 84 publications on the WGS reaction over Pt- and Au-based catalysts. The dataset included catalyst preparation variables such as base metal type and weight percentage, preparation method, support type, promoter type and weight percentage, and calcination temperature and time. Additionally, operating variables comprised the concentrations of reactants in the feed stream, along with the reaction temperature, time on stream, and catalyst weight/feed flow rate. CO conversion was selected as the output variable. Decision trees were used to identify heuristics leading to high conversion. ANNs determined the significance and impact of operating variables, while support vector machines (SVMs) predicted outcomes under previously untested experimental conditions [6]. Kim and Kim developed an ANN to predict CO conversion in the WGS reaction over $Pt/Ce_xZr_{1-x}O_2$ catalysts. They gathered data from 18 publications, considering various designs and operating conditions such as material composition and content, operating conditions, calcination conditions, impregnation methods, feed composition, and reaction performance [7].

Oxidative coupling of methane (OCM) reaction: Mine et al. prepared an updated dataset consisting of 4 759 experimental records for the OCM reaction, based on literature data reported before 2020. They tested several ML methods to predict C_2 yield. Following a similar approach to [8], they used elemental features as input representations rather than inputting the catalyst compositions directly. However, their method differed by removing elemental compositions (removing active metals mass fractions) from the inputs [9].

In another study, Suzuki et al. proposed an innovative method for leveraging literature data by employing elemental features instead of directly inputting catalyst compositions. This novel approach was evaluated across three distinct reaction databases: OCM, WGS, and CO oxidation. Various ML models were tested to validate the effectiveness of this method. The results demonstrated that this technique could facilitate the discovery of new catalysts, even for catalytic reactions with limited information and minimal overlap in catalyst compositions in the literature [8].

Methane steam and dry reforming reactions: Baysal et al. gathered a dataset from 81 publications, comprising 5 508 samples focused on methane steam reforming. The input variables included various catalyst preparations and reaction operating parameters, with the conversion of CH_4 as the output. Selectivity or yield of products was excluded from potential outputs due to the limited availability of these data in the databank. Decision trees were constructed using the dataset to identify heuristics and correlations, aiming to achieve high conversion for the most frequently reported catalysts [10]. Şener et al. compiled a database by extracting experimental data on methane dry reforming from 101 published articles, resulting in a dataset of 5 521 records. The inputs encompassed catalyst properties and operating variables, with CH_4 conversion as the output variable. They employed decision trees to identify trends associated with high performance and used an ANN to assess the relative importance of the features [11].

Methanol synthesis reaction: Tripathi et al. compiled a database from 27 scholarly papers and subsequently developed various ML models using this dataset. The input data comprised preparation variables such as catalyst composition, calcination temperature, preparation method, metal surface area, and pore size distribution, as well as process conditions including reaction temperature, reaction pressure, and H_2/CO_2 molar feed ratio. These models were designed to predict CO_2 conversion and methanol selectivity in the carbon-neutral synthesis of methanol over copper-based catalysts [12].

Selective catalytic reduction (SCR) reaction: Chen et al. compiled 2 126 sample entries from 41 papers on environmental catalysts for NO_x elimination using SCR with NH_3. They collected five types of input data: composition, structure, morphology, preparation method, and reaction conditions, with NO_x conversion as the output variable. An ANN was developed from this dataset to predict NO_x conversion. To test the model's extrapolation capability, they predicted results outside the data range by excluding a subset of data from the database and using the remaining data for training and testing the model [13]. Lu et al. compiled a dataset comprising 5 654 records from 126 papers that included cerium. They developed multiple ML models to identify ternary metal oxides capable of SCR of NO_x using NH_3. The input features for these models included catalyst composition, synthesis conditions, and operating parameters critical for NO_x conversion, with the output being the NO_x conversion efficiency [14].

2. Data preprocessing

Data integration unifies inputs and outputs into a single spreadsheet, standardizing units for specific variables from various sources [15]. Data cleaning aims to remove outliers or correct inconsistencies in the data. Data normalization scales all variables into a specified range [15]. Handling missing data involves deleting or estimating missing values using various techniques [16]. Noise identification enhances the quality of real-world data [17]. Additionally, feature engineering is conducted as part of data preprocessing, with dimensionality reduction being a critical component [18,19]. Dimensionality reduction reduces computational time and cost, prevents overfitting, and improves interpretability for high-dimensional data. It includes two main approaches: feature extraction and feature selection. Feature extraction methods, such as principal component analysis (PCA), linear discriminant analysis (LDA), and autoencoders, make use of the correlation between variables to create a lower-dimensional set of new variables from the original variables without significant loss of information. Feature selection methods aim to identify a subset of the original variables without altering them and are generally categorized as filter, wrapper, or embedded methods. Filter methods evaluate each variable independently of any ML model—for example, using Pearson correlation or mutual information. Wrapper and embedded methods use ML models to guide variable selection. Wrapper methods iteratively select subsets using techniques like genetic algorithms, sequential forward selection, and backward elimination. Embedded methods, on the other hand, determine variable importance during model training by leveraging internal attributes such as regression coefficients or feature importance metrics in ensemble models [20]. These techniques will be applied in following sections, where results will be presented to illustrate their role in addressing a knowledge extraction problem in the field of catalysis.

2.1 Missing data prediction

The initial dataset consists of 1 922 records and 30 variables (29 inputs and 1 output), as shown in Figure 6.2. However, these inputs do not fully represent catalyst properties or synthesis methods, and additional variables contain varying degrees of missing data (Figure B.1). Pore size has the fewest missing values (~26%), while calcination time has the most (~98%). Replacing missing values for variables with over 50% missing data is impractical, but missing values for pore size, particle size, preparation method, shape, calcination temperature, and specific surface area were

initially predicted rather than removed. Approaches for handling missing values include deleting rows or columns or imputing means [16]. ML models, especially autoassociative neural networks (AANNs), are increasingly used for imputation due to their ability to work without prior information [21].

To test AANN's ability to predict missing values, a 40-record evaluation dataset was set aside, while the remaining 1 882 records were split 80% training, 10% validation, and 10% testing. When one variable was missing, its value in the evaluation dataset was replaced with the mean, and the record was fed into the AANN to predict all input values, including the missing one. The estimated value was fed back for recalculation, repeating this process for 10 iterations per record. This works well when missing variables are correlated with other variables. The final AANN model was obtained through hyperparameter tuning (using MSE loss, Adam optimizer with 0.001 learning rate, 20,000 epochs, batch size 32) and demonstrated satisfactory performance on the complete dataset.

The MSE values for all three subsets fall within an acceptable range: 2.82×10^{-4} for training, 3.65×10^{-4} for validation, and 4.48×10^{-4} for testing. To further evaluate the performance of the AANN model to replicate the input variables, the parity plots of the training, validation, and testing datasets for two variables, namely the temperature and conversion, are shown in Figure B.2a and Figure B.2a.

Based on the R^2 and MSE values, it can be concluded that the model predicts the input variables with high precision. However, to rigorously test the AANN model's ability on entirely unseen data, where one variable's column is replaced with the mean value, the 40-record evaluation dataset was fed into the model, following the previously described procedure for predicting missing values. This process was applied to the same two variables, the temperature and conversion, for which the data were available, but purposely removed for evaluation. The results, displayed in Figure B.3a and Figure B.3b, indicate that the AANN model fails to predict the missing values accurately, despite its good performance shown in Figure B.2a and Figure B.2b.

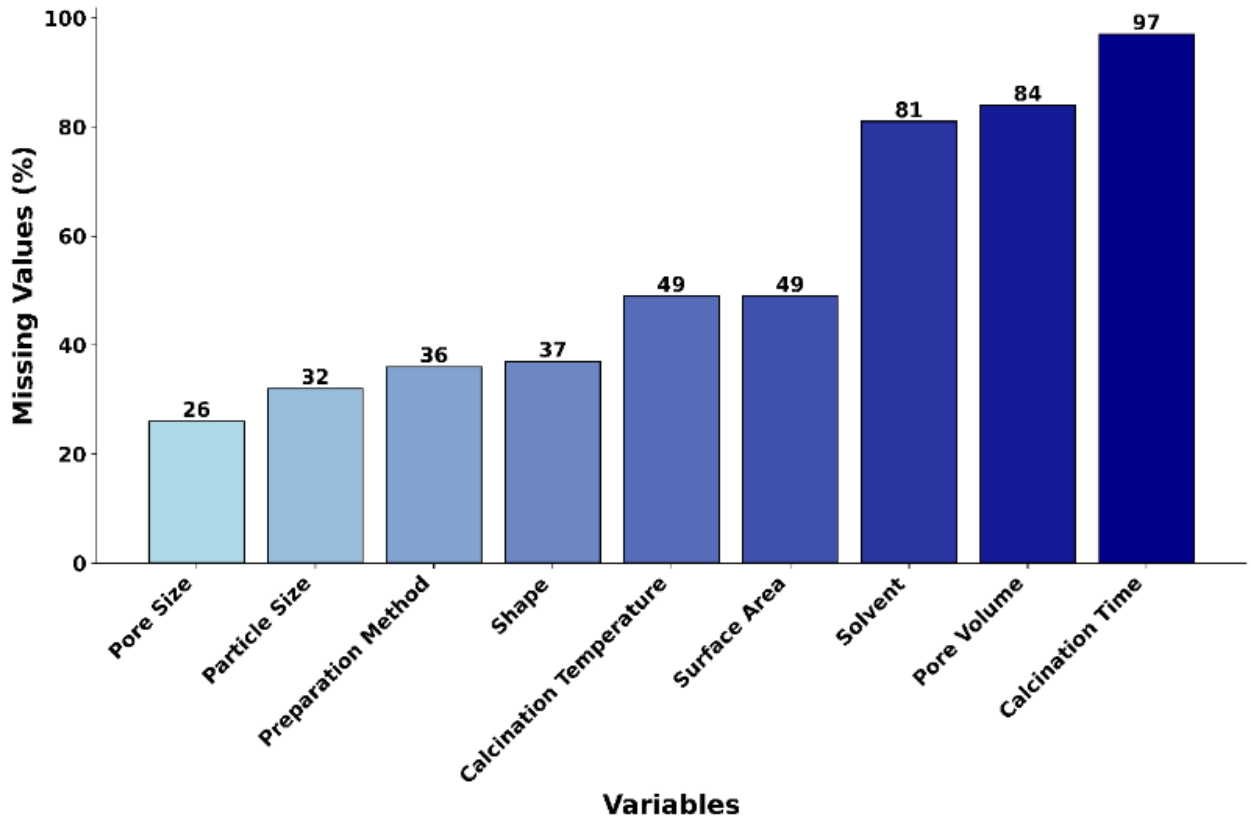


Figure B.1. Number and amount of missing values in the databank related to catalyst physical properties and synthesis techniques.

This discrepancy can be attributed to the low correlation between variables in the dataset. In our previous work [21], we concluded that the correlation between variables is paramount for accurately imputing missing data. In this study, the low correlation between the extracted features without missing values, as indicated by the Pearson correlation coefficient (Figure B.4), hampers the accurate prediction of missing values. It was observed that missing variables also had low correlation with the remaining variables. Consequently, this approach was not pursued to predict the missing values shown in Figure B.1. Despite the loss of important information, the variables with significant missing data from Figure B.1 had to be fully removed from the dataset, resulting in a 30-variable dataset.

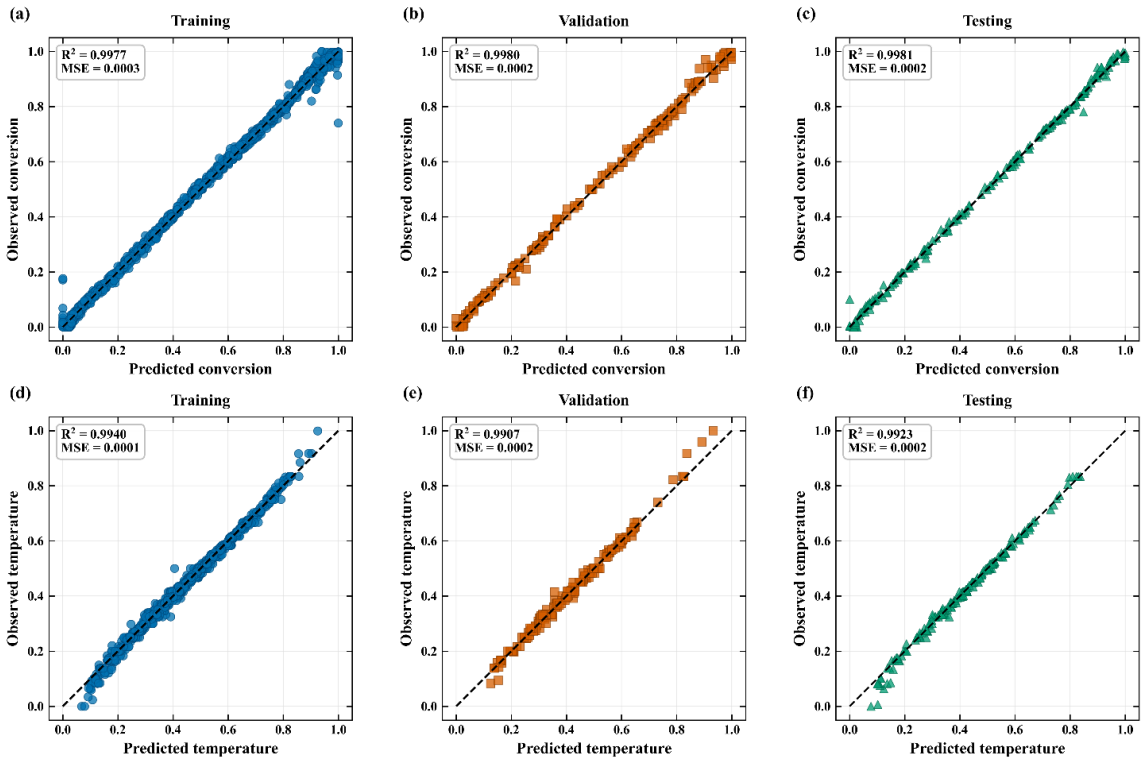


Figure B.2. Parity plots of the AANN model for training, validation, and testing for the prediction of the two randomly selected variables of the 30-variable dataset (a) normalized conversion and (b) normalized temperature.

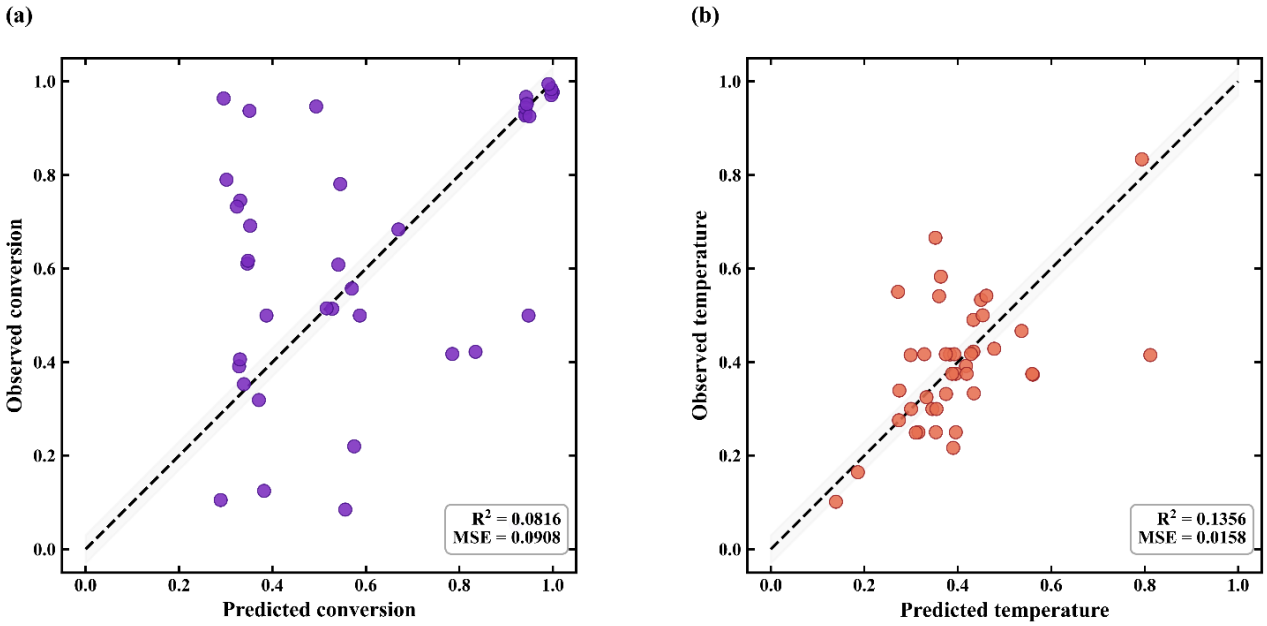


Figure B.3. Parity plots of the estimation of the missing values by the AANN model for conversion and temperature (a) normalized conversion and (b) normalized temperature.

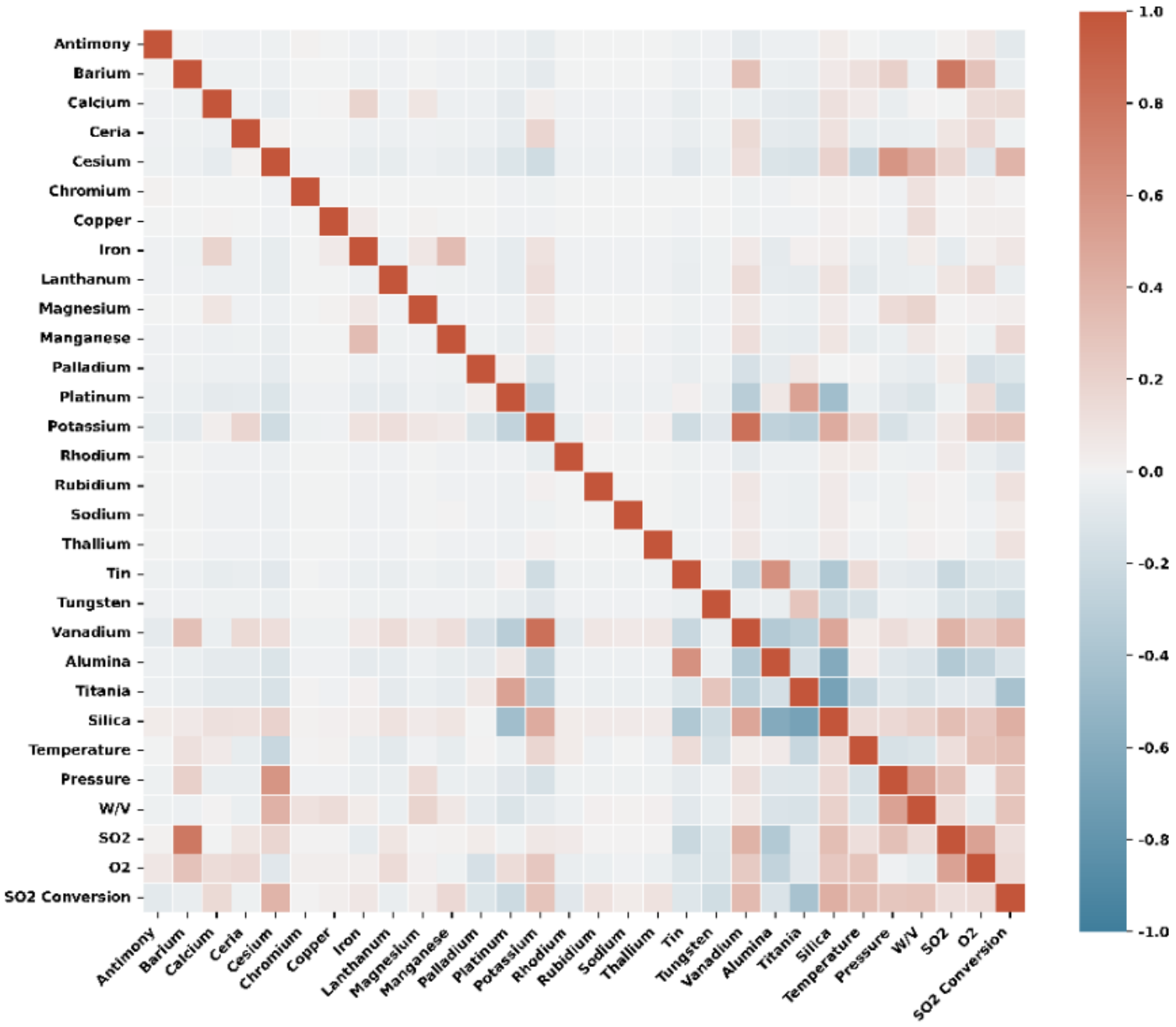


Figure B.4. Heat map of the correlation matrix using the Pearson correlation coefficients for the 30-variable dataset.

2.2 Additional preprocessing

Given the inconclusive results from previous sections, we validated the databank by reviewing data from each paper. This review revealed several duplicate records with negligible differences and records with similar process conditions but different conversions, mainly due to variations in

catalyst properties and preparation methods. As noted in the missing value imputation section, these variables were excluded due to the high number of missing values. Consequently, despite identical compositions and conditions, different conversions appeared. To address this, duplicates were consolidated, and for identical inputs with varying conversions, the average conversion was retained.

Certain variables were removed due to limited records and low variance (Table B.1). For example, antimony (13 records), chromium (34 records), barium (23 records, mostly from one paper with high mole fractions), rhodium (17 records), rubidium (12 records), thallium (11 records), and tungsten (12 records) were all reported with fixed conditions or low variation. Although relevant, these variables were excluded to improve dataset quality.

The removal of these seven metals aligns with the results of the random forest and PCA: Figure 6.3a showed they ranked low in importance, and Figure 6.3b confirmed that 23 PCs capture 95% of the cumulative explained variance, meaning their exclusion does not significantly impact information content.

Table B.1. Summary of elements removed during preprocessing.

Element	Number of records	Conditions/Details	Reason for removal
Antimony	13	Reported at T=450°C with chromium. Chromium had 34 records, also at T=450°C.	Limited records, low variance
Barium	23	One paper: mass fraction 36.08% with vanadium (9.48%), SO_2 (66%) and O_2 (33%) mole fractions.	High mole fractions
	7	Another paper: mass fraction 2.16% with potassium and vanadium.	Limited records, low variance
Rhodium	17	Reported in two cases: 2% mass fraction alone and with platinum at fixed mass fractions.	Limited records, low variance
Rubidium	12	Single mass fraction 1.34%, reported with fixed values of potassium and vanadium.	Limited records, low variance
Thallium	11	Single mass fraction 3.21%, reported with fixed values of potassium and vanadium.	Limited records, low variance
Tungsten	12	Fixed mass fraction 5.5%.	Limited records, low variance

Several preprocessing methods were applied to the reduced dataset prior to developing new ANN models. Initially, after modifying the databank's records, outliers for W/V ratio, SO_2, and O_2 mole fractions were removed using the Z-score method (for $|Z| > 3$). Specifically, records with W/V

values exceeding 30 kg_{cat}·s/L, SO_2 mole fractions above 20%, and O_2 mole fractions exceeding 25% were excluded as outliers. The next preprocessing step involved normalizing the variables to a [0,1] range. In contrast to the main dataset discussed in the previous section, which included 29 inputs and 1 920 data records, the revised dataset in this section is comprised of 22 inputs variables and 1 270 records.

3. Dataset with logical data

The first attempt involved the strategic introduction of logical samples to incorporate records with negative outcomes into the dataset. This was necessitated by the prevalent bias in the literature, where typically only positive outcomes are reported, thereby skewing the dataset towards positive results. To counteract this bias, multiple records representing the absence of metal under varying operating conditions and zero conversion were introduced into the dataset. For instance, the model was designed to understand that potassium as the sole catalyst would result in no conversion, as potassium functions primarily as a promoter. Consequently, 45 logical records were integrated into the revised dataset.

Following the methodology in Figure 6.1, outliers were removed, and Min/Max normalization was applied. Hyperparameter tuning was performed on the modified dataset with logical data, and the optimal ANN model was obtained. The model, trained with a learning rate of 0.001 over 32 epochs using the Nadam optimizer, achieved minimum error at epoch 9 231. Figure B.5 shows parity plots for training, validation, and testing, with MSE within an acceptable range.

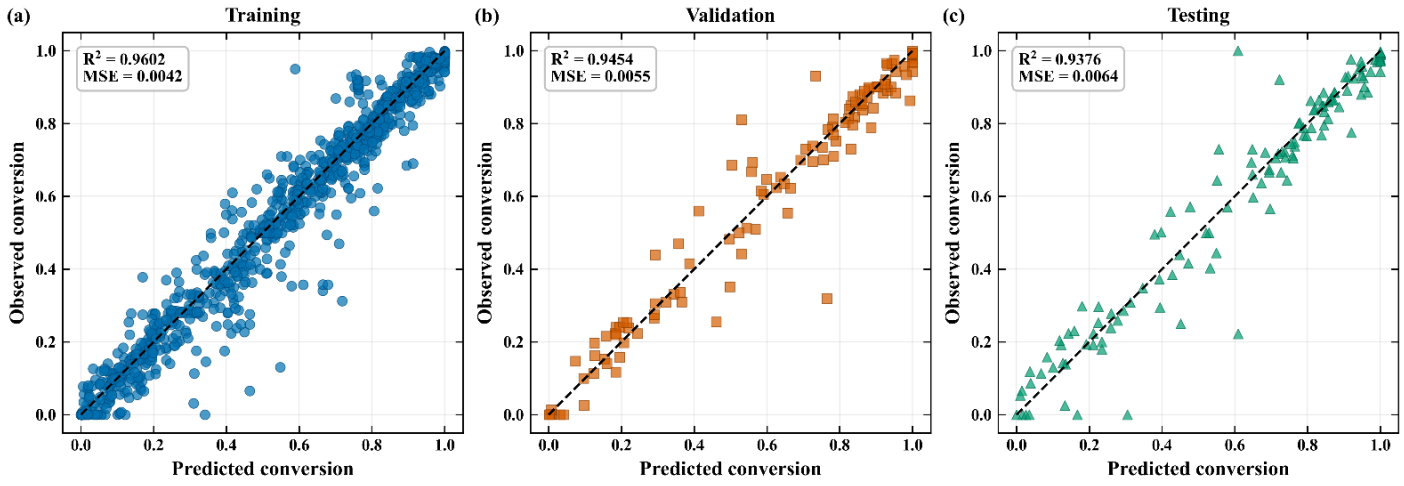


Figure B.5. Parity plots of the ANN model for scenario 1 (with logical data).

Figure B.6 presents the Pareto front derived from the NSGA-II algorithm, utilizing a population of 1 000 and iterating over 10 000 generations. Figure B.6a-c exhibit expected behaviour, revealing the typical trade-offs between conversion, productivity, and cost. However, Figure B.6d highlights that these outcomes are associated with the lower boundary of potassium concentration, suggesting that the incorporation of logical data may have introduced a bias toward negative outcomes. This bias appears to have led the model to incorrectly infer that potassium exerts no influence on conversion, even in the presence of vanadium.

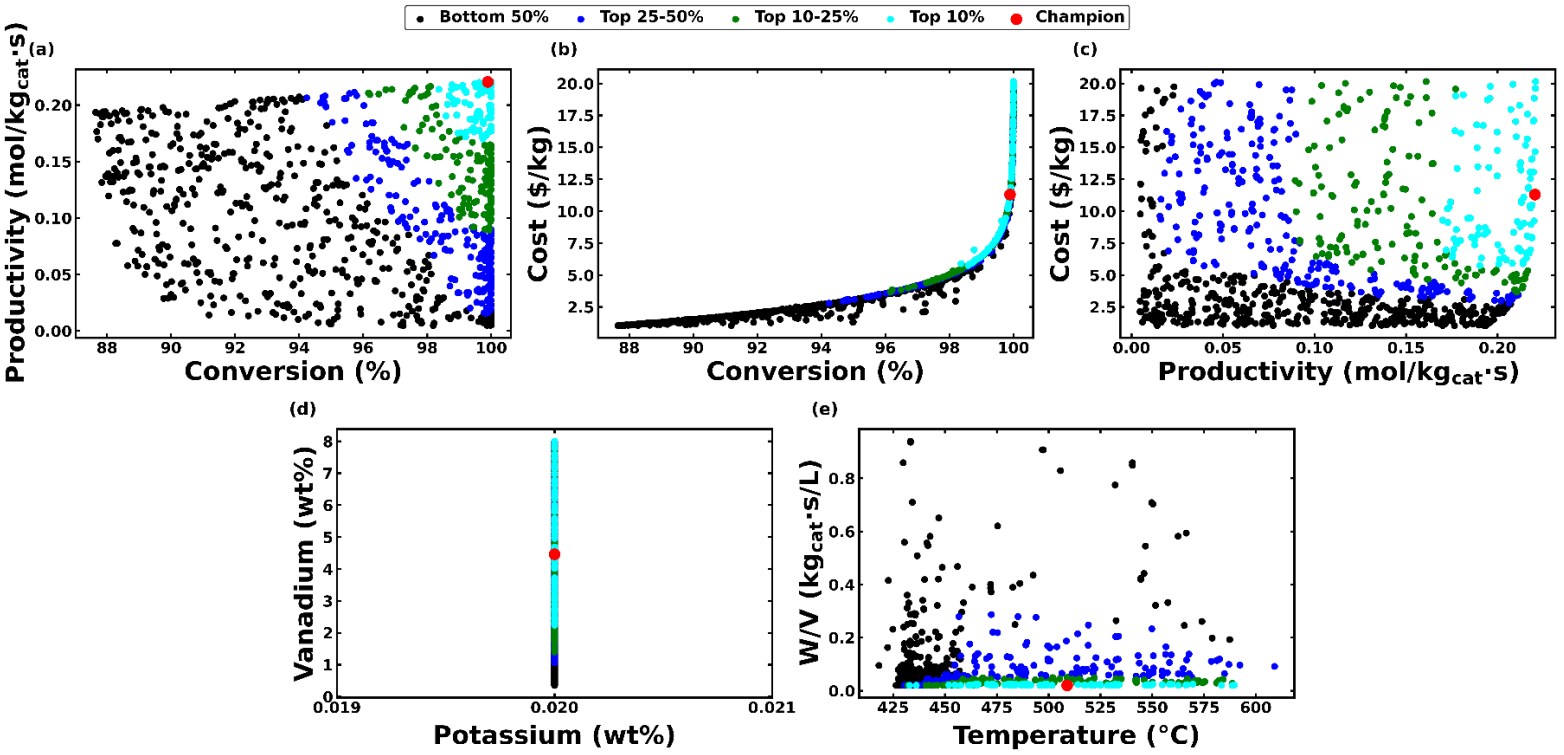


Figure B.6. Results of MOO for vanadium-potassium case study for scenario 1 (with logical data) (a) conversion vs productivity, (b) conversion vs cost, (c) productivity vs cost, (d) vanadium wt% vs potassium wt%, and (e) temperature vs W/V.

4. References

1. Kite, S., Hattori, T., & Murakami, Y. (1994). Estimation of catalytic performance by neural network—Product distribution in oxidative dehydrogenation of ethylbenzene. *Applied Catalysis A: General*, *114*(2), L173–L178. https://doi.org/10.1016/0926-860X(94)80169-X
2. Günay, M. E., & Yildirim, R. (2011). Neural network Analysis of Selective CO Oxidation over Copper-Based Catalysts for Knowledge Extraction from Published Data in the Literature.

Industrial & Engineering Chemistry Research, *50*(22), 12488–12500. https://doi.org/10.1021/ie2013955

3. Günay, M. E., & Yildirim, R. (2013). Knowledge Extraction from Catalysis of the Past: A Case of Selective CO Oxidation over Noble Metal Catalysts between 2000 and 2012. *ChemCatChem*, *5*(6), 1395–1406. https://doi.org/10.1002/cctc.201200665
4. Günay, M. E., & Yildirim, R. (2013). Developing global reaction rate model for CO oxidation over Au catalysts from past data in literature using artificial neural networks. *Applied Catalysis A: General*, *468*, 395–402. https://doi.org/10.1016/j.apcata.2013.08.056
5. Cavalcanti, F. M., Schmal, M., Giudici, R., & Alves, R. M. B. (2019). A catalyst selection method for hydrogen production through Water-Gas Shift Reaction using artificial neural networks. *Journal of Environmental Management*, *237*, 585–594. https://doi.org/10.1016/j.jenvman.2019.02.092
6. Odabaşı, Ç., Günay, M. E., & Yıldırım, R. (2014). Knowledge extraction for water gas shift reaction over noble metal catalysts from publications in the literature between 2002 and 2012. *International Journal of Hydrogen Energy*, *39*(11), 5733–5746. https://doi.org/10.1016/j.ijhydene.2014.01.160
7. Kim, C., & Kim, J. (2022). Machine learning-based high-throughput screening, strategical design and knowledge extraction of Pt/$CE_XZR_{1-X}O_2$ catalysts for water gas shift reaction. *International Journal of Energy Research*, *46*(15), 21293–21308. https://doi.org/10.1002/er.8488
8. Suzuki, K., Toyao, T., Maeno, Z., Takakusagi, S., Shimizu, K., & Takigawa, I. (2019). Statistical Analysis and Discovery of Heterogeneous Catalysts Based on Machine Learning from Diverse Published Data. *ChemCatChem*, *11*(18), 4537–4547. https://doi.org/10.1002/cctc.201900971
9. Mine, S., Takao, M., Yamaguchi, T., Toyao, T., Maeno, Z., Hakim Siddiki, S. M. A., Takakusagi, S., Shimizu, K., & Takigawa, I. (2021). Analysis of Updated Literature Data up to 2019 on the Oxidative Coupling of Methane Using an Extrapolative Machine-Learning Method to Identify Novel Catalysts. *ChemCatChem*, *13*(16), 3636–3655. https://doi.org/10.1002/cctc.202100495
10. Baysal, M., Günay, M. E., & Yıldırım, R. (2017). Decision tree analysis of past publications on catalytic steam reforming to develop heuristics for high performance: A statistical review.

International Journal of Hydrogen Energy, *42*(1), 243–254. https://doi.org/10.1016/j.ijhydene.2016.10.003

11. Şener, A. N., Günay, M. E., Leba, A., & Yıldırım, R. (2018). Statistical review of dry reforming of methane literature using decision tree and artificial neural network analysis. *Catalysis Today*, *299*, 289–302. https://doi.org/10.1016/j.cattod.2017.05.012
12. Tripathi, K., Gupta, V., Awasthi, V., Pant, K. K., & Upadhyayula, S. (2023). Forecasting Catalytic Property-Performance Correlations for CO_2 Hydrogenation to Methanol via Surrogate Machine Learning Framework. *Advanced Sustainable Systems*, *7*(3), 2200416. https://doi.org/10.1002/adsu.202200416
13. Chen, Y., Li, R., Suo, H., & Liu, C. (2021). Evaluation of a Data-Driven, Machine Learning Approach for Identifying Potential Candidates for Environmental Catalysts: From Database Development to Prediction. *ACS ES&T Engineering*, *1*(8), 1246–1257. https://doi.org/10.1021/acsestengg.1c00125
14. Lu, M., Gao, F., Tan, Y., Yi, H., Gui, Y., Xu, Y., Wang, Y., Zhou, Y., Tang, X., & Chen, L. (2024). Knowledge-Driven Experimental Discovery of Ce-Based Metal Oxide Composites for Selective Catalytic Reduction of NO_x with NH_3 through Interpretable Machine Learning. *ACS Applied Materials & Interfaces*, *16*(3), 3593–3604. https://doi.org/10.1021/acsami.3c18490
15. Malik, J. S., Goyal, P., & Sharma, A. K. (2010). A comprehensive approach towards data preprocessing techniques & association rules. *Proceedings of the 4th National Conference*, *132*.
16. Luengo, J., García, S., & Herrera, F. (2012). On the choice of the best imputation methods for missing values considering three groups of classification methods. *Knowledge and Information Systems*, *32*, 77–108. https://doi.org/10.1007/s10115-011-0424-2
17. Frénay, B., & Verleysen, M. (2013). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(5), 845–869. https://doi.org/10.1109/TNNLS.2013.2292894
18. Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, *1*(1), 56–70. https://doi.org/10.38094/jastt1224

19. Khaire, U. M., & Dhanalakshmi, R. (2022). Stability of feature selection algorithm: A review. *Journal of King Saud University-Computer and Information Sciences*, *34*(4), 1060–1073. https://doi.org/10.1016/j.jksuci.2019.06.012
20. Trinh, C., Tbatou, Y., Lasala, S., Herbinet, O., & Meimaroglou, D. (2023). On the Development of Descriptor-Based Machine Learning Models for Thermodynamic Properties: Part 1—From Data Collection to Model Construction: Understanding of the Methods and Their Effects. *Processes*, *11*(12), 3325. https://doi.org/10.3390/pr11123325
21. Agin, F., Thibault, J., & Fauteux-Lefebvre, C. (2024). Autoassociative neural network for missing data imputation: A case study via the styrene production process. *The Canadian Journal of Chemical Engineering*, *103*, 339–358. https://doi.org/10.1002/cjce.25377

Appendix C: Supporting Information of Chapter 7

A Data-Driven Approach to Competitive Hydrogenation: Structure-Informed Machine Learning with Chemoselectivity-Based Optimization

Web Application Deployment for Catalyst Screening and Optimization

1. Introduction

To improve the accessibility and usability of the developed predictive framework, the best-performing ML models were integrated into a web-based application for catalyst evaluation and operating-condition analysis. The web app was developed in Python using Streamlit (https://streamlit.io/), providing an interactive platform through which users can assess catalyst performance under specified conditions and through optimization-based analysis, without requiring direct interaction with the underlying modelling code. In this way, the application serves as a practical interface for applying the predictive capabilities developed in this work to hydrogenation reaction screening and catalyst selection.

The application was designed for the hydrogenation system investigated in this study and focuses on identifying the most suitable catalyst for selected pairs of substrates. Its purpose is to provide a structured and user-friendly environment in which trained ML models can be used to compare catalyst behaviour, examine the effect of operating conditions, and support decision-making using predicted reaction outcomes. By translating model outputs into an interpretable digital tool, the application facilitates the use of ML results in a form that is more accessible for analysis, comparison, and practical evaluation.

The web app is organized into two complementary workflows. The first workflow evaluates catalyst performance under a set of pre-defined operating conditions. In this mode, two substrates are selected by the user, and the selectivity score is computed iteratively for each catalyst across a set of representative operating conditions derived from trends observed in the available experimental data. A total of 21 operating conditions are considered in this workflow. For each case, the trained ML models generate the predicted reaction outcomes, and the selectivity score is then evaluated using Equation (7.1). The catalyst associated with the highest selectivity score is reported as the preferred catalyst under the examined conditions. To support interpretation of the results, the application also provides visual outputs, including heat maps and score plots, which allow users to compare catalyst performance across the selected operating-condition space.

The second workflow is based on optimization. In this mode, a genetic algorithm is coupled with the trained ML models to maximize the selectivity score. In contrast to the first workflow, which evaluates a discrete and pre-defined set of operating conditions, the optimization-based workflow searches across the feasible operating space to identify more favourable process

conditions. This enables the app to explore combinations of operating variables beyond the specific cases directly represented in the available experimental dataset. As a result, the optimization workflow provides a broader search capability for identifying promising catalyst and condition combinations.

For this study, the optimization was conducted separately for all possible substrate-pair combinations considered in the analysis. For each selected pair, the application reports the catalyst predicted to provide the best performance together with the corresponding optimal operating conditions. This allows the user to move from simple catalyst screening to a more advanced analysis in which both catalyst choice and process settings are considered simultaneously.

Overall, the developed web application demonstrates the deployment of trained ML models into an interactive and practical digital environment for catalyst screening and operating-condition evaluation. By combining predictive modelling, visual interpretation, and optimization within a single platform, the app provides a clear and efficient means of examining catalyst performance under both fixed and optimized conditions. The following sections of this appendix describe the structure of the application and explain the two main workflows used for catalyst assessment and process-condition analysis.

Figure C.1 provides the overall view of the web application interface and illustrates how the user interacts with the deployed tool. The layout is divided into two main areas. On the left, a sidebar is provided for user input, where the two substrates are selected before running the analysis. In the main panel, the two workflows are displayed as separate sections so that the user can clearly distinguish between screening under pre-defined conditions and optimization-based analysis. This arrangement was designed to make the navigation straightforward and to allow the user to move easily between input selection and result generation.

The figure also serves as the starting point for the appendix by introducing the structure of the app before presenting detailed examples. Rather than discussing the application only in abstract terms, the next sections examine how each part functions through a specific example in which one compound is assigned as **Substrate A** and another as **Substrate B**. Using this example-based format makes it possible to show the sequence of user actions, the outputs displayed by the app, and the way the results are interpreted in practice.

Accordingly, Figure C.1 should be viewed as a general overview of the deployed interface. The detailed operation of each module, including the displayed results and their interpretation, is presented in the following sections through step-by-step examples for the selected substrate pair.

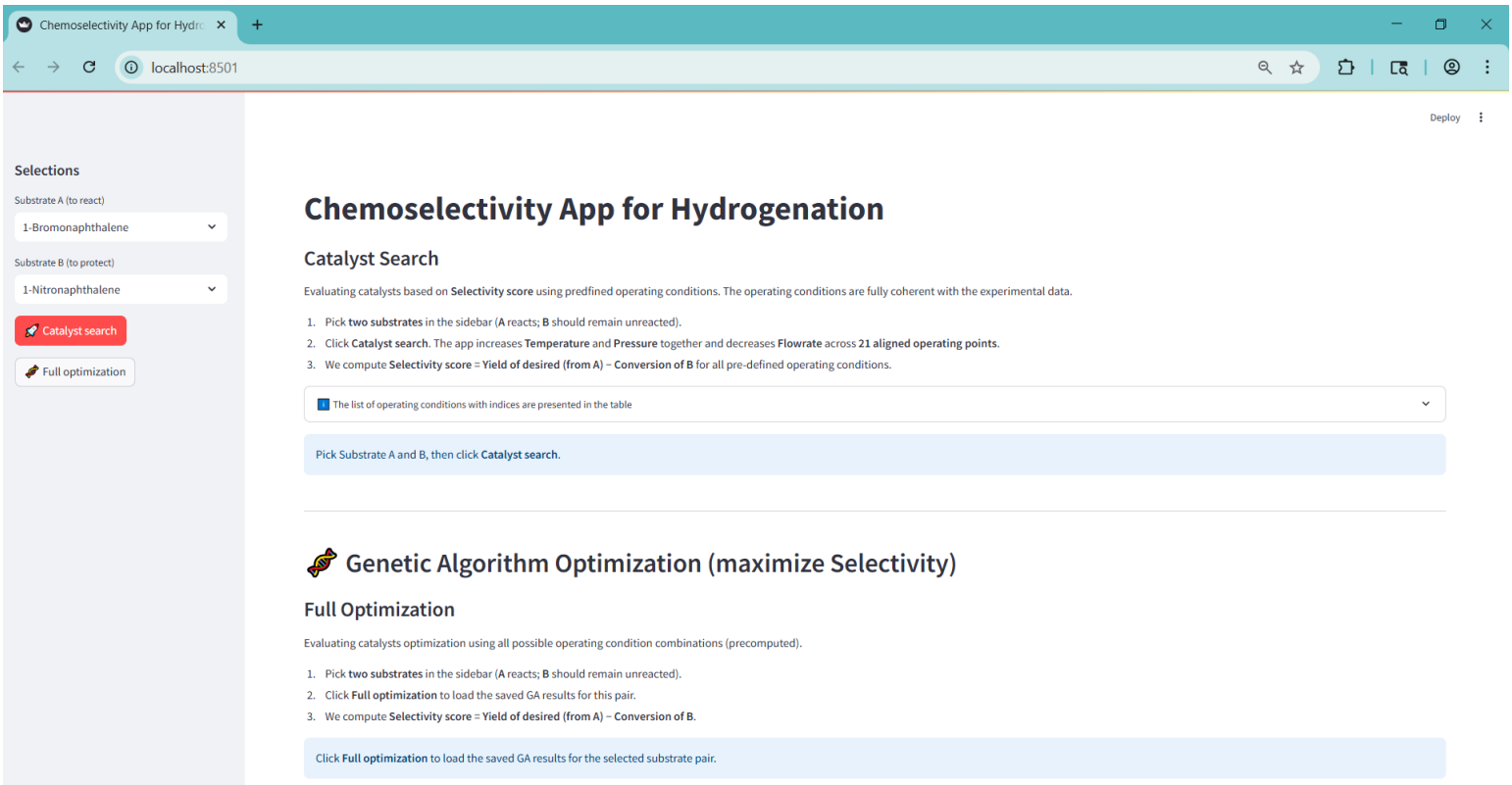


Figure C.1. Overall interface of the developed Streamlit-based chemoselectivity web application for hydrogenation, showing the sidebar for substrate selection and the two main analysis sections, Catalyst Search and Full Optimization.

Figure C.2 shows the first stage of user interaction with the web application. Before running any analysis, the user defines the substrate pair directly from the sidebar by selecting one compound as Substrate A and the other as Substrate B. In the example presented here, 1-Nitronaphthalene is assigned as the reacting substrate and 2-Phenylacetophenone is assigned as the protected substrate. This selection establishes the chemoselective hydrogenation problem to be evaluated by the app and determines which stored predictions and optimization results are retrieved in the subsequent steps.

The same panel also provides the two analysis buttons available to the user. These buttons serve as the entry point to the two workflows described earlier and allow the selected substrate

pair to be analyzed under either the screening-based or optimization-based framework. At this stage, the role of the interface is to define the problem setup clearly and provide a simple transition from substrate selection to result generation.

For consistency, the same substrate pair is used throughout the following demonstration of the appendix. The discussion begins with the Catalyst Search workflow, where the selected pair is evaluated under the pre-defined operating conditions, and then proceeds to the optimization-based results presented by the Full optimization module. This step-by-step presentation makes it possible to illustrate not only how the user interacts with the app, but also how the outputs are generated and interpreted for a specific chemoselective hydrogenation case.

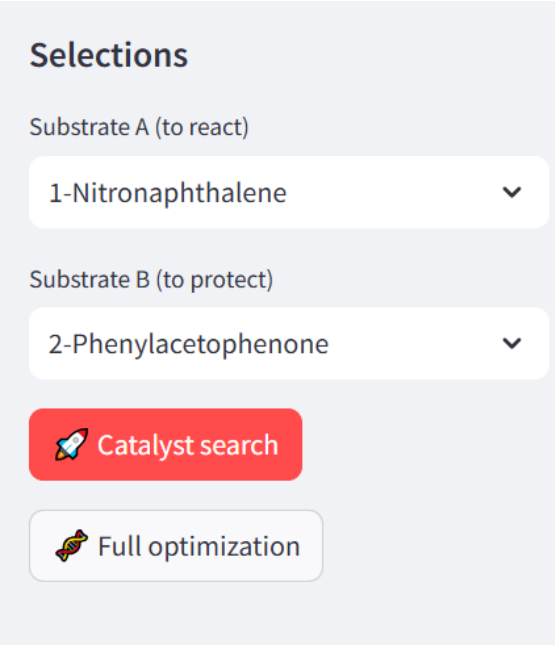


Figure C.2. Sidebar of the chemoselectivity web application showing an example substrate-pair selection, with 1-Nitronaphthalene chosen as Substrate A and 2-Phenylacetophenone chosen as Substrate B, together with the two analysis options, Catalyst search and Full optimization.

2. Catalyst search

Figure C.3 presents the introductory content displayed in the Catalyst Search section of the app. This part of the interface provides a concise explanation of the screening workflow and outlines the main steps required to perform the analysis for the selected substrate pair. By presenting these

instructions directly in the main panel, the app guides the user through the procedure before any results are displayed and clarifies how the screening process is structured within this module.

The information shown in this section is intended to make the workflow more transparent and easier to follow. Rather than requiring the user to infer how the analysis is carried out, the interface explicitly states the sequence of actions and the general basis of the catalyst search procedure. This contributes to a more interpretable and user-friendly deployment environment, particularly for users who may not be familiar with the computational details behind the app.

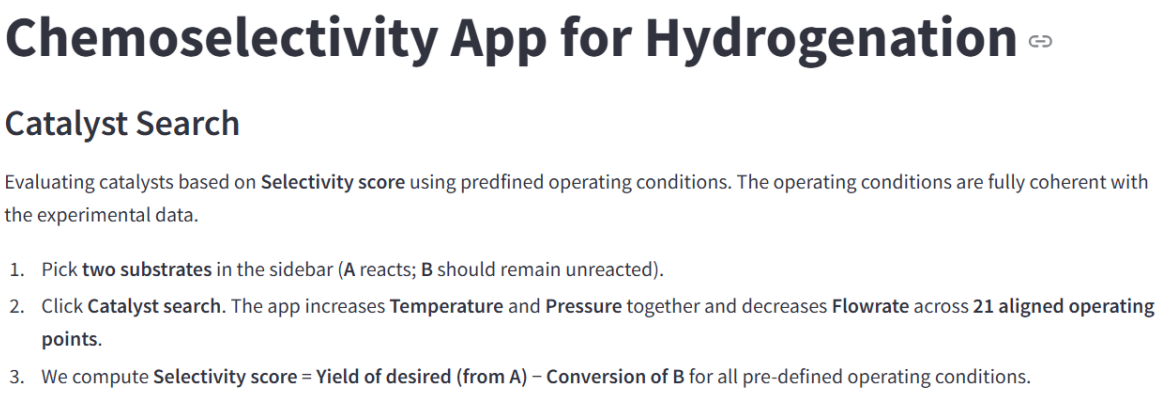


Figure C.3. Overview of the Catalyst Search section in the web application, showing the on-screen guidance provided to the user before running the screening analysis.

Figure C.4 shows the expandable More Info panel included in the Catalyst Search section of the web app. This panel displays the complete list of operating conditions used during the screening analysis and allows the user to inspect the exact temperature, pressure, and flow rate values applied in catalyst evaluation. Presenting this table directly in the app improves transparency by making the basis of the catalyst screening explicitly visible to the user rather than leaving it implicit in the background calculations.

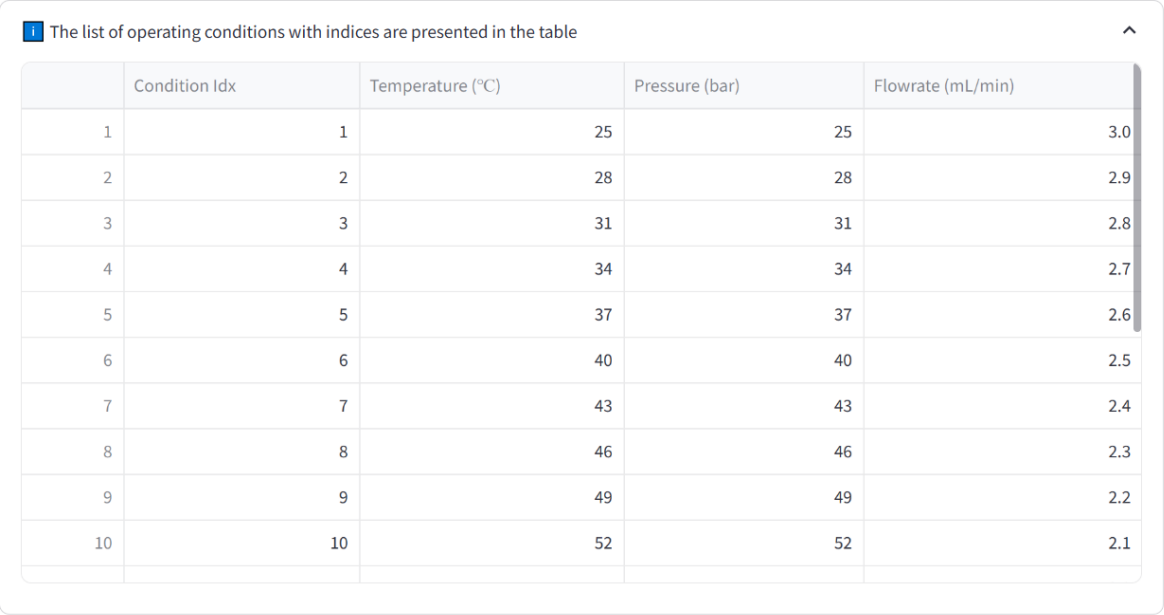
The table consists of 21 indexed operating points, but only five of these correspond to the original experimental conditions available in the dataset. These five points, shown in bold, are the operating conditions on which the ML models were trained. They serve as the anchor points of the screening workflow and are identified in the app by condition indices 1, 6, 11, 16, and 21. The remaining 16 points were added as interpolated intermediate conditions between these

experimental anchor points to provide a smoother and more informative screening trajectory across the operating domain.

These interpolated points were not generated as arbitrary combinations of operating variables. Instead, they were constructed along a single aligned path in which temperature and pressure increase together while flow rate decreases simultaneously. Temperature is varied from 25 to 85 °C in increments of 3 °C, pressure is varied from 25 to 85 bar in increments of 3 bar, and flow rate is varied from 3.0 to 1.0 mL/min in decrements of 0.1 mL/min. This produces a total of 21 operating conditions that follow a continuous transition from mild to harsh operating severity.

For example, Condition 1 corresponds to 25 °C, 25 bar, and 3.0 mL/min. The next operating point, Condition 2, becomes 28 °C, 28 bar, and 2.9 mL/min, followed by Condition 3, which is 31 °C, 31 bar, and 2.8 mL/min. This pattern continues throughout the table such that each step increases temperature and pressure by the same amount while decreasing flow rate by a fixed amount. At the opposite end of the range, Condition 21 corresponds to 85 °C, 85 bar, and 1.0 mL/min, representing the harshest operating point in the predefined screening path. In this way, the full 21-point table should be interpreted as a continuous operating trajectory anchored by the five experimental conditions and filled in with intermediate points to show the gradual evolution of operating severity.

This structure provides two practical benefits in the app. First, it preserves a direct connection to the experimental basis of the predictive models by clearly retaining the original trained conditions. Second, it allows the catalyst search module to evaluate and visualize catalyst performance across a denser and more continuous set of operating conditions, rather than restricting the analysis to only five discrete points. As a result, Figure C.4 is important not only because it displays the operating-condition table, but also because it explains how the predefined screening space was constructed to balance experimental grounding with a smoother and more interpretable catalyst screening process.



	Condition Idx	Temperature (°C)	Pressure (bar)	Flowrate (mL/min)
1	1	25	25	3.0
2	2	28	28	2.9
3	3	31	31	2.8
4	4	34	34	2.7
5	5	37	37	2.6
6	6	40	40	2.5
7	7	43	43	2.4
8	8	46	46	2.3
9	9	49	49	2.2
10	10	52	52	2.1

Figure C.4. Expandable More Info panel in the Catalyst Search section, showing the 21 operating conditions used in the screening workflow. The bold rows indicate the original experimental operating conditions used for model training, while the remaining rows represent interpolated intermediate conditions introduced to create a continuous operating path.

Figure C.5 presents the outcome of the catalyst screening analysis for the selected substrate pair, 1-Nitronaphthalene as Substrate A and 2-Phenylacetophenone as Substrate B. In this step, all available candidate catalysts are evaluated over the full set of 21 pre-defined operating conditions, and their predicted performances are compared on the basis of the chemoselectivity score. The purpose of this figure is to identify the catalyst that provides the most favourable balance between promoting the desired hydrogenation of Substrate A and minimizing the undesired conversion of Substrate B.

For each catalyst and each operating condition, the trained ML models are used to predict the relevant reaction outcomes, including the yield of the desired product from Substrate A and the conversion of Substrate B. These predicted values are then combined through the chemoselectivity expression defined previously in Equation 1. After performing this calculation over all 21 operating conditions, the maximum chemoselectivity score achieved by each catalyst is retained. The catalyst associated with the overall highest score is then selected as the best catalyst candidate for the chosen substrate pair.

As shown in Figure C.5, 1% Pt(V)/C is identified as the best hydrogenation catalyst, achieving a maximum chemoselectivity score of 99%. This result indicates that, among all catalysts considered, 1% Pt(V)/C provides the strongest predicted performance for selectively hydrogenating the desired substrate while suppressing the undesired reaction of the protected substrate under at least one of the pre-defined operating conditions. The remaining catalysts are ranked according to their own maximum achievable chemoselectivity scores, allowing the user to compare not only the top candidate but also the relative performance of the full catalyst set.

The bar chart format makes this comparison straightforward by summarizing the best achievable screening result for each catalyst in a single view. Catalysts with high positive scores can be interpreted as strong candidates for selective hydrogenation, whereas catalysts with low or negative scores indicate poor selectivity for the selected substrate pair. Therefore, Figure C.5 provides the first key decision output of the web app by identifying the most promising catalyst candidate before examining the detailed operating-condition dependence of its performance.

Figure C.6 shows the metric-selection component of the Catalyst Search workflow. After the best catalyst candidate is identified, the user can further examine catalyst behaviour by selecting one of the available performance metrics for visualization. The available options include selectivity score, yield of desired product from Substrate A, conversion of A, and conversion of B. This feature allows the user to move beyond a single summary result and investigate catalyst performance from different analytical perspectives.

Once a metric is selected, the application generates two complementary visual outputs based on that choice: a heat map and a score plot. These plots are used to examine how the selected metric varies across catalysts and operating conditions, thereby providing a more detailed view of catalyst behaviour for the chosen substrate pair. In this way, the user can assess not only which catalyst performs best overall, but also how performance changes across the operating range and whether the observed behaviour is driven by desired reactivity, undesired conversion, or the combined chemoselectivity outcome.

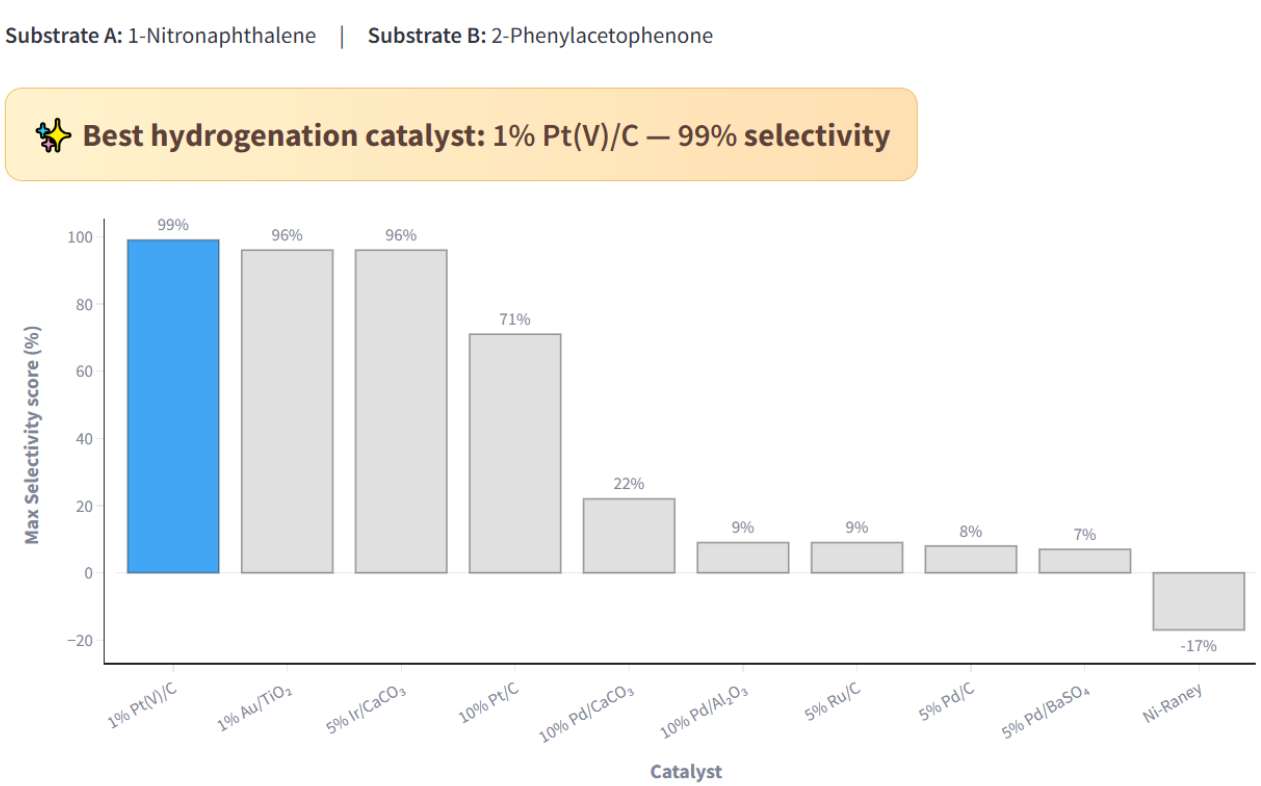


Figure C.5. Result of the Catalyst Search workflow for the selected substrate pair, showing the maximum chemoselectivity score achieved by each candidate catalyst across the 21 pre-defined operating conditions. The highlighted bar identifies the catalyst with the highest score and therefore the selected best catalyst candidate.

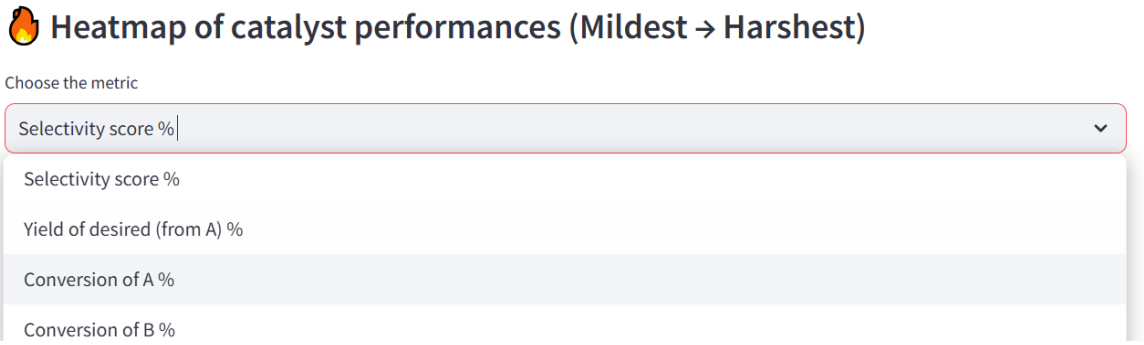


Figure C.6. Metric-selection menu in the Catalyst Search workflow, allowing the user to choose the performance indicator to be visualized. Based on the selected metric, the application generates the corresponding heat map and score plot for catalyst comparison across the pre-defined operating conditions.

The following sections present these visual outputs in detail and explain how they can be interpreted for catalyst assessment under the selected screening conditions.

Figure C.7 presents the heat map generated after selecting chemoselectivity score as the metric of interest. This visualization provides a compact comparison of all available catalysts across the five representative operating conditions used to summarize the predefined screening path. As described previously, these five conditions correspond to the original experimental anchor points distributed across the full operating range, namely the mildest, mild, intermediate, harsh, and harshest conditions. By focusing on these five key levels, the heat map enables the user to observe how catalyst performance changes as the operating severity increases.

Each cell in the heat map reports the predicted value of the selected metric for a specific catalyst at a specific operating severity. In the present case, the values correspond to chemoselectivity score, allowing the user to assess how effectively each catalyst promotes the desired hydrogenation of Substrate A while suppressing the undesired conversion of Substrate B. The colour scale provides an immediate visual indication of performance, with higher positive values representing more favourable catalyst behaviour and lower or negative values indicating poorer chemoselectivity.

This type of visualization is especially useful because it reveals not only which catalyst performs best overall, but also how the relative ranking of catalysts changes across the operating range. A catalyst may perform well under mild conditions but lose selectivity as the severity increases, whereas another may show more stable behaviour across all operating points. In this way, the heat map complements the summary ranking presented earlier by showing the operating-condition dependence of catalyst performance rather than only the single best score.

For the selected substrate pair in this example, Figure C.7 shows that 1% Pt(V)/C maintains consistently high chemoselectivity across all five representative conditions, which is in agreement with its selection as the best catalyst candidate in the previous figure. Other catalysts display more variable behaviour, with some showing moderate selectivity only under certain conditions and others exhibiting strongly negative scores as the operating severity increases. Therefore, this figure

provides a more detailed interpretation of catalyst performance by showing how the selected metric evolves across the key operating conditions used in the screening analysis.

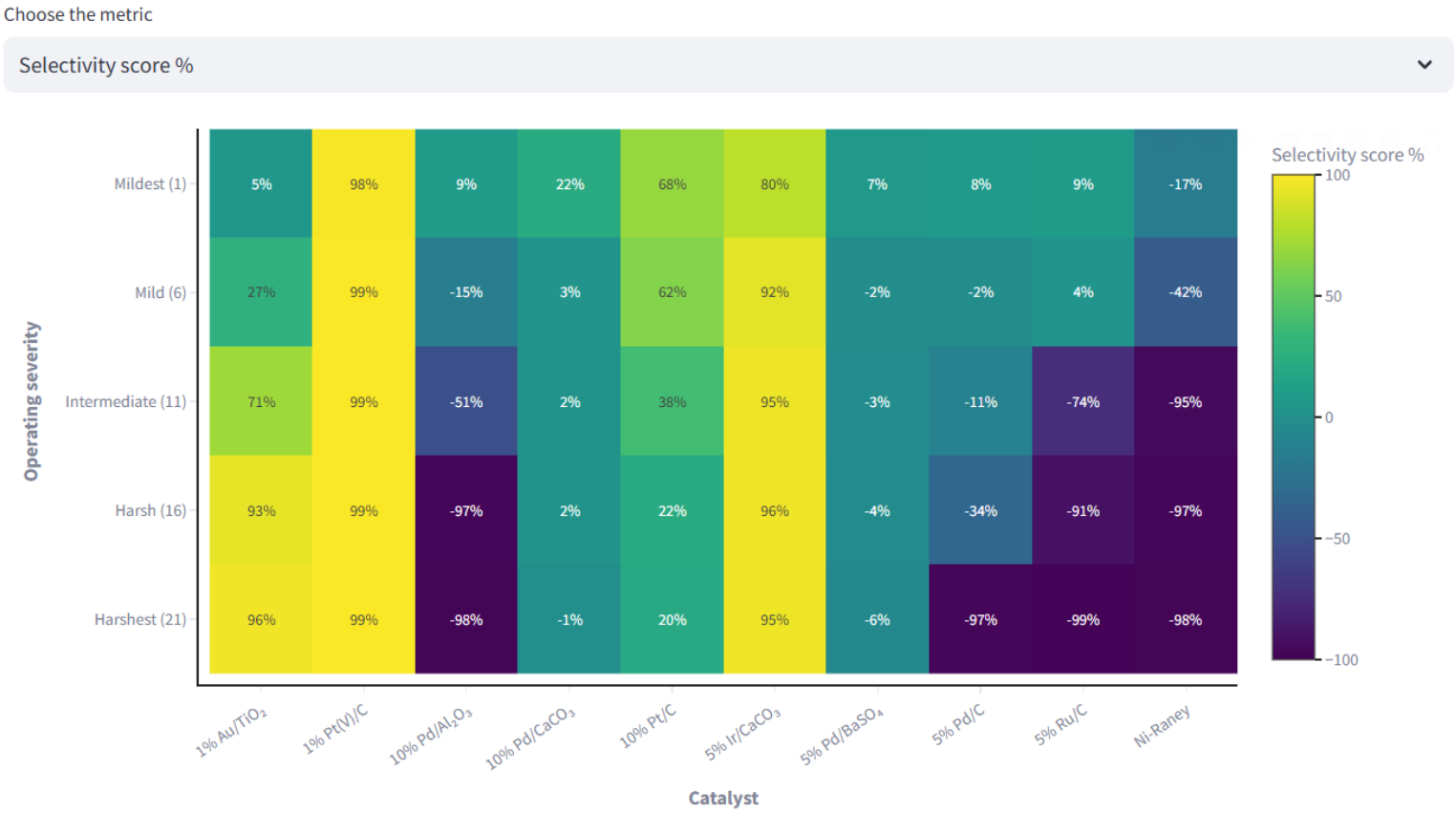


Figure C.7. Heat map of catalyst performance for the selected substrate pair based on the chosen metric, shown here for chemoselectivity score. The heat map compares all available catalysts across the five representative operating conditions corresponding to the mildest, mild, intermediate, harsh, and harshest levels of operating severity.

Figure C.8 presents the second visualization linked to the metric selected in the Catalyst Search workflow. While the heat map summarizes catalyst performance at the five representative operating conditions, this plot shows the variation of the selected metric across all 21 pre-defined operating conditions for every available catalyst. In the present example, the selected metric is chemoselectivity score.

This figure provides a more detailed view of catalyst behaviour over the full operating trajectory. Each line represents one catalyst, and the horizontal axis follows the indexed operating conditions from the mildest to the harshest level. In this way, the user can observe not only the

overall performance level of each catalyst, but also how that performance evolves as temperature and pressure increase and flow rate decreases across the screening range.

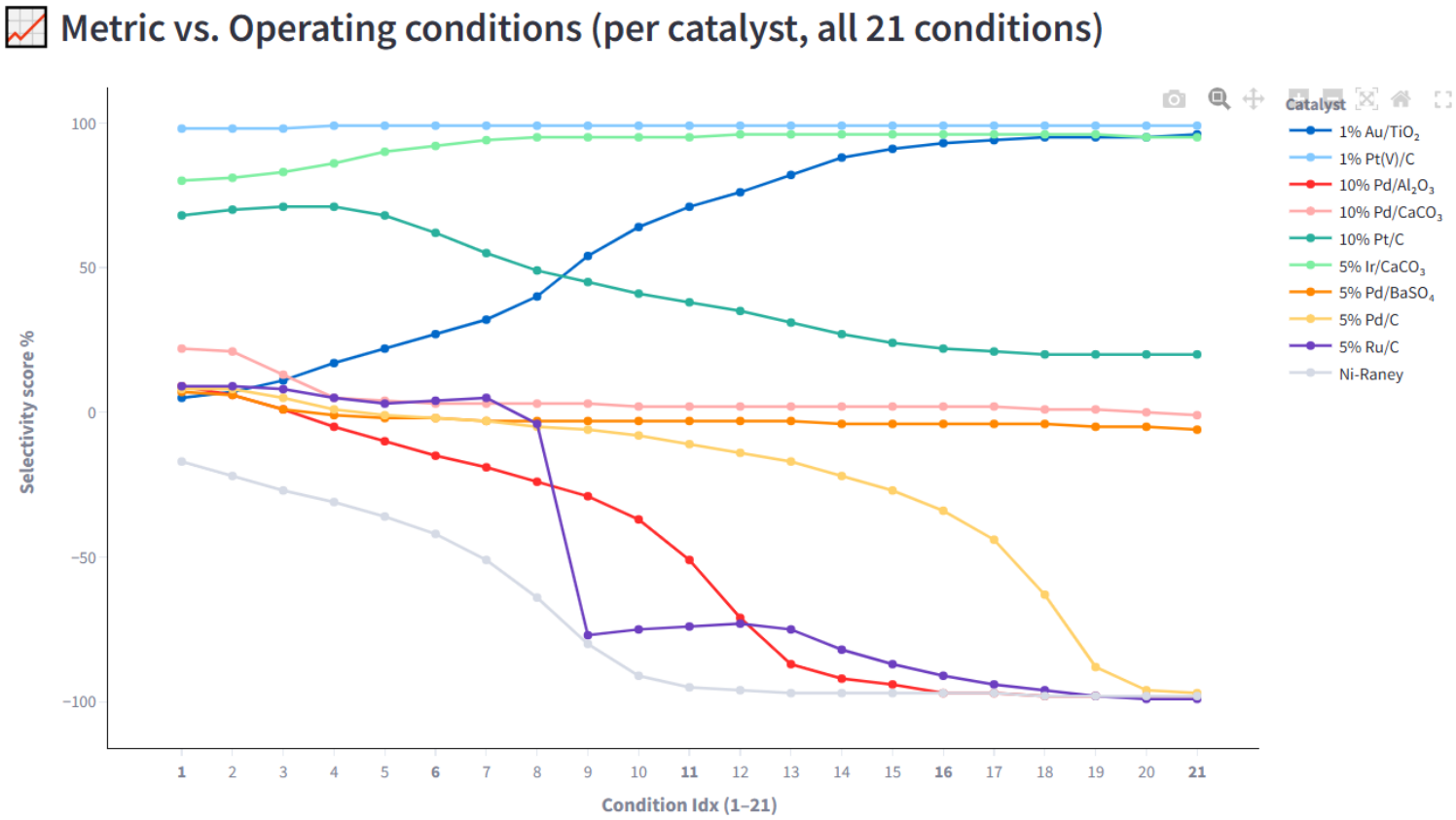


Figure C.8. Score plot of the selected metric across all 21 pre-defined operating conditions for each available catalyst. In this example, the plotted metric is chemoselectivity score, allowing comparison of catalyst performance over the full screening range.

The main value of this plot is that it reveals performance trends that may not be fully visible in a summary ranking or a five-point heat map. For example, some catalysts may show consistently high chemoselectivity across nearly the entire operating range, while others may perform reasonably well only under mild conditions and then deteriorate rapidly as the operating severity increases. Similarly, catalysts with low or negative values can be identified clearly, indicating poor suitability for the selected substrate pair over part or all of the operating domain.

For the substrate pair considered in this example, Figure C.8 shows that 1% Pt(V)/C maintains an almost constant and very high chemoselectivity score across all 21 operating conditions, which further supports its identification as the best catalyst candidate. In contrast, other catalysts exhibit

more variable behaviour, with some improving as the severity increases and others declining markedly. Therefore, this plot complements the previous figures by providing a continuous view of catalyst performance throughout the full predefined operating path and allowing a broader comparison of catalyst stability and selectivity trends.

Figure C.9 presents the complete tabulated output of the Catalyst Search workflow. After the screening analysis is performed, all predicted results are compiled into a single table that includes the operating condition index, temperature, pressure, flow rate, catalyst name, yield of the desired product from Substrate A, conversion of A, conversion of B, and the resulting chemoselectivity score. In this way, the table provides the full numerical basis underlying the summary plots and ranking outputs shown earlier in the application.

This section is important because it gives the user direct access to the detailed results for every catalyst under every pre-defined operating condition. Rather than relying only on graphical summaries, the user can inspect the exact predicted values associated with each catalyst-condition combination and identify specific cases of interest. This is particularly useful when a more detailed comparison is needed or when the user wishes to examine the numerical values behind the trends observed in the heat map and score plot. In addition to being displayed within the app, the table can also be downloaded as a CSV file. This feature allows the user to save the complete set of screening results and use them for further analysis outside the application environment. Therefore, Figure C.9 represents the final output stage of the Catalyst Search workflow by providing a complete, accessible, and exportable summary of all screening results.

Full catalyst search results

	Condition Idx	Temperature (°C)	Pressure (bar)	Flowrate (mL/min)	Catalyst	Yield of desired (from A) %	Conversion of A %	Conversion of B %	Selectivity score %
1	1	25	25	3.0	1% Pt(V)/C	98	100	0	98
2	1	25	25	3.0	5% Ir/$CaCO_3$	84	93	4	80
3	1	25	25	3.0	10% Pt/C	93	100	25	68
4	1	25	25	3.0	10% Pd/$CaCO_3$	98	100	76	22
5	1	25	25	3.0	10% Pd/Al_2O_3	96	100	87	9
6	1	25	25	3.0	5% Ru/C	10	10	1	9
7	1	25	25	3.0	5% Pd/C	96	100	88	8
8	1	25	25	3.0	5% Pd/$BaSO_4$	97	100	91	7
9	1	25	25	3.0	1% Au/TiO_2	6	4	1	5
10	1	25	25	3.0	Ni-Raney	82	100	98	-17

Download full catalyst search results (CSV)

Figure C.9. Full results table generated by the Catalyst Search workflow, summarizing the predicted performance of all catalysts across the 21 pre-defined operating conditions for the selected substrate pair. The table can also be downloaded by the user as a CSV file for further analysis.

3. Full optimization

Figure C.10 shows the initial interface displayed when the user enters the Full Optimization section of the web app. This part of the application corresponds to the second analysis workflow and is designed to identify the highest achievable selectivity score together with the corresponding optimal operating conditions for the selected substrate pair. In contrast to the screening-based workflow, which evaluates catalyst performance over a fixed set of pre-defined operating conditions, this module is intended to identify the best attainable operating point within the broader operating space considered in the optimization procedure.

Because the selectivity score is evaluated for ordered substrate pairs, and because the study includes ten substrates, a total of 90 possible substrate pairs must be considered. Performing a full genetic algorithm optimization for every pair is computationally demanding. For this reason, the optimization analyses were carried out in advance, and the resulting outputs were stored for later retrieval in the web application. These precomputed optimizations were performed on the Digital Alliance of Canada platform and required nearly two days of computational time. The app therefore functions as an efficient interface that allows the user to access the saved optimization results directly without repeating the full optimization procedure during each session.

As shown in Figure C.10, the first section of the optimization module provides a concise description of the workflow and explains how the user can load the optimization results for the selected substrate pair. Once the user chooses Substrate A and Substrate B from the sidebar and clicks Full optimization, the app retrieves the previously saved genetic algorithm results corresponding to that specific pair. The following sections present the optimization outputs generated through this workflow and explain how these results are displayed and interpreted within the application.

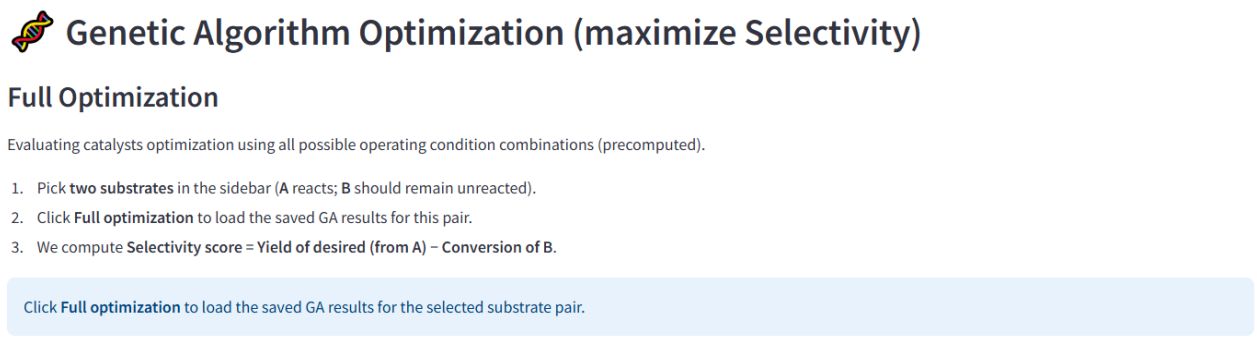


Figure C.10. Introductory section of the Full Optimization module in the web application, presenting the optimization objective and the user instructions for retrieving the saved genetic algorithm results for the selected substrate pair.

Figure C.11 presents the first set of results displayed after the user selects a substrate pair and clicks Full optimization in the sidebar shown previously in Figure C.2. At this stage, the app retrieves the saved optimization results corresponding to the selected pair and summarizes the catalyst ranking in a format similar to that used in the Catalyst Search workflow. This consistency in presentation allows the user to compare the outputs of the two workflows more easily while recognizing that the underlying basis of evaluation is different.

In this section, the application first highlights the overall winner of the genetic algorithm optimization, namely the catalyst associated with the highest selectivity score obtained under optimized operating conditions. The accompanying bar chart then presents all available catalysts together with their own highest achievable selectivity scores from the optimization results. In this way, the figure provides both the identity of the best catalyst and the relative standing of the remaining candidates under the optimization-based framework.

For the selected substrate pair shown in this example, 1% Pt(V)/C is identified as the genetic algorithm winner, reaching a maximum selectivity score of 99%. The remaining catalysts are ranked according to their best optimized scores, which allows the user to assess not only the top-performing candidate but also how close the other catalysts come to the optimum. This is useful because some catalysts may perform competitively under optimized conditions even if they are not selected as the final best candidate.

A comparison with Figure C.5 shows that both workflows identify 1% Pt(V)/C as the best catalyst for this substrate pair, with the same maximum selectivity score of 99%. This agreement suggests that the catalyst identified as optimal under the broader genetic algorithm search is consistent with the best result obtained from the predefined screening conditions. At the same time, some differences can be observed in the ranking of the remaining catalysts. For example, in Figure C.5, 1% Au/TiO_2 and 5% Ir/$CaCO_3$ both achieve 96%, whereas in Figure C.11 their optimized scores become 95% and 97%, respectively, resulting in a slight change in ordering. Similarly, 10% Pd/$CaCO_3$ improves from 22% in the catalyst search to 25% in the optimization results, while Ni-Raney decreases from −17% to −23%. These differences indicate that although the best overall catalyst remains unchanged, the broader operating-space search can slightly modify the relative performance of the other catalyst candidates.

Overall, Figure C.11 serves as the first key output of the Full Optimization workflow. It provides a direct summary of the catalyst ranking based on the optimization results and establishes the best catalyst candidate before presenting the detailed optimal operating conditions and other associated outputs in the following sections.

Figure C.12 presents the best optimization result in a compact visual format. After the optimization results are loaded, the app displays the top catalyst candidate together with the operating conditions associated with the highest selectivity score. In this example, the Sankey plot links the best candidate to four key outputs: the selected catalyst, the optimal temperature, the optimal pressure, the optimal flow rate, and the resulting selectivity score. This provides a clear summary of the single best solution identified for the selected substrate pair.

For the example shown here, the optimal solution corresponds to 1% Pt(V)/C, with an optimal temperature of 85 °C, an optimal pressure of 43 bar, and an optimal flow rate of 2.4 mL/min, yielding a maximum selectivity score of 99%. Presenting the result in this form makes it easy for the user to interpret the optimization outcome at a glance, since both the catalyst identity and the associated operating conditions are displayed simultaneously.

This figure also complements the information presented in Figure C.11. While Figure C.11 compares all catalysts according to their best achievable optimized selectivity scores, Figure C.12 focuses only on the single winning solution and provides the exact operating conditions at which that optimum is obtained. In this way, the user can first identify the best catalyst from the ranking

view and then immediately examine the detailed optimal settings for that catalyst in the Sankey-based summary.

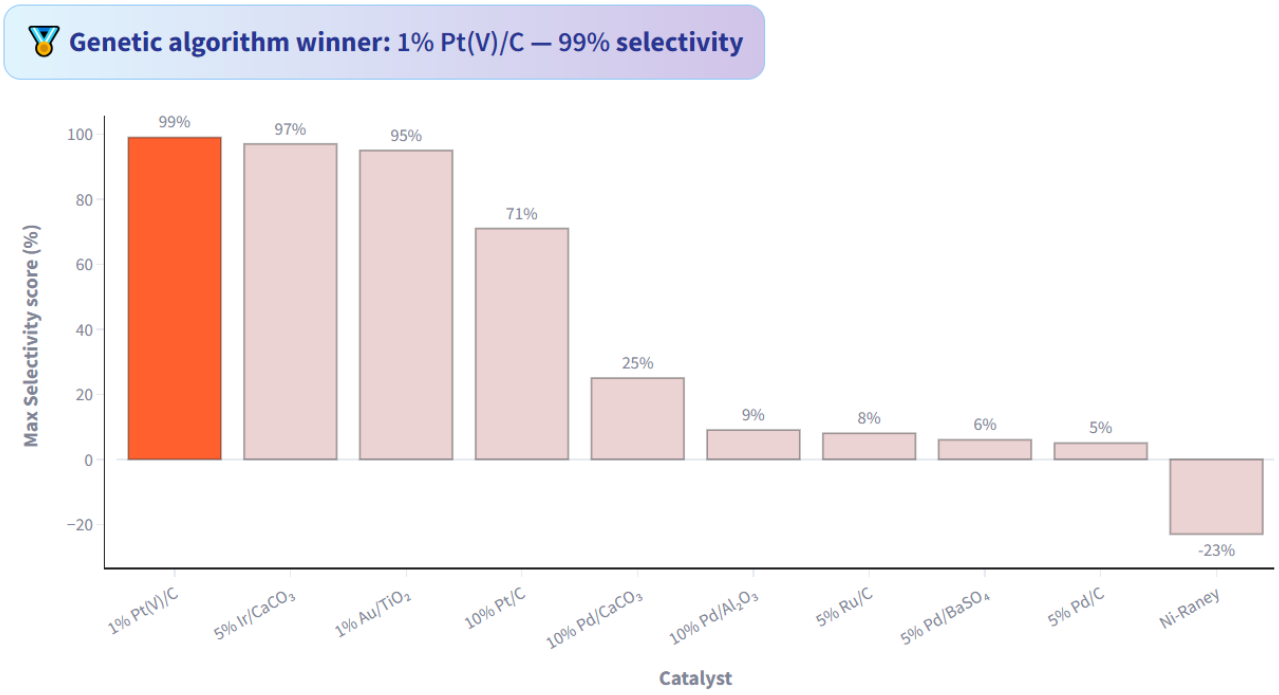


Figure C.11. Initial optimization result displayed in the Full Optimization module, showing the catalyst identified by the genetic algorithm as the best candidate for the selected substrate pair, together with the maximum selectivity score achieved by each catalyst under optimized operating conditions.

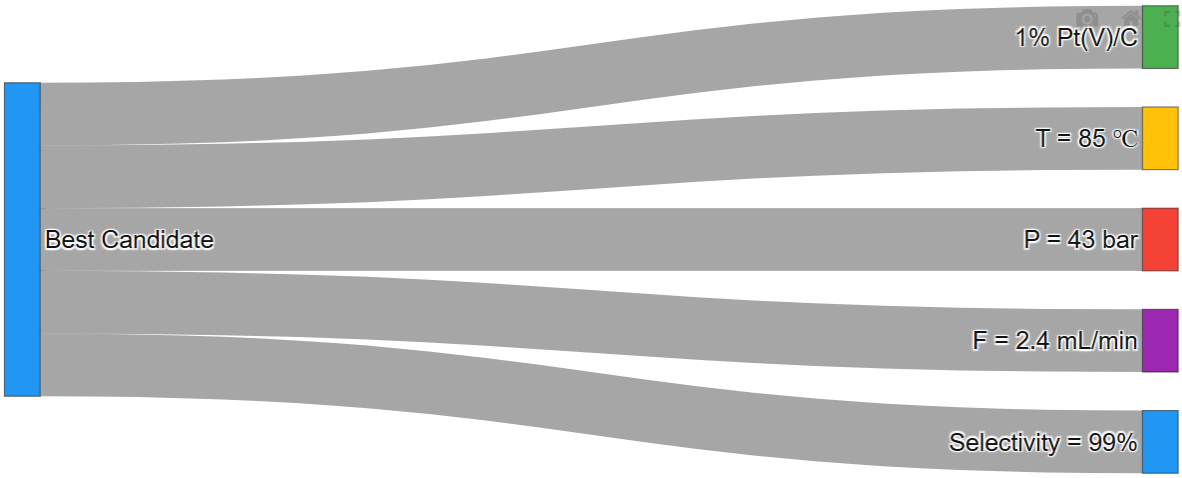


Figure C.12. Sankey-style summary of the best optimization result, showing the catalyst selected by the genetic algorithm together with the corresponding optimal temperature, pressure, flow rate, and maximum selectivity score for the chosen substrate pair.

Figure C.13 provides a broader view of the optimization results by presenting the best operating point obtained for each catalyst rather than focusing only on the single overall winner. For every catalyst, the operating conditions corresponding to its highest selectivity score are extracted from the optimization results and displayed in three separate scatter plots. These plots show how the maximum selectivity achieved by each catalyst is associated with a specific temperature, pressure, and flow rate.

The first panel relates selectivity score to temperature, the second relates selectivity score to pressure, and the third relates selectivity score to flow rate. Together, these three views allow the user to examine how the optimal operating conditions differ among catalysts and whether high-performing catalysts tend to cluster within similar regions of the operating space. This representation is useful because catalysts with similar maximum selectivity scores may still require substantially different operating conditions to reach that performance.

For the selected substrate pair in this example, Figure C.13 shows that the best-performing catalysts are concentrated at relatively high selectivity levels, but their associated optimal operating conditions are not identical. The overall winning catalyst, 1% Pt(V)/C, achieves its maximum selectivity at a high temperature and a comparatively moderate pressure and flow rate, whereas other strong candidates such as 5% Ir/$CaCO_3$ and 1% Au/TiO_2 reach similar selectivity levels under different combinations of pressure and flow rate. At the same time, lower-performing catalysts remain grouped at much lower selectivity values, making their limited competitiveness evident even when their own best operating points are considered.

This figure complements Figure C.12 in an important way. While Figure C.12 isolates the single best catalyst and its corresponding optimal settings, Figure C.13 expands the view to all catalysts and shows the operating conditions associated with each catalyst's individual optimum. As a result, the user can compare not only which catalyst performs best overall, but also how the optimized operating space differs across the full catalyst set.

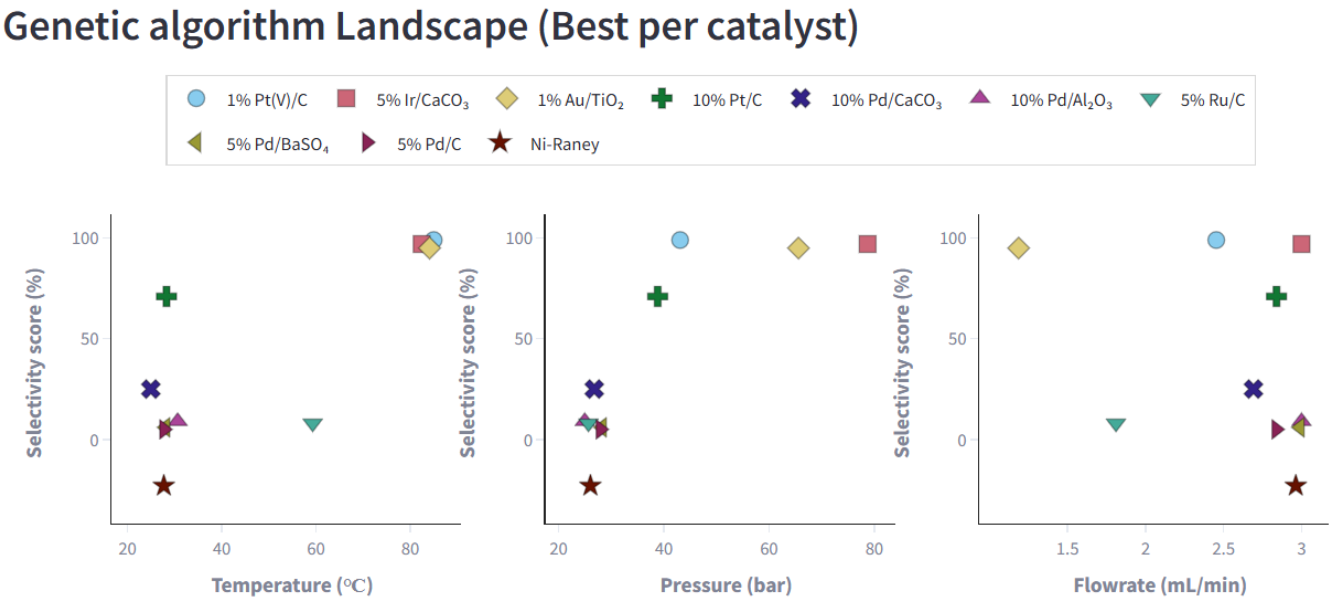


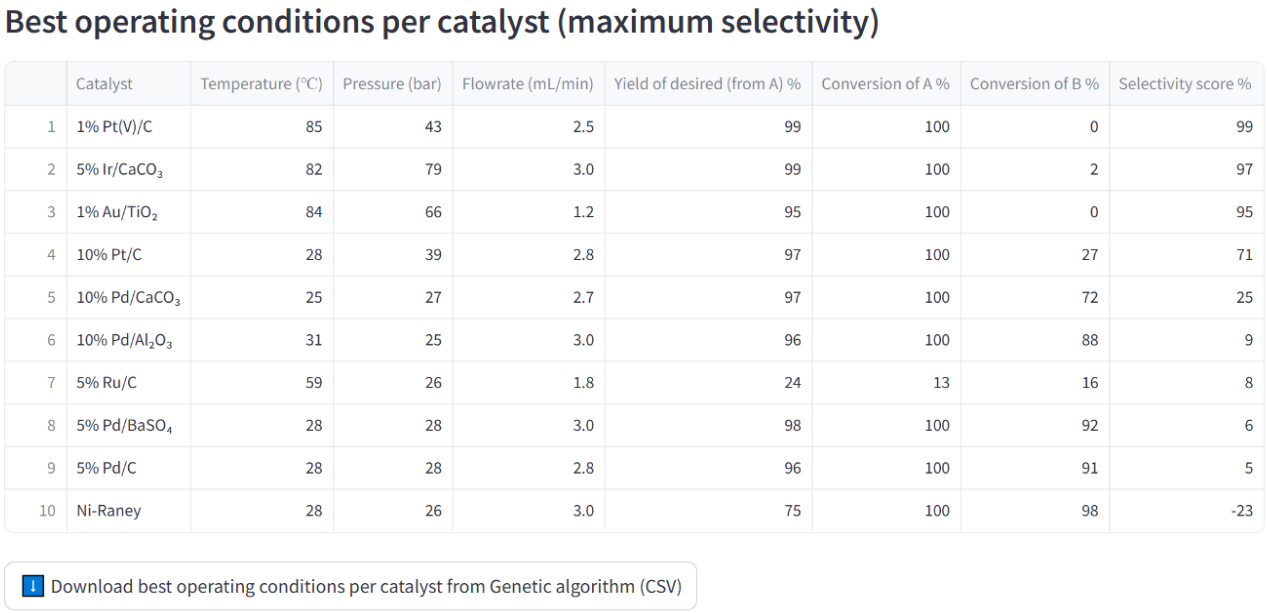
Figure C.13. Genetic algorithm landscape showing, for each catalyst, the operating conditions at which its maximum selectivity score is achieved. The three panels separately display the optimal temperature, pressure, and flow rate associated with the best optimization result for each catalyst.

Figure C.14 presents the final output section of the Full Optimization module. Similar to the results table provided in the Catalyst Search workflow, this section compiles the optimization outcomes into a structured and accessible tabular format. For each catalyst, the table reports the operating conditions at which its maximum selectivity score is achieved, along with the corresponding predicted yield of the desired product from Substrate A, conversion of A, conversion of B, and the resulting selectivity score. In this way, the table provides the full numerical summary underlying the optimization-based results shown in the preceding figures.

This table is particularly useful because it brings together, in a single view, both the performance of each catalyst and the operating conditions required to achieve its individual optimum. While the previous optimization figures provide graphical summaries of the winning catalyst, the best operating point, and the landscape of optimized results, Figure C.14 offers the precise numerical values needed for closer inspection, comparison, and documentation. It therefore serves as the most complete summary of the optimization results for the selected substrate pair.

As in the first part of the app, the results table can also be downloaded by the user. This allows the optimization outputs to be saved and used for further analysis outside the web application,

including comparison across catalyst candidates, external visualization, or incorporation into additional records and reports. Therefore, Figure C.14 represents the final stage of the Full Optimization workflow by providing a complete, exportable, and user-accessible summary of the optimized catalyst results.



Best operating conditions per catalyst (maximum selectivity)

	Catalyst	Temperature (°C)	Pressure (bar)	Flowrate (mL/min)	Yield of desired (from A) %	Conversion of A %	Conversion of B %	Selectivity score %
1	1% Pt(V)/C	85	43	2.5	99	100	0	99
2	5% Ir/$CaCO_3$	82	79	3.0	99	100	2	97
3	1% Au/TiO_2	84	66	1.2	95	100	0	95
4	10% Pt/C	28	39	2.8	97	100	27	71
5	10% Pd/$CaCO_3$	25	27	2.7	97	100	72	25
6	10% Pd/Al_2O_3	31	25	3.0	96	100	88	9
7	5% Ru/C	59	26	1.8	24	13	16	8
8	5% Pd/$BaSO_4$	28	28	3.0	98	100	92	6
9	5% Pd/C	28	28	2.8	96	100	91	5
10	Ni-Raney	28	26	3.0	75	100	98	-23

Download best operating conditions per catalyst from Genetic algorithm (CSV)

Figure C.14. Final tabulated output of the Full Optimization workflow, listing for each catalyst the operating conditions associated with its maximum selectivity score, together with the corresponding predicted yield, conversion values, and selectivity score. The table can also be downloaded by the user for further analysis and recordkeeping.

This appendix has presented the structure and functionality of the developed Streamlit-based web application for catalyst screening and optimization in chemoselective hydrogenation. Through the two integrated workflows, namely Catalyst Search and Full Optimization, the app provides an accessible interface for evaluating catalyst performance under both pre-defined and optimized operating conditions. By combining predictive ML outputs, visual summaries, and downloadable result tables, the application enables users to examine catalyst behaviour, compare candidate catalysts, and identify promising operating conditions for selected substrate pairs. Overall, the web app serves as a practical and user-oriented deployment of the developed framework, translating the predictive and optimization results into an interpretable tool for catalyst evaluation and decision support.