WPI

SIT

Using Machine Learning to Predict Network Performance of a Micro-Structured Gas Separator Using the Soret Effect

A Major Qualifying Project Report

submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

degree of Bachelor of Science

Submitted by:

Charlotte Roscoe

01/26/2025

Project Advisors:

Professors Bashima Islam and Adam Powell

Worcester Polytechnic Institute

Report Submitted to:

Professor Naoki Ono

Shibaura Institute of Technology

ABSTRACT

Hydrogen is a promising clean energy source, emitting no CO2 during use. However, production methods rely on fossil fuels, contributing to carbon emissions. To combat this, the Soret Team at the Shibaura Institute of Technology in Tokyo is investigating hydrogen separation using a thermophoretic microstructured separator device. Alternatively, this project simulates using multiple smaller devices in a network structure, employing a Random Cut Forest machine-learning regression model to predict separation. A custom GUI was developed to visualize and interact with this network. Results showed potential benefits in scalability. Future work involves replacing synthetic datasets with experimental data. This project advances clean energy and highlights the benefits of interdisciplinary research.

ACKNOWLEDGEMENTS

The author would like to thank those who made this project possible.

To start, the author would like to thank their host professor, Professor Naoki Ono of the Shibaura Institute of Technology, for his guidance and kind hospitality.

Furthermore, the author would like to thank the Soret Team at SIT, Ryo Masuo, Soichiro Nakanishi, Mimori Osada, and Mizuki Yamazaki, for their continuous assistance and patience throughout the term. Their help was crucial to the completion of this project.

Finally, the author would like to thank the advisors for this project, Professors Bashima Islam and Adam Powell. Without Professor Islam's guidance in machine learning and Professor Powell's knowledge of Mechanical Engineering, this project would not have been possible.

EXECUTIVE SUMMARY

Hydrogen has drawn attention as a valuable fuel, notable for its ability to produce energy without generating CO_2 emissions. However, current hydrogen production primarily relies on fossil fuels, contributing to significant annual carbon emissions. To address this challenge, the Soret Team at the Shibaura Institute of Technology (SIT) is optimizing a micro-structured gas separator device that uses the Soret effect to promote hydrogen separation.

The author's project focused on modeling a network of smaller microstructured devices as opposed to one large device. This network was simulated using machine learning regression models trained on experimental and simulated data from the SIT Laboratory. The author achieved the following objectives for the completion of this goal:

(1) Preprocess, compile, and analyze data

(2) Train and tune machine learning models to predict separation concentration

(3) Model interconnected device network in a Graphical User Interface

Random Cut Forests (RCFs) were identified as the most effective regression model, achieving better prediction accuracy than other tested models. This is due to their ability to efficiently capture complex patterns in high-dimensional data while maintaining robustness against noise and outliers. RCFs outperformed other models such as decision trees and linear regression models, having a lower overall mean squared error (MSE). The final software creates a binary tree structure of separation results based on input parameters, such as initial starting concentration and flow rate.

Flow Rate (ml/min)	One Device Simulation Results	Network Simulation Results
30	26.0%	31%
60	26.2%	32%
90	26.1%	32%
120	26.2%	32%
150	26.2%	32%
180	26.5%	36%

Figure 1: One Device Simulations vs Network Simulations

Results show increased separation percentages in the network of devices versus the older device of the same length, as seen above in Figure 1. These indicate that a network of devices may achieve greater separation efficiency than a single device, particularly at higher flow rates.

Despite these results, the project faced limitations due to a lack of experimental data. To combat this, the author used a synthetic dataset, which introduces a degree of uncertainty in the results. Furthermore, due to data constraints, the software can only simulate an outdated device iteration.

To conclude the report, the author recommends replacing the synthetic dataset with experimental data from real devices to enhance accuracy. If the results prove favorable, the SIT team could implement and test this network of devices further. This project demonstrates the value of combining mechanical engineering and computer science to advance clean energy innovation.

TABLE OF CONTENTS

ABSTRACT....ii
ACKNOWLEDGEMENTS....iii
EXECUTIVE SUMMARY....iv
TABLE OF CONTENTS....vi
TABLE OF FIGURES....vii
1.0 Introduction....1
2.0 Background....2
2.1 The Soret Effect....2
2.2 The Current Gas Separator Device....3
2.3 Regression Models to Consider....4
2.3.1 Linear Regression....4
2.3.2 Decision Trees and Random Cut Forests (RCFs)....4
2.3.3 Neural Networks....4
2.3.4 Summary....5
2.4 Previous Research of Regression Model Use for Prediction....5
3.0 Methods....6
3.1 Objectives....6
3.2 Datasets....7
3.2.1 Laboratory Data....7
3.2.2 Synthetic Data....8
3.3 Using Machine Learning Models to Predict Network Separation....10
3.4 Visualizing the Network through a Graphical User Interface (GUI)....11
3.5 Challenges & Limitations....12
4.0 Final Software and Results....13
5.0 Recommendations, Future Work, and Conclusion....13
5.1 Recommendations and Future Work....14
5.2 Conclusion....14
BIBLIOGRAPHY....15
APPENDICES....16
Appendix A: Timeline of Project Work....16
Appendix B: Data Extrapolation Code (Python)....17
Appendix C: GitHub Repository Link....19

TABLE OF FIGURES

Figure 1: One Device Simulations vs Network Simulations……iv

Figure 2.1: The Soret Effect……..…..2

Figure 2.2: The Soret Device's Microstructure.…….……3

Figure 2.3: The Pros and Cons of each Model……...5

Figure 3.1: Data Preprocessing Flowchart……...……..7

Figure 3.2: Additional Simulations provided by SIT Laboratory……..8

Figure 3.3: Simulations with the old dataset (top) vs. the synthetic dataset (bottom).....9

Figure 3.4: Performance of Four Machine Learning Models..……..……...10

Figure 3.5: Modeling a Device Network……..……10

Figure 3.6: Software GUI, Windows version……...11

Figure 3.7: Software GUI, Error Message, Mac version……..11

Figure 4.1: Network Program in Action, Mac version…….13

Figure 4.2: One Device Simulations vs Network Simulations…….....13

1.0 Introduction

With the current rising concerns surrounding climate change and global warming, hydrogen has been drawing attention as a valuable fuel. It is notable in the sense that when used exclusively as an energy source, it does not produce CO_2 emissions, therefore providing a promising lead to reducing our carbon footprint. However, currently, the creation of hydrogen is quite energy intensive, the production of which involves mainly fossil fuels such as coal and natural gasses. This means the production of hydrogen is still responsible for significant annual carbon emissions (IEA, n.d.). Obtaining cheap, efficient, and specifically clean hydrogen is an ongoing research topic, one which the Soret Team at the Shibaura Institute of Technology (SIT) in Tokyo, Japan is currently investigating.

The Soret Team (Ryo Masuo, Soichiro Nakanishi, Mimori Osada, Mizuki Yamazaki) at SIT is focusing on optimizing the microstructure of a thermophoretic gas separator device to accomplish this task. The device itself functions via the application of a thermal gradient, thereby utilizing the Soret effect to drive hydrogen separation. The microstructure within the device is specifically designed to guide the gas flow for optimal separation concentrations.

While the current experiment functions by use of one larger device, the author investigated the use of multiple smaller devices in a network structure to improve separation via simulation. To achieve this, this project used the data obtained from the larger device and simulation software to train a regression model, which was then used to attempt predictions of the separation percentage of each smaller device in a network. The author modeled these connections between devices in an easy-to-use GUI, with the intent to visualize the network for holistic analysis. This program is meant to provide an understanding of the potential advantages of this structure, give insight into overall system performance, and possibly provide a quicker prediction tool in comparison to standard simulation software. This report concludes with recommendations for future work, mainly centering on improving the underlying dataset powering the software.

This project not only highlights the benefits of interdisciplinary research, but also represents efforts toward cleaner energy. The success of cleaner and more efficient hydrogen separation could pave the path for advancements in sustainable energy practices, potentially enabling more efficient fuel production, reducing reliance on fossil fuels, and contributing to the global shift toward renewable energy solutions.

2.0 Background

Before detailing the methods of this project, the background section elaborates further on the Soret effect, explains in detail the use of the current gas separator device, and discusses possible regression models for project use. Finally, it investigates past research into relevant areas relating to the project, specifically the use of machine learning for separation predictions.

2.1 The Soret Effect

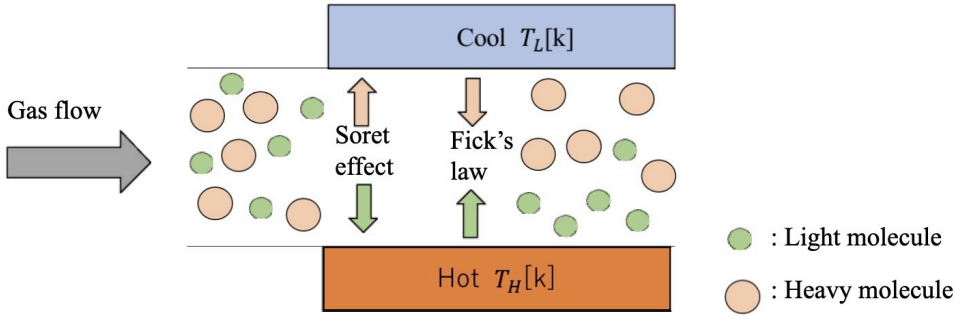

Figure 2.1: The Soret Effect (SIT Laboratory, n.d.)

The main force behind the particle separation within the device is the Soret effect. This effect occurs when temperature differences are applied to a fluid mixture, where it generates a difference in concentration– lighter molecules move towards the hotter temperature, while heavier molecules move towards the cooler temperature. Another force impacting mass flux in the device is Fick's diffusion law. When both fluxes reach a balance, a steady state is achieved. A day-to-day example of the Soret effect can be seen in lava lamps: at the bottom of the lamp, heat is applied to the liquid within it, causing the liquid to rise to the top. Once the liquid reaches the top and cools again, it floats back down to the bottom of the device. These are the same forces that govern the experimental device. A visual representation of the Soret effect is pictured above in Figure 2.1.

2.2 The Current Gas Separator Device

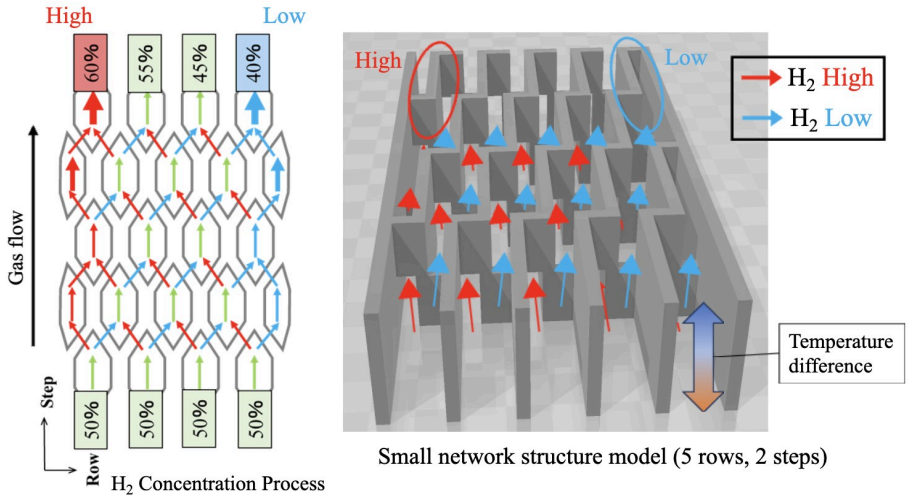

Figure 2.2: The Soret Device's Microstructure (SIT Laboratory, n.d)

The Soret Team at the Shibaura Institute of Technology is optimizing a microstructured gas separator device that uses this Soret effect to promote particle separation. The experiment begins with a gas mixture of 50% Hydrogen and 50% Carbon Dioxide, which is funneled into the device via a tube. On one side of the device, a heat of between 116°-127° Celsius is applied, while the 'cool' side applies between 16°-27° Celsius. An exact temperature difference of 100° Celsius is maintained between the sides to ensure that variations in separation efficiency are attributed primarily to the microstructure design rather than fluctuations in thermal conditions. The gas mixture flows through the device, with separation being promoted through the microstructure within it, pictured in Figure 2.2. The microstructure 'guides' the particles diagonally in both horizontal and vertical directions. At the opposite end of the device, there are 3 tubes for 'high', 'medium', and 'low' concentrations where the mixture is funneled out and measured. The device has undergone multiple improvements to its microstructure over time, with the most recent device iteration achieving a maximum concentration separation of 15.7%. Compared to traditional gas separation technologies such as membrane separation and pressure swing adsorption, this microstructured gas separator offers a more energy-efficient approach by utilizing the Soret effect rather than relying on high-pressure or cryogenic processes.

2.3 Regression Models to Consider

Below the author will detail some of the most commonly used regression models, along with their benefits and drawbacks. The goal of this section is to provide insight into which models were considered for this project.

2.3.1 Linear Regression

Linear regression is the most commonly used regression algorithm in predictive analysis due to its simplicity. This model attempts to fit a line that best represents the scatter plot of the given data to draw predictions (Gupta et al., 2017). The benefits of this approach include its simplicity, agility, and the fact that it is easy to understand. As the name implies, the drawbacks to this method are mainly that it is less effective on non-linear data, which means it is not as effective at understanding more complex relationships.

2.3.2 Decision Trees and Random Cut Forests (RCFs)

Decision trees create ‘branches’ at each feature decision where the model learns the relationships between them. At the end of each branch is a ‘leaf’, at which point the decision tree presents a prediction. Variations of this algorithm include Random Cut Forests, where many trees are created and compiled, further improving the final accuracy (Amazon Web Services, n.d.). Its pros include once again its simplicity, however, single decision trees are prone to overfitting. While multiple trees are usually more accurate, they are more difficult to conclude from. RCFs were predicted to be the most suitable model early on due to their ability to handle high-dimensional data, detect anomalies, and reduce overfitting compared to single decision trees. This approach improves stability and accuracy, making them a strong candidate for capturing complex patterns in the dataset.

2.3.3 Neural Networks

Neural Networks are more complex algorithms that make use of neurons, similar to modeling neurons in the brain (Amazon Web Services, n.d.). These neurons are connected by edges, similar to synapses. All of these neurons send signals to each other, with computations being performed at each neuron, termed the activation function. Each signal also has a weight associated with it, which is constantly updating. The benefits of neural networks are vast, as it can understand complex relationships within data. However, to make use of such a system, very large amounts of data are required. Additionally, neural networks tend to overfit. The

time to train this type of model is also quite long and requires significant computing power. Finally, the results from this model are generally more difficult to understand.

2.3.4 Summary

Below is a summary of the benefits and drawbacks of each discussed model.

Model	Pro	Con
Linear Regression	Simple, easy to understand, fast computation	Not effective for non-linear relationships
Decision Trees	Intuitive, easy to interpret	Prone to overfitting
Random Cut Forests	Handles high-dimensional data, reduces overfitting, good for anomaly detection	Less interpretable due to multiple trees
Neural Networks	Can model complex relationships, powerful for large datasets	Requires large data, long training time, prone to overfitting, hard to interpret

Figure 2.3: The Pros and Cons of each Model

2.4 Previous Research of Regression Model Use for Prediction

Finally, this chapter closes by investigating an example of the use of regression models in similar research. In an article published in the International Journal of Coal Preparation and Utilization titled ‘Comparison of the prediction performance of separating coal in separation equipment using machine learning based cubic regression modelling and cascade neural network modelling’, Bharath Kumar Shanmugam et al. tackle a similar topic: Predicting the separation efficiency of a substance within a structured separation device. In their scenario, it was the prediction of the separation efficiency of coal through a mesh vibrated at certain frequencies via the use of cubic regression and neural network modeling. This method proved to be very successful in their case– the article describes that the R-squared value for their cubic regression model achieved 97.7%, while the neural network was even more successful with an R-squared value of 99.92% (Shanmugam et al., 2022). While their study focuses on coal rather than hydrogen, the methodology they employ is highly relevant to hydrogen separation. Both processes involve predicting the efficiency of separation within a structured system, where machine learning models can play a critical role in improving accuracy.

3.0 Methods

The methods chapter details the tools that the author used to complete the overall project goal and objectives, which will be described within the chapter. In summary, this project made use of data processing and machine learning models to create a device network prediction software, visualizing it in a Graphical User Interface (GUI). The chapter closes with an additional note on the limitations and challenges this project faced.

3.1 Objectives

Before listing the methods of this project, it is necessary to state the project goal:

"To optimize hydrogen gas separation by use of a simulated network of smaller microstructured devices as opposed to one large device."

The author identified the following objectives as vital to the completion of this goal:

- Preprocess, compile, and analyze data
    - Prepare data for use by the ML models
- Train and tune machine learning models to predict separation concentration
    - Determine which model predicts most accurately
- Model interconnected device network in a Graphical User Interface
    - To create a better overview of network performance

These objectives contributed towards creating a program to predict the separation of a network of devices, to determine if this network will produce better hydrogen separation than a larger device of the same length. Project work took place almost exclusively at the WPI Japan Project Center in Tokyo, Japan, with off-shore work being limited to final report completion. Additionally, all work was explicitly computer-based, as the author did not personally execute experiments, and solely relied on data supplied by the Soret Team and simulation software. The full timeline for this project can be found in Appendix A.

3.2 Datasets

3.2.1 Laboratory Data

The first step was preprocessing and compiling the data received from the various sources from the SIT Laboratory. At first, the two sources were an Excel spreadsheet of experimental results, and a PDF document containing the dimensions of the microstructure of each generation of device over the years. The Excel spreadsheet data contained information recorded at each experiment, including the flow rate, simulation predictions from software at the laboratory, the actual experimental result (resulting hydrogen concentration difference), and the percent success rate (how well the actual result compared to the simulated result). The data was compiled into a CSV file for use by the machine learning model. This step also included filtering out redundant and incomplete columns in the data set, for example, the name of each experiment. At the end of preprocessing and before the addition of simulation data, the dataset contained 115 rows and around 9 features, from 8 variations of the microstructure. Next, a Principal Component Analysis (PCA) was applied to reduce the dimensionality of the data before using it to train the regression model. PCA reduces dimensionality by transforming the data into a set of uncorrelated components that capture the most significant variance in the dataset. This reduction in complexity helps to minimize noise and prevent overfitting, improving the efficiency and accuracy of the model during training. Below is a flowchart depicting the general flow of preliminary preprocessing.

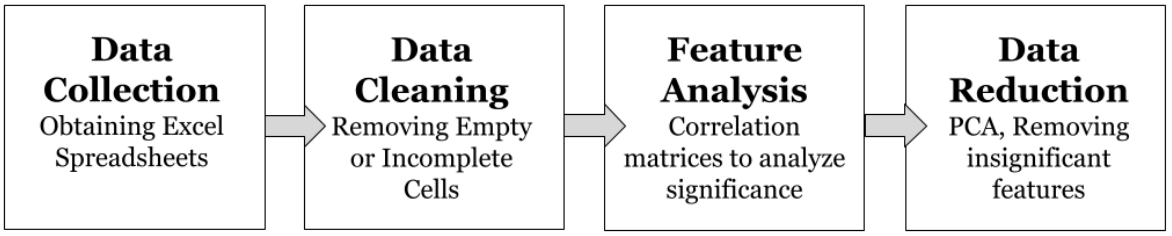

Figure 3.1: Data Preprocessing Flowchart

As time progressed, the goal of this project's work morphed from simply predicting the separation of the devices to the current project objective, modeling a network of smaller separators. With this change, the data also required some modification to more accurately service the project's needs.

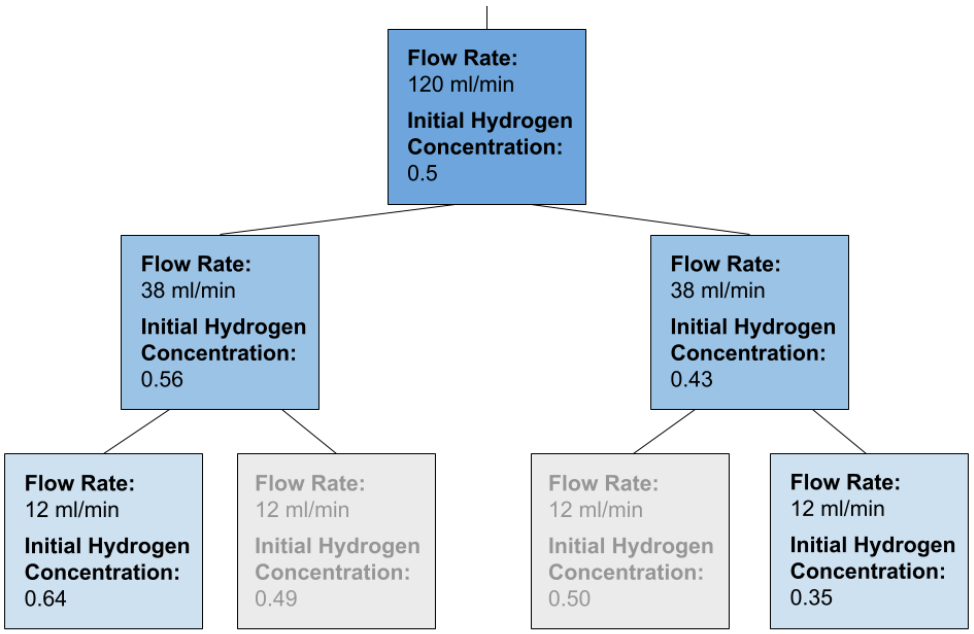

Figure 3.2: Additional Simulations provided by SIT Laboratory (in blue)

As mentioned previously, the SIT Laboratory had only completed experiments on starting concentrations of gas at exactly 50% Hydrogen, and 50% Carbon Dioxide. When simulating a network of devices, the subsequent device would operate on a different starting concentration than 50/50, which was not represented in the previously available data. Additional simulations were required to support this change, which the Laboratory at SIT completed on the author's behalf. Five additional simulations were conducted, of various starting concentrations, as pictured above in Figure 3.2. The starting concentrations for the simulations were taken from the previous network device's ending concentration for optimal accuracy in prediction.

Additionally, since the author needed to only model a single device of shorter length for the network, the dataset of actual usable data points was reduced significantly. Only simulation data from a single iteration of the device could be used, and since the device had to be of shorter length, the ending concentration had to be estimated at a shorter length as well. One benefit of this change was the reduction in dimensionality, as device dimensions no longer needed to be considered. However, the resulting data was no longer enough to use for prediction, which leads to this project's next chapter.

3.2.2 Synthetic Data

Because of the changes in the project's direction, the usable data was reduced significantly. At this point, the author had data from one experimental device version, and

five simulations of shorter device lengths. To combat the data shortage and provide more accurate results, the author created a synthetic dataset from existing data for temporary use by the program until the SIT Laboratory decides to replace it with actual experimental data. There were three issues with the previously available data: it was of a device length that was too long, only contained 50%/50% starting concentrations, and only contained the most extreme values recorded, when the simulated network required average values of certain sections.

To solve the first problem, the author used a previous full device simulation, which simulated each cell in the microstructure, determined a best-fit line for the data points, projected it onto the second set of data points, and then calculated the separation concentration at a shorter length (x-value) of the device. The author repeated this for each previous simulation. To solve the second and third problems, the author calculated the relative change in the simulation data for both the starting concentrations and differences between average and extreme values, and used these values to extrapolate the dataset. To accomplish this, the author wrote a Python script to perform the calculations on the .CSV file, which can be found in Appendix B.

The final dataset brings the base set of five rows to a total of 216 rows, having four columns: flow rate, starting concentration, left output, and right output. Due to the predictive nature of the data, it is not entirely reliable, which will be discussed in the 'Challenges & Limitations' Chapter.

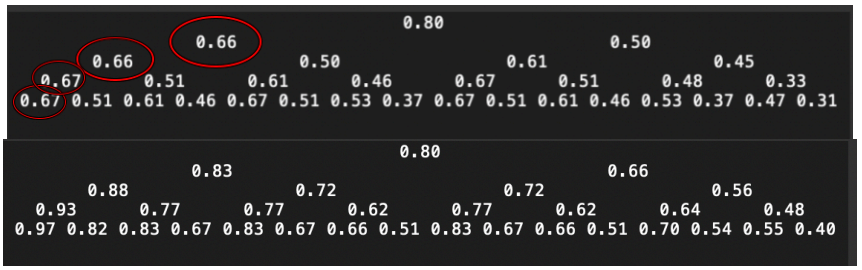

Figure 3.3: Simulations with the old dataset (top) vs. the synthetic dataset (bottom)

Despite its limitations, the synthetic dataset had a significant impact on prediction quality. Figure 3.3 compares results at the same flow rates (120 ml/min). As seen circled above, without the necessary data to make predictions for different starting concentrations, the model makes visible mistakes, such as the model predicting the same "left" output consecutively.

3.3 Using Machine Learning Models to Predict Network Separation

The next step was to train machine learning models with the data to attempt the prediction of the separation concentration of a single device. Five separate models were tested, Linear Regression, Decision Trees, Random Cut Forests, SVR, and Gradient Boosting. Linear Regression performed the worst, having the highest MSE, and was quickly cut from tests.

```
Decision Tree MSE (Split 1): 0.00029923776041666637
Random Forest MSE (Split 1): 0.00030052661003896245
Gradient Boosting MSE (Split 1): 0.000299201237408748 3
SVR MSE (Split 1): 0.00023425000000000003
Decision Tree MSE Percentage (Split 1): 1.62%
Random Forest MSE Percentage (Split 1): 1.62%
Gradient Boosting MSE Percentage (Split 1): 1.62%
SVR MSE Percentage (Split 1): 1.27%
Decision Tree MSE (Split 2): 0.001844165798611112
Random Forest MSE (Split 2): 0.00182491261551307
Gradient Boosting MSE (Split 2): 0.001843912870738888
SVR MSE (Split 2): 0.0021439166666666672
Decision Tree MSE Percentage (Split 2): 9.97%
Random Forest MSE Percentage (Split 2): 9.87%
Gradient Boosting MSE Percentage (Split 2): 9.97%
SVR MSE Percentage (Split 2): 11.59%
Decision Tree MSE (Split 3): 0.0015103478472222223
Random Forest MSE (Split 3): 0.0014585170661799286
Gradient Boosting MSE (Split 3): 0.001510263799849374
SVR MSE (Split 3): 0.0013362000000000003
Decision Tree MSE Percentage (Split 3): 8.17%
Random Forest MSE Percentage (Split 3): 7.89%
Gradient Boosting MSE Percentage (Split 3): 8.17%
SVR MSE Percentage (Split 3): 7.22%
```

Figure 3.4: Performance of Four Machine Learning Models

From the author's testing, the Random Cut Forests typically outperformed other models, with a smaller Mean-Squared Error overall, and was selected as the final model for the network prediction software. The author used the free-to-use Python library 'sci-kit-learn' to work with the models, using a 90/10 train/test split, splitting the data 3 times each to achieve a better understanding of results, as seen above in Figure 3.4.

```
                                0.50
                0.56                                0.44
        0.65                0.35                0.54                0.46
    0.69        0.31    0.54        0.46    0.61        0.39    0.54        0.46
0.72 0.28 0.54 0.46 0.61 0.39 0.54 0.46 0.68 0.32 0.54 0.46 0.61 0.39 0.54 0.46
```

Figure 3.5: Modeling a Device Network (with outdated data)

After training the model to predict device network separation based on the beginning flow rate and hydrogen concentration, the author set up the network in a perfect binary tree structure, pictured above in Figure 3.5. The model executes the prediction based on the starting concentration and flow rate of the parent node, and the left and right child nodes each

store the left and right output concentrations and flow rates. The child nodes then each become parent nodes, repeating the process.

3.4 Visualizing the Network through a Graphical User Interface (GUI)

Finally, the author created a Graphical User Interface (GUI) for the program. The goal of this step was to provide a holistic overview of the network, give insight into system-wide performance, and provide a more accessible version of the program.

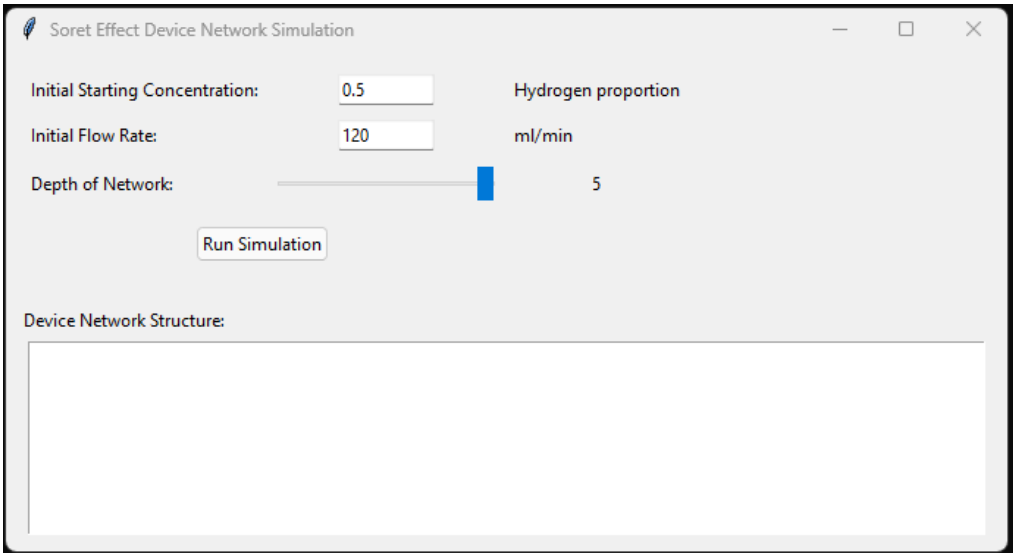

Figure 3.6: Software GUI, Windows version

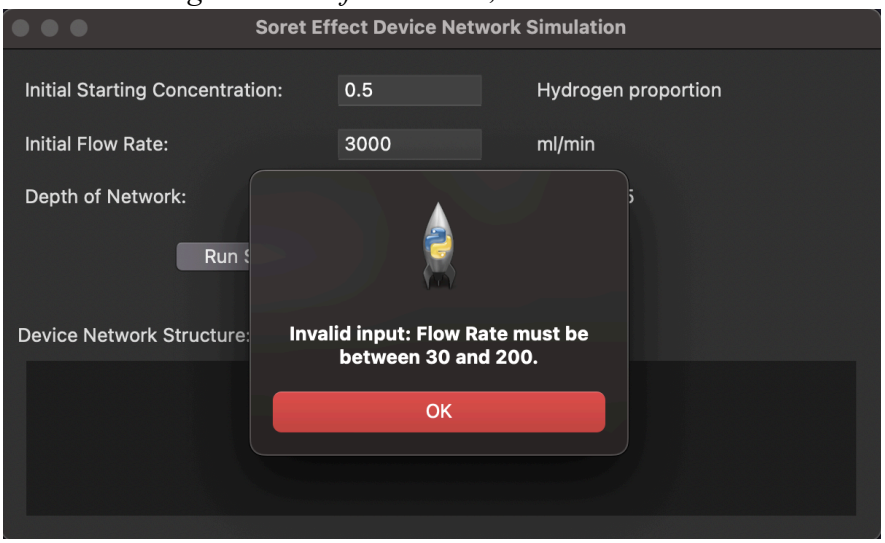

Figure 3.7: Software GUI, Error Message, Mac version

The author used the free-to-use Python library 'Tkinter' for this task, creating a simple GUI that allows the user to modify the initial starting concentration, flow rate, and depth of the network for prediction. Using another Python library, 'PyInstaller', the author

compiled the program into a single executable for ease of use by the SIT team. User testing by the student team at SIT resulted in feedback that the program is “easy to use”.

3.5 Challenges & Limitations

The biggest limitation of this project was the amount of data available. Execution of the experiment is time-consuming and as such only a limited amount of data had been collected. Similarly, large device simulations take from days to a week to complete depending on the version of the device, meaning only a handful of additional simulations could be completed for the sake of the project. Additionally, as mentioned previously, the data that was available required some modifications. Additional simulations were used to create a synthetic dataset to fill these gaps, however, as it is only unverified estimated data, there is a degree of uncertainty about its accuracy, and the author does not believe it is reliable in its current state. The additional simulations could also only be performed on the fastest iteration of the device, which also happened to be the most outdated version. This means the software is only able to be used to simulate a network of these outdated devices. This device also behaves differently than its successors, having a different internal structure. It is also worth noting that the synthetic data used by this network prediction program is based solely on simulations. According to past results from the SIT Lab, there is typically still a considerable difference between the simulated data and the results from actual experiments. Lastly, machine learning as a tool is not as accurate as scientific models that rely on calculations, and its results should always be considered with caution.

4.0 Final Software and Results

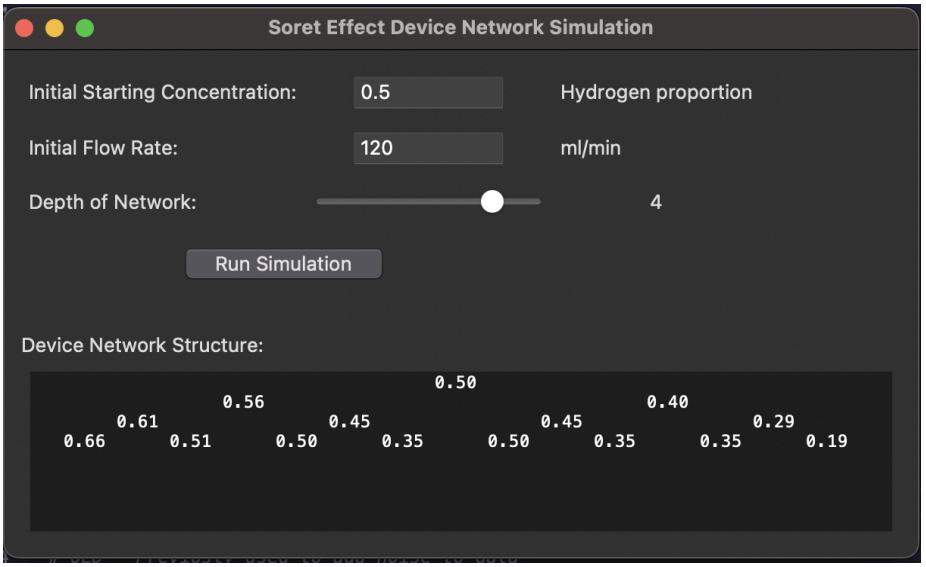

Figure 4.1: Network Program in Action, Mac version

As pictured above and described previously, the final software uses a simple GUI through the Python library 'Tkinter'. There are three editable fields; Initial Starting Concentration, Initial Flow Rate, and Depth of Network. Clicking 'Run Simulation' creates the binary tree structure of separation results. Changing the fields above and clicking 'Run Simulation' will alter the resulting device network structure based on the parameters given. The software is compiled via the Python library 'PyInstaller' into an easy-to-use executable, allowing the SIT lab to simply open the software without interacting with any code.

Flow Rate (ml/min)	**One Device Simulation Results (high - low)**	**Network Simulation Results (high - low)**
30	26.0%	31%
60	26.2%	32%
90	26.1%	32%
120	26.2%	32%
150	26.2%	32%
180	26.5%	36%

Figure 4.2: One Device Simulations vs Network Simulations (initial H proportion of 0.5, depth of network: 3)

When comparing the results of this software with previous device simulations, the network simulation software shows greater separation, measured as the difference between the highest and lowest value recorded, as pictured above in Figure 4.2. This could point towards the benefits of a networked approach over one larger device. The network simulation also follows the simulation's general trend of higher differences for higher flow rates. A link to the GitHub repository containing the full program can be found in Appendix C.

5.0 Recommendations, Future Work, and Conclusion

This chapter details the author's recommendations for possible future work and closes the report with conclusions on the project as a whole.

5.1 Recommendations and Future Work

As described in the ‘Challenges & Limitations’ chapter, due to the nature of the data being used, the software’s results should be considered with a degree of caution. The author’s recommendation for future work would be to replace the synthetic dataset with actual data from real experiments. To mitigate the limitations associated with relying on synthetic data, a more robust simulation pipeline could be developed to boost the quantity of validated data available. This approach could help improve the simulation's accuracy and reliability over time. If the team at SIT considers those results favorable, they could also potentially implement this network of devices in real experiments for further testing.

5.2 Conclusion

During two months of project work, with kind assistance from the students at the Shibaura Institute of Technology, the author used concepts of data science and machine learning to create software that simulates a network of microstructured devices for the use of Professor Naoki Ono’s Laboratory at the Shibaura Institute of Technology. The resulting software, despite its shortcomings, is a testament to the benefit of interdisciplinary research, fostering further collaboration. The insights gained from this project can serve as a foundation for refining the software and exploring new applications, advancing innovation at the intersection of Mechanical Engineering and Computer Science.

BIBLIOGRAPHY

Amazon Web Services. (n.d.) What is RCF? *Amazon Web Services Documentation.* https://docs.aws.amazon.com/quicksight/latest/user/what-is-random-cut-forest.html

Amazon Web Services. (n.d.). What is a neural network? *Amazon Web Services.* https://aws.amazon.com/what-is/neural-network/

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. (2013). Regression Models. In L. Fahrmeir, T. Kneib, S. Lang, & B. Marx, *Regression* (pp. 21–72). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-34333-9_2

Gupta, A., Sharma, A., & Goel, A. (2017). Review of Regression Analysis Models. *International Journal of Engineering Research & Technology (IJERT)* https://www.academia.edu/download/83419421/review-of-regression-analysis-models-IJERTV6IS080060.pdf

IEA (n.d). *Hydrogen*. (n.d.). IEA. Retrieved October 30, 2024, from https://www.iea.org/energy-system/low-emission-fuels/hydrogen

Shanmugam, B. K., Vardhan, H., Raj, M. G., Kaza, M., Sah, R., & Hanumanthappa, H. (2023). Comparison of the prediction performance of separating coal in separation equipment using machine learning based cubic regression modelling and cascade neural network modelling. *International Journal of Coal Preparation and Utilization*, *43*(2), 248–263. https://doi.org/10.1080/19392699.2022.2040492

SIT Laboratory, Masuo, R., Nakanishi, S., Osada, M., Yamazaki, M., Ono, N (n.d.). Performance Improvement of Gas Separator with Micro-Network Structure and Prototyping of the Device with a 3D printer. *Shibaura Institute of Technology (SIT)*

APPENDICES

Appendix A: Timeline of Project Work

Week (2024)	Objective
Week 1: (Oct 21 - Oct 25)	● Coordinated available data with Professor Naoki Ono's Laboratory ● Cleared up any uncertainties regarding data ● Preprocessed Data
Week 2: (Oct 28 - Nov 1)	● Utilized PCA for dimensionality reduction of data ● Applied Regression Models to evaluate prediction accuracy based on available data (Decision Tree, Random Cut Forests) ● Observed Experiments
Week 3: (Nov 4 - Nov 8)	● Created and analyzed correlation matrix between data points ● Investigated simulation software at the Laboratory to determine fitness to supply additional data ● Tested Regression Model on Fewer data points for the possibility of using exclusively the newest microstructure rows for training and testing
Week 4: (Nov 11 - Nov 15)	● Researched and tested additional models to determine which gives optimal results (SVR, Gradient Boosting) ● Further Observation of Experiments
Week 5: (Nov 18 - Nov 22)	● Finalized plan for simulations to generate additional data ● Analyzed Mean-Squared-Error and MSE Percentage of Models ● Preliminary Testing
Week 6: (Nov 25 - Nov 29)	● Created simulation program of smaller device network (without GUI) ● Created synthetic dataset for more accurate testing
Week 7: (Dec 2 - Dec 6)	● Improved synthetic dataset ● Improved Python program for more accurate simulation ● Created preliminary GUI
Week 8: (Dec 9 - Dec 13)	● Created Final Iteration of synthetic dataset ● Finalized GUI ● Created executable for easier use of the program by SIT Lab

Appendix B: Data Extrapolation Code (Python)

```python
import pandas as pd
# Read the input CSV file
df = pd.read_csv("File.csv", delimiter=',')
# Create a list to hold the new data
new_data = []

# Iterate through each row in the DataFrame, performing forward and
reverse calculations
# This process produces some duplicates, which were removed afterwards
# Left_Out_perc - 'High' hydrogen end prediction of device
# Right_Out_perc - 'Low' hydrogen end prediction of device
for index, row in df.iterrows():

    # Base value to extrapolate
    starting_val = row["Left_Out_perc"]
    # Determining the relative change in the dataset between the
starting and ending concentrations
    increment = row["Left_Out_perc"] - row["Starting_Concentration"]

    # Append the original row to the new data
    new_data.append({
        "Flow_Rate_ml_min": row["Flow_Rate_ml_min"],
        "Starting_Concentration": row["Starting_Concentration"],
        "Left_Out_perc": row["Left_Out_perc"]
    })

    # Forward calculation
    forward_starting_val = starting_val
    while (forward_starting_val + increment) < 1:
        new_row = {
            "Flow_Rate_ml_min": row["Flow_Rate_ml_min"],
            "Starting_Concentration": forward_starting_val,
            "Left_Out_perc": forward_starting_val + increment
        }
        new_data.append(new_row)
        forward_starting_val += increment

    # Reverse calculation
    reverse_starting_val = starting_val
    while (reverse_starting_val - increment) > 0:
        new_row = {
            "Flow_Rate_ml_min": row["Flow_Rate_ml_min"],
            "Starting_Concentration": reverse_starting_val - increment,
            "Left_Out_perc": reverse_starting_val
```

```
        }
        new_data.append(new_row)
        reverse_starting_val -= increment

# Defining the function that calculates Right_Out_perc based on
Flow_Rate_ml_min (x) and Left_Out_perc (y)
# -> Since flow rate was what determined the separation efficiency, and
since the separation was asymetrical,
# -> the author created another best-fit line based on existing data,
and then applied it to the Left_Out_Perc
# -> collumn to determine the 'Right_Out_Perc' value
def calculate_right_out_perc(x, y):
    return y - 0.1537876 + 0.00005798503 * x - 0.000001621239 * x ** 2

# Apply the function to fill the Right_Out_perc column
df['Right_Out_perc'] = df.apply(lambda row:
calculate_right_out_perc(row['Flow_Rate_ml_min'], row['Left_Out_perc']),
axis=1)

# Convert the new data into a DataFrame
new_df = pd.DataFrame(new_data)
# Save the modified dataset to a new CSV file
new_df.to_csv("New_File.csv", index=False)
# Inform the user of completion of task
print("Modified dataset saved as 'New_File.csv'.")


''' Pseudocode

Create dataset from csv;
For each row X:
      starting_val = Row X Collumn C;
    # relative change calculation
      increment = (Row X Collumn C) - (Row X Collumn B);
    # before the resulting concentration reaches 1
      While ((starting_val + increment) < 1)
            Append the dataset with a row as follows:
                  Collumn A: Row X Collumn A;
                  Collumn B: starting_val;
                  Collumn C: starting_val + increment;
            starting_val += increment;

Apply best fit line for right exit to collumn
Save dataset to new csv titled "Fake_Data_3";

'''
```

Appendix C: GitHub Repository Link

GitHub