NTNU – Trondheim
Norwegian University of
Science and Technology

TKP4580: Specialization Project Report
Application of Machine Learning in heat exchanger network control

Khanh Hoang
Supervisor: Prof. Sigurd Skogestad
Co-supervisor: Allyne dos Santos

December 18, 2020

Contents

List of Figures

List of Tables

Abstract

Machine Learning is a famous research topic in computer science area. With the development of the computational hardware, there are many Machine Learning algorithms that could perform many complex tasks. Self-driving car is a great example of highly complex controlled tasks. There are many research on the controlled capability of Machine Learning algorithms and chemical engineering is not an exception.
In this project, we aim to investigate the performance of Machine Learning algorithms in controlled tasks. Our case study is a heat exchange system and we will use Machine Learning techniques to control the splits. There are three types of algorithms that used. The first algorithm is Multivarible linear regression that widely used in both data science and chemical engineering. The other two algorithms is Random Forest and Neural Network. They represent for two well-known ML techniques which are decision trees method and Artificial neural network.
Results show that Machine Learning algorithm could be used as a controlled algorithm. Two advanced algorithms - Random Forest and Neural Network have the advantage in nonlinear data set.

1 Introduction

Process control is a crucial part of a production plant. The ultimate goal of process control is to operate the plant in economical, stable, and optimal conditions. To make it possible, the plant requires a complex hierarchy of control, so that it could perform several control objectives such as safety or product quality. The hierarchy of control has different layers that manage the plant in many different time scales. In unit operating levels, the controlled layers normally consist of three layers: local optimization, supervisory control, and regulatory control. Regulatory control mainly uses PID controllers, however, local optimization and supervisory control require to solve optimization problems. Because of numerically solving optimization problems, it causes some difficulties when these two layers are implemented to the plant. Firstly, solving optimization problems normally consumes many computational resources because all the solving algorithms are expensive. The optimization layer will lose its advantages if the computational resources are limited. Besides the computational cost, the cost of building models is also a problem. For example, in the supervisory layer, we sometimes use MPC to perform controlled tasks and it requires the process model. To use MPC, the process model must be accurate enough to have good predictions. Last but not least, we usually face the problem that the license of commercial software is limited, thus, we could not perform the optimization online.

These above problems are great motivations to find an alternative method that handles these difficulties and Machine Learning(ML) could be a promising method. In a review paper, Venkatasubramanian(8) expects that ML will change all aspects of chemical engineering although it still faces many difficulties. Indeed, Shang's work (7) proofs the potential of data-driven technique through a crude distillation unit case study. Although ML is originally a research branch of the data science area, it could be beneficial for a reason. Many great ML frameworks are available such as Scikit-learn(5). It helps to start research in ML areas easier.

In this project, we aim to use ML algorithms to perform an optimally controlled task. The case study that we choose is a heat exchanger system which will be mentioned below section. We choose this heat exchanger system because it is simple enough to test the predicted ability of ML algorithms. Also, the generation of optimal data does not take much effort. We expect these ML algorithms could find the optimal operating condition without periodically solving optimization problems. The content of this project could be divided into four main parts. Section 2 gives a brief description of the heat exchanger network and the explanation of used ML algorithms. In section 3, the methodology of building and evaluating ML models will be presented. In section 4 and 5, the results and discussion on the controlled tasks will be mentioned. In section 6, it will conclude the advantages/disadvantages of used ML algorithms, as well as further work.

2 Background

2.1 Process description and measurement sets

2.1.1 Process description

Process information:

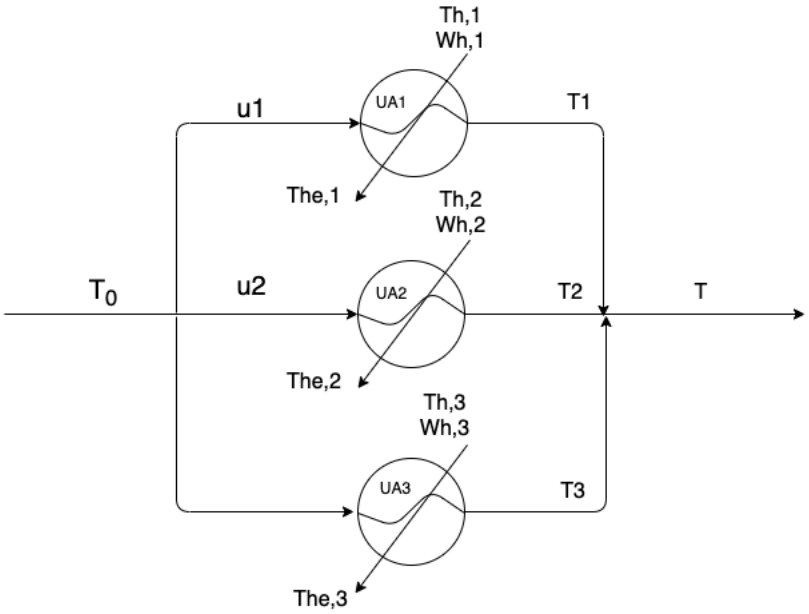

Figure 1: The heat exchanger system.

The process that we used in this project is a heat exchanger network with 3 branches. Our goal is that finding the splits u_1 and u_2 in order to optimize the outcome temperature T. We use the split u_1 and u_2 as the manipulated variables and T as the controlled variable. The disturbances are the following parameter : $[T_0, w_0, w_{h1}, w_{h2}, w_{h3}, T_{h1}, T_{h2}, T_{h3}, UA_1, UA_2, UA_3]$. Figure 1 depicts the heat exchanger system and notations.

Generate the training data set:

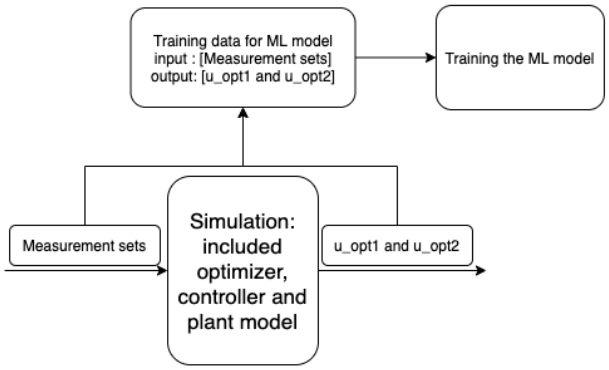

Figure 2: Scheme of data generation.

Machine Learning models require labeled data to tune model parameters. After being tuned, the model will have the ability to predict. Therefore, we couple the information of the process with the optimal splits u_1, u_2, then take them as the labeled data for training a Machine

Learning model. We have some measurement strategies to extract the information of heat exchanger system such as T_{h1} or w_0. Then, the optimal value of splits u_1, u_2 will be found by solving the steady-state optimization problem in Casadi - an optimization framework in Matlab (1). The mathematical model was given in Matlab.

2.1.2 Measurement sets

There are plenty of strategies to perform the measurement and we need to evaluate them in term of the efficiency in training the model. Therefore, we decide to investigate 4 measurement strategies for this heat exchanger network. Each measurement set combined with the optimal value of splits u_1, u_2 makes a data set for training Machine Learning models. The detail measurement sets are shown below and all notation can be found in Figure 1.
+ Measurement set 1 : $[T_0, T_1, T_{h1}, T_2, T_{h2}, T_3, T_{h3}]$
+ Measurement set 2 : $[T_0, T_{h1}, T_{h2}, T_{h3}, T_{h1e}, T_{h2e}, T_{h3e}]$
+ Measurement set 3 : $[T_0, T, T_{h1}, T_{h2}, T_{h3}, w0, w_{h1}, w_{h2}, w_{h3}]$
+ Measurement set 4 : $[T_0, T, T_{h1}, T_{h2}, T_{h3}, T_{h1e}, T_{h2e}, T_{h3e}, \alpha, \beta]$

2.2 Machine Learning algorithms

Machine learning is the research area that focuses on the algorithm that makes "machine" have the ability of "learning". To make it possible, Machine learning algorithms require a large enough amount of data which normally called *training data*. Then, we use this data to create a mathematical model and this model has the ability of new data prediction. That is called "learning" ability. Machine learning algorithms could be classified into 3 types: *Supervised Learning*, *Unsupervised Learning* and *Reinforcement Learning* In our problem, we use the supervised learning algorithm class.
In this project, we aim to use two different algorithms to perform the regression task. There are *Random Forest regression* and *Neural Network*.

2.2.1 Random Forest regression

"Random forests (Breiman, 2001) is a substantial modification of bagging that builds a large collection of de-correlated trees, and then averages them."(4)
- The Elements of Statistical Learning: Data Mining, Inference, and Prediction -

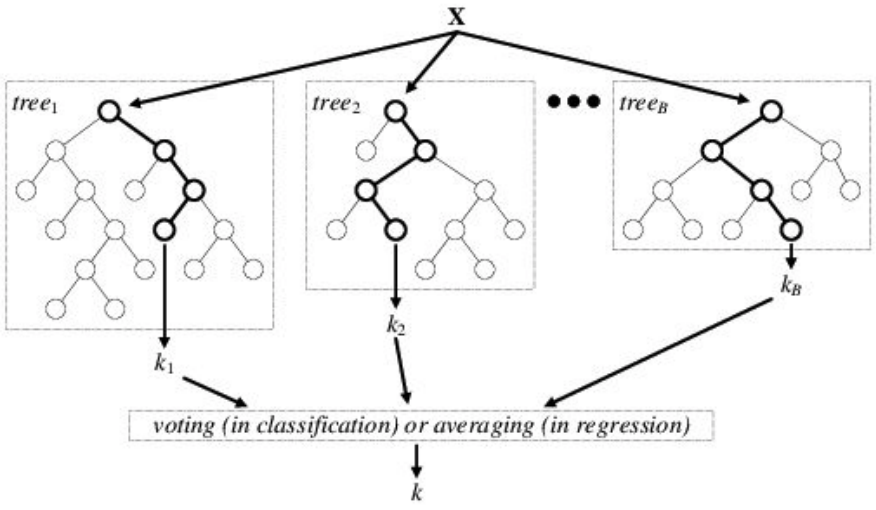

Figure 3: Architecture of the Random Forest algorithm.

Figure 3 (9) illustrates the Random Forest algorithm. *Random Forest algorithm* combines *Ensemble Learning method* and *Decision Tree Method* together, then create a generalized model and reduce the variance. Random Forest algorithm is widely used in the regression problem as well as the classification problem. In the pharmaceutical industry, Random Forest regression has been used in the prediction of dose-response (6). Before going into detail about the Random Forest algorithm, the concept of the Ensemble Learning method and Decision Tree Method should be explained.

- *Ensemble Learning method* is the method that combines several Machine Learning models. The purpose of ensemble learning is that reducing the variance, bias by averaging all the combined models.
- *Decision trees method* is the method that separates the domain of the predicted variable into several regions. In each sub-region, the value of the predicted variable is approximately the same with an averaging value.

The algorithm could be summarized through the following steps:
+ From N samples of the training set, we withdraw Z samples and divide into several subgroup.
+ Using the decision trees techniques to the subgroup, we can find the best splitting variable j and splitting point s.
+ After applying into the subgroup, we can achieve what called *the ensemble of trees* and use it for predicting the new point.

2.2.2 Neural Network regression

Neural Network is a widely used term nowadays because of the ability to solve many complex problems such as the classification or recognition of objects. With the increasing amount of data, Neural Network becomes an effective tool for data scientists and engineers. Neural Network algorithm is inspired by researches of the animal brain and has been interpreted in the mathematical language. There are many advantages of Neural Network algorithm that make it well-known. One of them is the generality of Neural network in many different areas. Neural network could have the ability of complex pattern recognition or data prediction using the same configuration. Therefore, neural network algorithm could be applied in the process control area due to the enormous amount of data in every process.

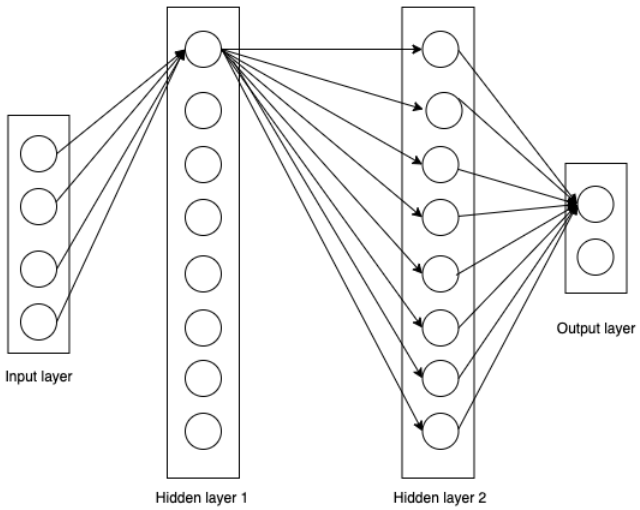

Figure 4: A typical structure of a Neural Network.

Neural network components

Figure 4 illustrates a typical neural network. A neural network has an input layer, an output layer and several hidden layers in between.
Input layer is the layer that informs the input information into the neural network. The number of nodes should be equal to the number of input variables.
Output layer is the layer that returns the values of predicted variables. The number of nodes in this layer should be equal to the number of output variables.
Hidden layers are layers in between *Input layer* and *Output layer*. Hidden layers create the complexity of the neural network. So, it could predict the output variables which have a nonlinear correlation with the input variables. We can optionally define the numbers of hidden layers and numbers of nodes in each layer a.k.a tuning the neural network.
Now, we zoom into each node and Figure 5 depicts the mathematical algorithm in each node. For short, we can understand it through the below equation :

$$y = f(w_1.x_1 + w_2.x_2 + w_3.x_3)$$

with f(x) is the activation function.

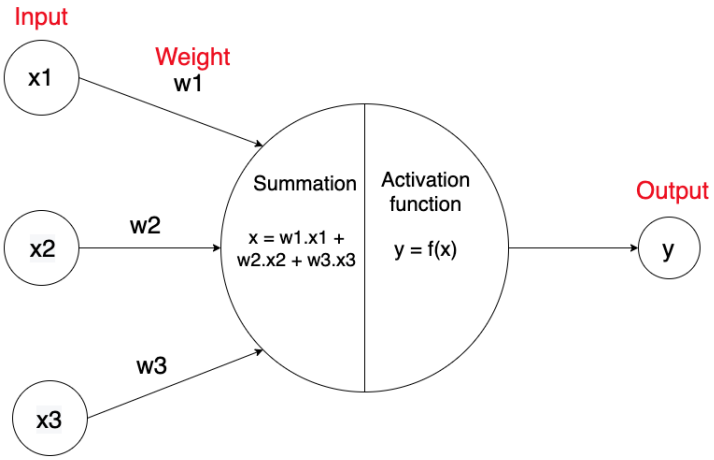

Figure 5: Mathematical description in each node.

In order to understand further concepts in Neural Network, we should be aware of two following definitions: *Weights* and *Activation functions*.
Weights are parameters that will be multiplied to input nodes. These weights will be found by solving a optimization problem in total neural network. Then, they will be adjusted by *backpropagation* step until the neural network could have a good predicting ability.
Activation functions are a mathematical rule that turn the summation of inputs to the output. We have many types of activation functions such as Relu or sigmoid.

Neural Network and the regression problem

Regression problem is finding the set of parameters that closely estimated the correlation between independence and dependence variables. For example, in linear regression - the simplest regression, we assume that the relationship between the independence variable x and the dependence variable y is a linear function $y = f(x) = a.x + b$ with a, b is parameters. In order to find a and b, we solve an optimization problem that bring the mean square error $\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$ close to 0.
Neural network regression is the same with the linear regression's idea. The difference is that the approximated function of Neural network is far more complicated, therefore, the number of parameters(weights) is greater. Then, solving the optimization also requires much more computational efforts.

3 Methodology - Machine learning controller frameworks

This framework inspires from modelling steps in data science areas. Figure 6 illustrates a typical workflow of Machine Learning project.

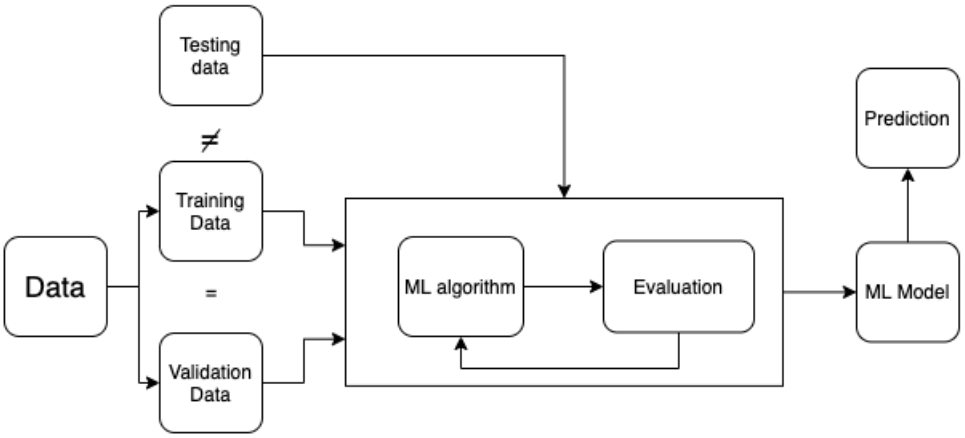

Figure 6: A typical workflow of a Machine Learning project.

In general, building a machine learning model could be separated into three main steps. The first step is gathering and processing data. In this step, we collect needed data, then analyze the data if it has missing data or how large the noise is. Then, we did some visualizations or normalization. That could suggest the first idea of which Machine Learning models that we should use. The next step is fitting the model and tuning hyperparameters. In this step, we mainly base on the computer to finish this step. The difficulty is tuning hyperparameters of ML model and mostly based on the experience. The final step is model evaluation and in fact, we have several techniques to assess the model.

In this project, we mainly base on the above workflow. Due to the purpose that building a controller, we have several engineering techniques in order to analyse the data as well as evaluate the model. Figure 7 is the workflow that we use in this project and the below sections will mention further about each step. The workflow could be divided into three main steps: *Pre-modelling step*, *Modelling step* and *Model Evaluation*.

Pre-modelling step
In this step, data analysis will be mainly focused. Data information such as distribution, the level of noise, or the outlier is crucial for the choice of ML algorithms. So, it is beneficial that we could have an initial analysis of the data that we have. In addition, in this project, we expect to build an optimal controller, then, the surface of the objective function should be researched. We could know how accurate the prediction.

Modelling step
This step is the same in data science. We mainly tune the hyperparameters of models and the tuning techniques come from data science areas.

Model Evaluation
There are several techniques that can evaluate a model in data science as well as process control. In this project, we take advantage of these two areas to evaluate our models. The first evaluation is opened loop analysis. Opened loop analysis is quite similar to the evaluation step in data science and we use some metrics such as MSE or MAE to assess the accuracy of

the prediction. We expect the model could give a good result of u_{opt1} and u_{opt2} that make the outcome temperature optimal. Therefore, we have other evaluations - closed loop evaluation. In other words, we plug our ML controller in the simulation loop to test it.

Programming language that we used for this project is *Python*. There are several reasons that Python has been used but the main reason is that Python has many Machine Learning frameworks. They could be easily implemented and stable.

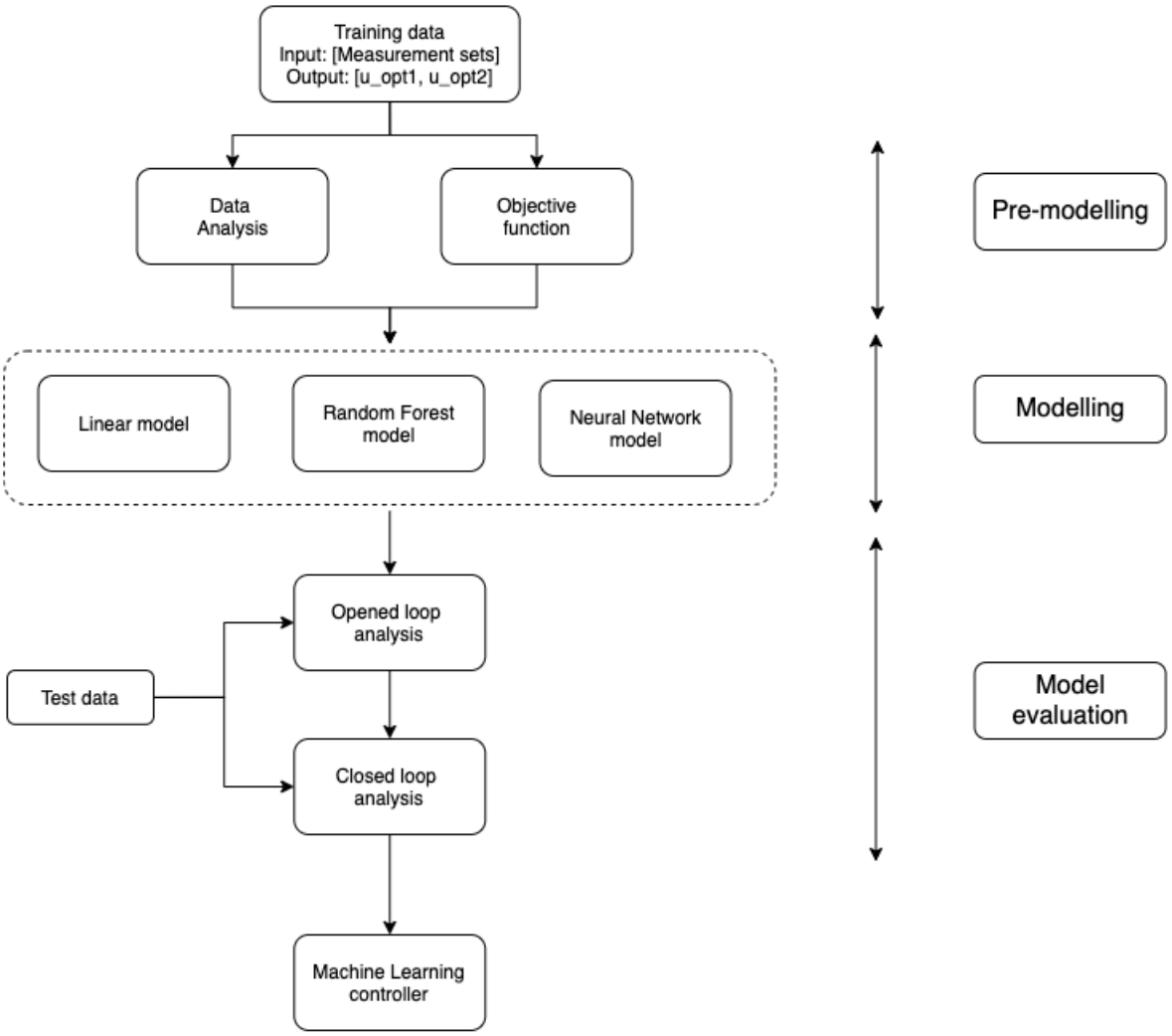

Figure 7: A ML control design framework.

3.1 Pre-modelling steps: Data analysis and Objective functions

3.1.1 Data Analysis

We use two types of plot: Box plot and Scatter plot to analyse the data. These plots are widely used in system identifications.

Box plot
Box plot is a good way to observe the distribution of data through some characteristic numbers such as minimum, maximum and median. The information that can be presented by the box plot includes:

- The data distribution, so that we can know if our data is normal distribution or unsymmetrical.
- Outliers - if we have many outliers or not and the values of them.
- Variance - the information of data noise, then we could use some kind of filters before modelling.

Box plot could hint for us which type of algorithms that we should use or if we need to pre-process data.

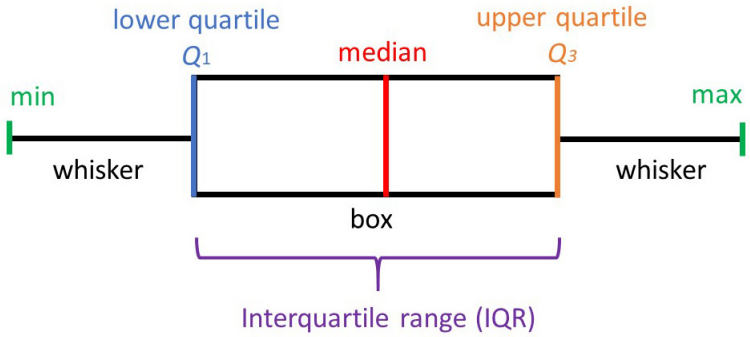

Figure 8: The interpretation of a Box plot.

Scatter plot

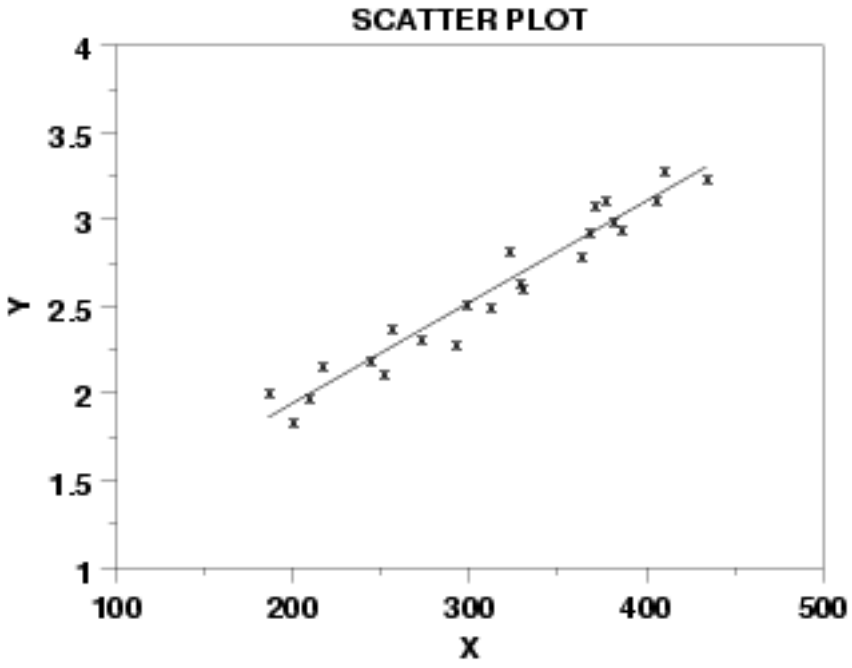

Figure 9: An example of the linear relationship.

Scatter plot is the most basic plot which widely used in the engineering and data science area. It represents the relationship between an input variable and an output variable through a dot. Mostly, the y-axis is the value of the output variable and the x-axis is the value of the input variable. It could be in 2-D or 3-D. Scatter plot is the easiest way that we can indicate

the linearity of data. Figure 9 is an example of the linear relationship.

3.1.2 Objective function

Before coming to modelling step, we should know the accurate level of predicted results. Due to the availability of plant model, it is possible that we investigate the changing of the cost depends on u_{opt1} and u_{opt2}. To achieve the surface of objective function, we vary values of u_{opt1} and u_{opt2}. Then, based on plant model, we could compute the surface of the cost which is a function of u_{opt1} and u_{opt2}.

The smoothness of the objective function's surface
We should concern about the smoothness of the surface because Machine Learning algorithms do not contain information related to the process. Therefore, algorithms could logically find out the optimal point. As a result, algorithms work well if we have a smooth surface and on the contrary, algorithms has the bad prediction if the surface has many local minimum points. Furthermore, awareness of the smoothness will significantly reduce the computational effort on building the model.

The accuracy of predicted splits: u_{opt1} and u_{opt2}
To investigate the expected accuracy, we plot the relationship of the cost gradient and predicted splits. The cost gradient is expected to converge to zero where indicate an optimal point. By plotting the variance of the cost gradient depends on u_{opt1} and u_{opt2}, we will know how accurate the split should be. The slop or the gradient has the smaller variance, the required accuracy is less. It is crucial and beneficial to know this information before building the model.

3.2 Choices of Machine learning models

After data analysis, we come to the modelling step. In this section, we decide to use 3 models, so that we could compare them to each other. We expect the model from Machine Learning techniques could be more effective than the linear model that is often used in chemical processes. The loss function that we choose for all 3 models is *mean square errors(MSE)*. In linear regression, we are quite straightforward to the concept of finding the parameter set. However, in the purpose of understanding the random forest and neural network algorithm, we are able to distinguish between two fundamental definitions : *hyperapameters* in *tuning model* and *model parameters* in *training model*. To illustrate the difference between them, we distinguish them in two separated steps in figure 10 and it is clear that they have different functions in two different steps.

Hyperparameters could be considered as the characteristics of the algorithm. They mostly are the configuration of a ML model such as the number of nodes in neural network, or the feature of an optimization algorithm such as learning rate. We will specify hyperparameters of each algorithm in the below sections. There are some computational methods of tuning these hyperparameters, however, it is better if we could tune them manually.

Model parameters are the mathematical parameters in a ML model. We can relate to the parameter set in the linear model, however, we have a lot more parameters in the random forest or neural network algorithm. Weights in neural network is a good example for the model parameters. These parameters could be found by solving optimization and it will be solved automatically by using the computer.

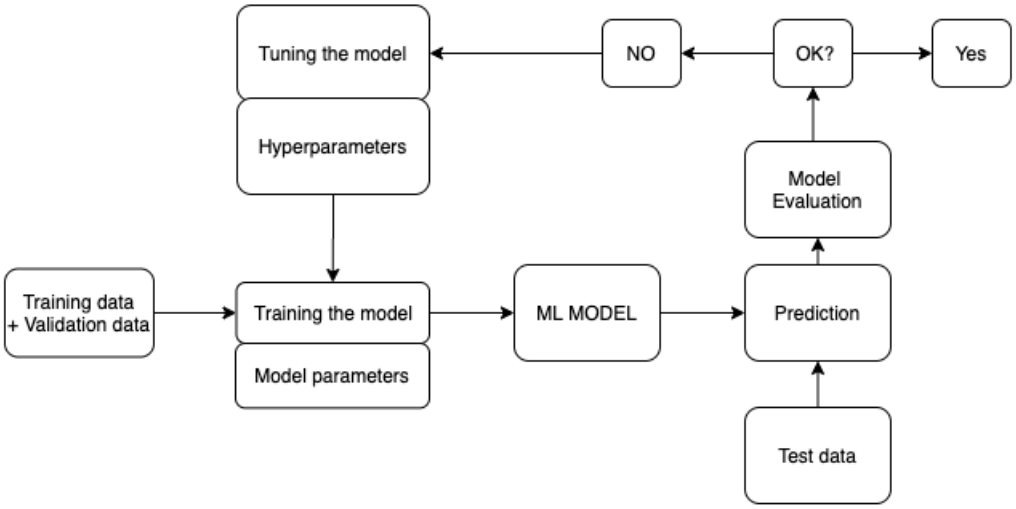

Figure 10: The tuning and training model step in the ML workflow.

3.2.1 Linear model

Linear regression is most basic algorithm in Machine learning and it also has many practical applications. The "linear" term is not the linear correlation between variables, it refers to the linearity in parameters. The mathematical expression of linear model is simple and could be illustrated in the below formula.

$$\hat{y} = \alpha_i.x_i + b_i$$

with i is the index of variables and $\hat{y}$ is the predicted value. x_i could be the variables or the non linear expressions of variables such as $sin(x)$ or x^2. The problems is that finding the set of parameters , therefore, we use the criteria is minimizing the error between the predicted value and true value: $\varepsilon = (y - \hat{y})^2$.

3.2.2 Random Forest model

The general concept of Random Forest algorithm has been depicted in the section 2. In this section, we will mention further about the hyperparameters of this algorithm and how we can tune them. In addition, the hyperparameter set is based on the framework from scikit-learn.

Hyperparameters
Scikit-learn framework provide us many hyperparameters to tune the algorithm. However, we just use three common hyperparameter: The number of trees in forest, Number of random variables from p-variable, and Nodesize. The others will set by default values of framework.
+ *The number of trees in forest* is the number of estimators, also is the number of subset which randomly withdrawn from the original set. In sklearn framework, we adjust 'n_estimators' to change the number of trees. The default value of n_estimators is 100.
+ *The number of random variables from p-variables* is the number of variables that algorithm will randomly pick from p variables. This hyperparameter can be changed by adjusting the value of max_features.
+ *Node size* is the number of observations in one node a.k.a leaf of each trees. In sklearn framework, we tune the value of min_samples_leaf.

Tuning method
In this project, we use the tuning values recommended from papers and books because they are good enough to achieve our goals. More detailed, In chapter 15 of The Elements of Statistical Learning(3), they recommend the value of m - the number of random variables should

be $p/3$ and the minimum mode size is five for regression. In this paper(8), they also suggest the same values of hyperparameters. Table 1 summarizes the value of these hyperparameters that we used.

Sklearn notations	Hyperparameter	Tuning values
max_features	Number of drawn candidate variables in each split	$p/3$
min_samples_leaf	Minimum number of observations in one node	5
n_estimators	Number of trees in the forest	20

Table 1: Summary of tuning values for Random Forest algorithm.

However, if we need to tune the model, there are two tuning approaches. The first approach is grid search and random search. It is simply that adjust the set of hyperparameter to minimum a evaluation metric or criteria. The second approach is the same idea, but we base on optimization problem to find an optimal set of hyperparameter. There are many auto-tuning framework that are available.

3.2.3 Neural Network model

Hyperparameters in Neural Network is easier to illustrate than in Random Forest. In the same way, adjusting the hyperparameter set is to find an appropriated model structure, so that, the tuned model has the ability of prediction with highly generalized capability. Instead of using sklearn framework, we use the framework that built by Keras(2).

Hyperparameters

In comparison to Random Forest, Neural Network has more hyperparameters to select and tune. In brief, two main purposes of hyperparameters are to construct the model's complexity To structure the model, and to construct the training process - searching algorithm. To define the model' s structure, we adjust the number of hidden layers and nodes, activation functions and its definitions are explained in the above section. To build up the searching algorithm, we have a set of hyperparameters: Choice of optimizer and the loss function, Learning rate and epochs, Regularization.

+ Choice of optimizer and the loss function: Optimizer is the built-in searching algorithm and the algorithm will search differently with different optimizer. The loss function is the criteria that the algorithm searches for the minimum value, normally it is Mean Squared Error (MSE).

+ Learning rate and epochs: Learning rate is the walking-step between two searching point and we adjust this to converge the search algorithm to a minimum point. If we have greater learning rate, the algorithm will converge faster, however, it is easier to drift away from a minimum point. Epochs is the number of re-calculated time. In each epochs, the algorithm re-calculates the weights - backpropagration algorithm. It combines with learning rate to converge the algorithm to the optimal point.

+ Regularization: Neural Network always has overfitting problems and the purpose of regularization is to reduce the overfitting. In general, regularization will randomly remove amount of weights, so that, the model will be more generalized. For example, regularization will automatically eliminate some weights - it is called as dropout technique. In other regularization technique, we put a penalty term to the loss function to minimize the total sum of weights.

Tuning method

In the same way as Random Forest, there are auto-tuning tools that are available for Neural Network. However, we aim to understand more about the algorithm. Thus, we tune the Neural Network manually and the below is the method that we used to tune the model.

Tuning steps:
+ Step 1: Select numbers of layers and nodes and select the loss function and metric.
For regression, the loss function normally is MSE, the metric is MAE and the optimizer is Adam. The selection of layers and nodes is based on this paper(10). They stated that a very simple two-layer ReLU network with p = 2n+d parameters that can express any labeling of any sample of size n in d dimensions. It fints us the number of layers and nodes that should use
+ Step 2: Select activation functions for each layer.
As the paper, they recommend that the activation function in two layers are ReLu. However, in our case study, we expect the output results should be in the range of 0 and 1. Sigmoid is the function guaranteed it could happen, so that, we choose sigmoid as the activation function for the second layer.
+ Step3: Apply some regularization.
As mentioned above, the purpose of using the regularization is to avoid the overfitting issue. In this project, we only use the L2 regularization. L2 regularization add the sum of squared weights into the loss function. Then, we adjust the coefficient of that term to manipulate the effect of the penalty term.
+ Step4: Adjust Learning rate and epochs, so that the algorithm could convert to a good enough local minimum

3.3 Models evaluations

Model evaluation is a common term in both data science and process control. However, we have different methods to evaluate a model. In data science, we mainly focus on comparing data. It means that we measure the predicted ability by using several types of metrics. Otherwise, model evaluation in process control focuses on the accuracy of a model in terms of interpreting the physical system. We use knowledge-based methods to assess the chosen model and it should be good enough to serve engineering tasks. In this project, we aim to use techniques in both areas to evaluate our models. Opened loop analysis refers to data-based techniques and Closed loop analysis mainly uses engineering metrics.

3.3.1 Opened loop analysis

In Machine Learning research, we have many evaluation metrics, however, in this project, we just use two common metrics: *Mean Absolute Error - MAE* and *Root Mean Square Error - RMSE.*
Mean Absolute Error - MAE :
Mean Absolute error is one of the most common metrics. MAE is the average of predicted error by taking the sum of errors, then divide to the number of observations. The calculation of MAE is shown below.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$$

Root Mean Squared Error - RMSE
Root Mean Squared Error is another common metric that widely use in statistical study. The difference with MSE is that RMSE is more sensitive to the outlier. RMSE could be calculated by the below formula.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

There is not much information if we compare MAE to RMSE. To evaluate the model, we analyse these metrics of the training set and testing set. It could tell us the accuracy of predictions. Besides MAE and RMSE, there are few more techniques that evaluated a ML

model. However, due to the limited research time, we could not include them in this project. Furthermore, our process is quite simple, so that, MAE and RMSE are good enough to analyse the model. We aim to use more evaluating techniques in further research.

3.3.2 Closed loop analysis

As mentioned before, the main purpose of closed loop analysis is to evaluate the ability to control the process of the ML model. Our goals are to find the optimal operation and to stabilize the process. In this project, we focus on analyzing the optimizing ability of ML model and partly discuss the stabilizing ability. We use three engineering metrics to evaluate the model: The cost, the gradient, and the loss.

The cost is normally notated as J. In our case study, the cost is the outcome temperature which is notated as T in figure 1. We aim to find an optimal split to maximize T. By using the process model, we could calculate the cost based on the values of splits.

The gradient which notated as J_u is the rate and direction of changing in the cost(J) depends the split(u_1 and u_2). J_u is used to interpret the accuracy of predictions and how close we are to the optimal point($J_u = 0$).

The loss is the difference in costs if we use the ML model in comparison to optimal costs and it is simply calculated by taking the difference between the resulted cost and the optimal cost.

$$Loss = J_{opt} - J$$

4 Results

4.1 Data information and analysis

Data is an important element to build data-based model. Therefore, it is essential that give attentions to analyse the data before modelling. Table 2 summarizes the basic information about our data.

	Information
Input	data set 1 - $[T_0, T_1, T_{h1}, T_2, T_{h2}, T_3, T_{h3}]$
	data set 2 - $[T_0, T_{h1}, T_{h2}, T_{h3}, T_{h1e}, T_{h2e}, T_{h3e}]$
Output	$[u_{opt1}, u_{opt2}]$
Parameter	$[T0, w0, wh1, wh2, wh3, Th1, Th2, Th3, UA1, UA2, UA3]$
Training set	1000 data points
Test set	1000 data points

Table 2: Basic information of data sets

1000 data points is not a large data set in comparison to a data set that is normally used in a ML project. Therefore, the generalization capability of the ML model should be concerned. However, we have several ways to handle this problem such as model regularization. We expect that the model still works well with this amount of data.

Box plot
Box plot is an informative way to have an overview about data. Figure 26 and 12 is the box plot of measurement set 1 and 2. From these plots, we have some common comments on both measurements:
+ Both data sets are normally distributed.
+ There are not many outlier in both datasets.
+ T_{h1}, T_{h2} have the largest variance in both datasets.

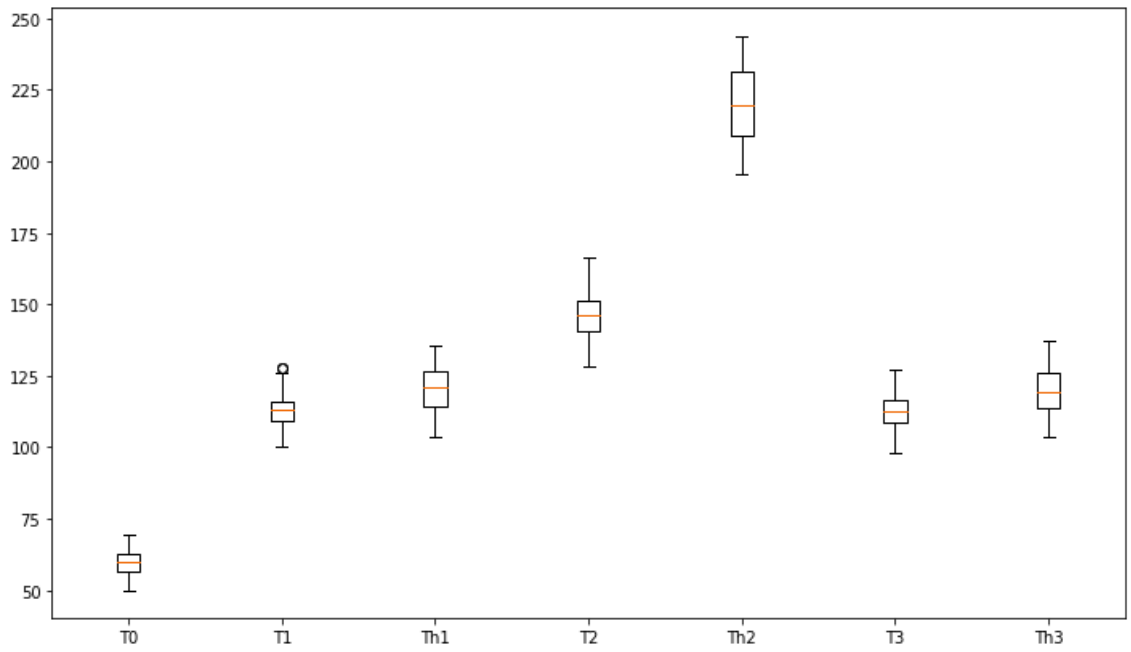

Figure 11: Box plot for measurement set 1.

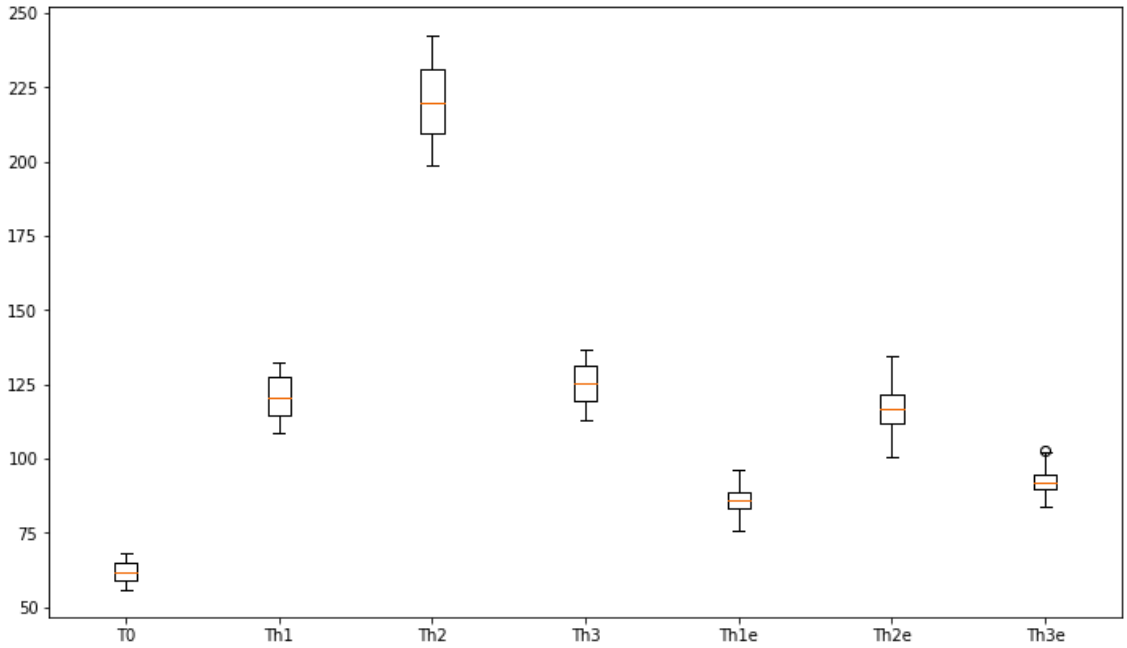

Figure 12: Box plot for measurement set 2.

In both measurements, we have a good quality of datasets because these datasets do not have any missing values or outliers. More importantly, they are symmetric distribution. In addition, they are all temperature measurements, so that, their scales are not largely different. Due to the good dataset, we do not need to preprocess the data such as data cleaning or data normalization.

Scatter plot

The purpose of the scatter plot is to identify the linearity relationship between dependent and independent variables. In our cases, u_{opt1} is linear with T_{h1} and u_{opt2} is linear with T_{h2}. This could be illustrated in Figure 13 and 14.

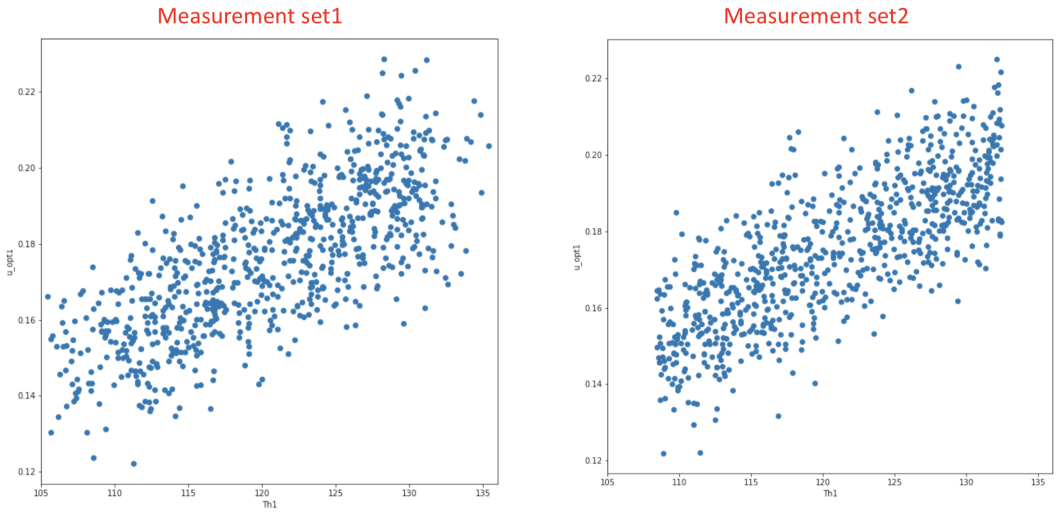

Figure 13: The linearity between u_{opt1} and T_{h1}.

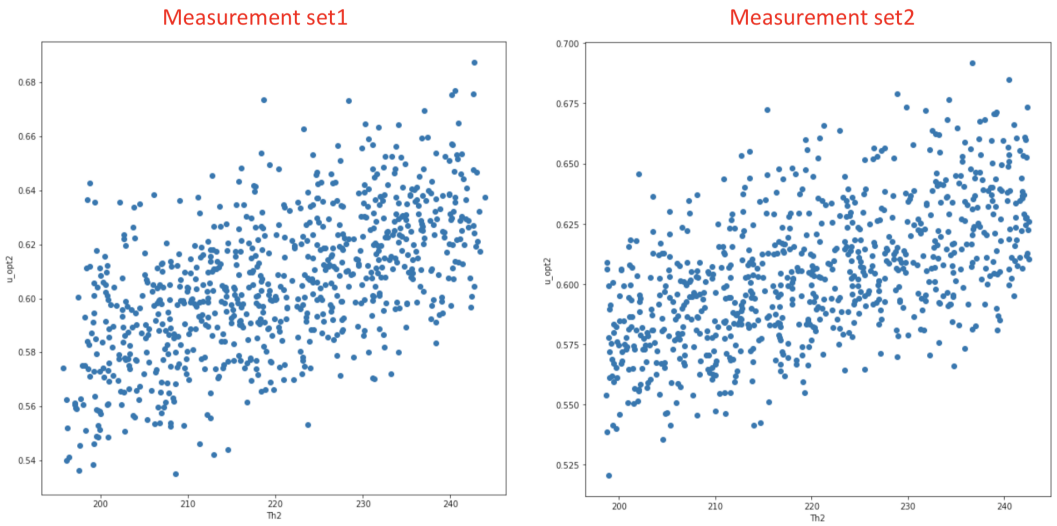

Figure 14: The linearity between u_{opt2} and T_{h2}.

However, the linearity in the measurement set 2 is not as strong as measurement set 2. Thus, we expect that neural network and random forest will be more effective than the linear model in the measurement set 2.

Surface of of objective function
Figure 15 shows how J changes with u_{opt1} and u_{opt2} and it is a quite smooth surface.

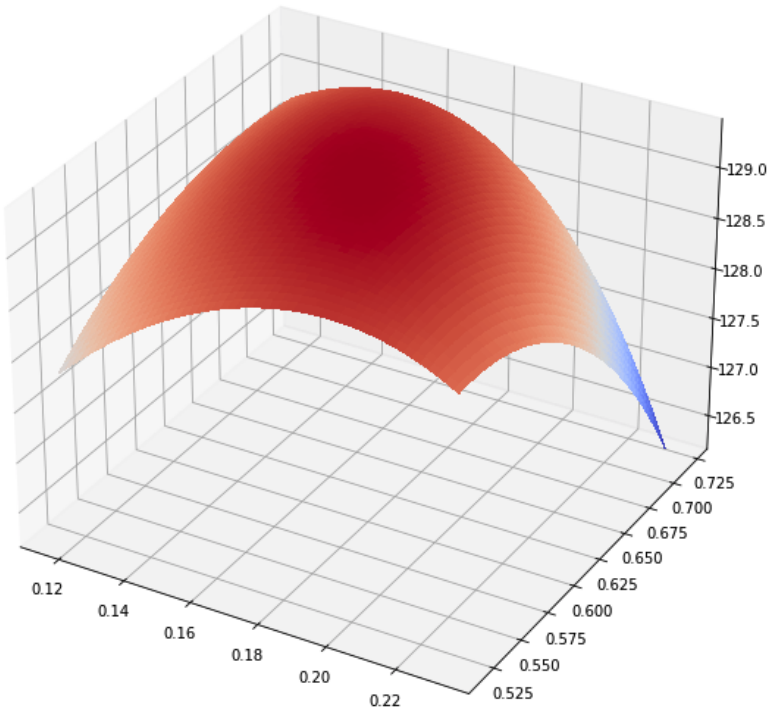

Figure 15: Surface of the objective function.

Figure 16 is the contour of the function of J depended on u_{opt1} and u_{opt2}. Since the surface of objective function is smooth, the cost is not much changed when the accuracy of u_{opt} is kept at 10^{-3}. It means that the algorithm could be allowed to have the predicted error of 10^{-3}.

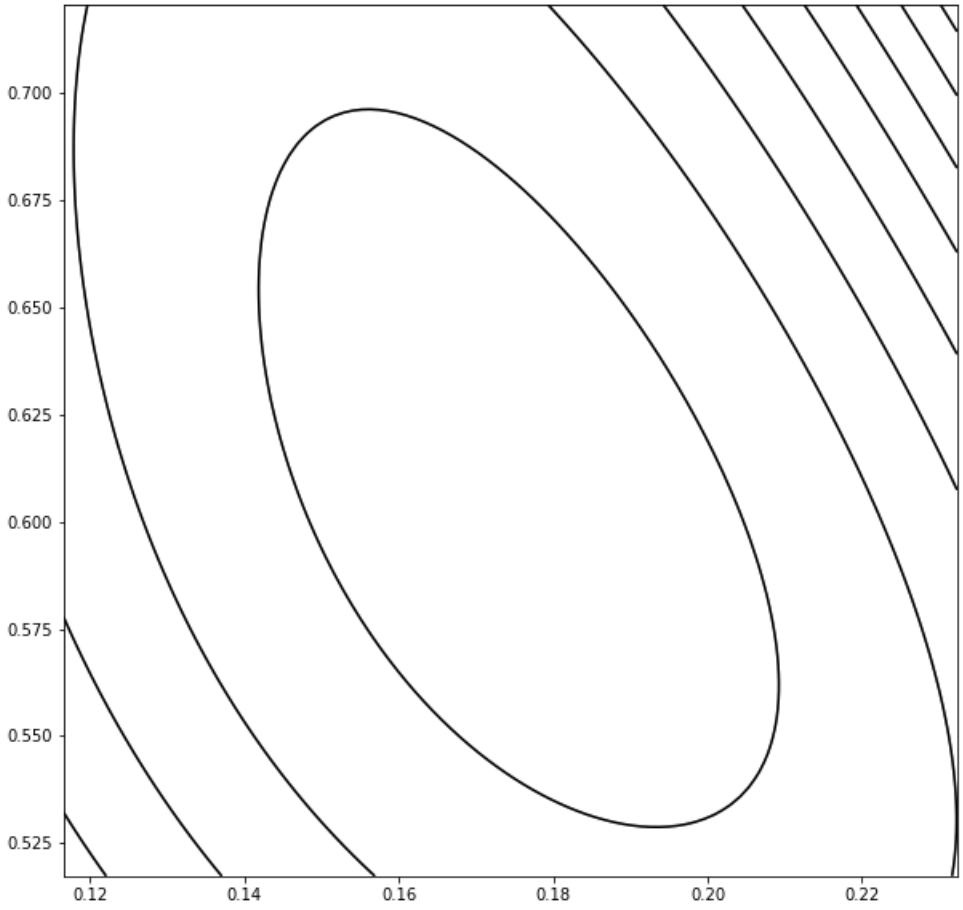

Figure 16: The contour lines of the objective function.

4.2 Opened loop analysis

Measurement set 1: $[T_0, T_1, T_{h1}, T_2, T_{h2}, T_3, T_{h3}]$

Table 3 shows values of MAE and RMSE of each model in training and testing set. RMSE is always greater than MAE and is more sensitive to outliers. By comparing RMSE and MAE, we could indicate which method is sensitive to outliers and it is Random Forest algorithm. Values of MAE and RMSE are quite similar, thus, Linear model and Neural Network are less sensitive. However, when we compare the testing results to the training results, multivariable linear regression has a better generalization capability than Random Forest and Neural Network. These methods have a good prediction in u_{opt1}, but have a bias with the u_{opt2} prediction. On the other hand, linear model seems to have the same results in both training and testing sets. In measurement set 1, linear model has a better performance in comparison to the other two. Therefore, it makes sense that we expect this also happens in closed loop analysis.

u_opt1	Multivariable Linear Regression	Random Forest Regression	Neural Network Regression
MAE	0.012	0.0118	0.0159
RMSE	0.016	0.0149	0.0197
MAE in training set	0.0034	0.00021	0.0107
RMSE in training set	0.0046	0.0147	0.0132

u_opt2	Multivariable Linear Regression	Random Forest Regression	Neural Network Regression
MAE	0.018	0.0195	0.016
RMSE	0.022	0.0245	0.02
MAE in training set	0.011	0.43	0.435
RMSE in training set	0.014	0.432	0.437

Table 3: Statistical Metrics in the measurement set 1.

Measurement set 2: $[T_0, T_{h1}, T_{h2}, T_{h3}, T_{h1e}, T_{h2e}, T_{h3e}]$

Results in measurement set 2 are shown in Table 4.

u_opt1	Multivariable Linear Regression	Random Forest Regression	Neural Network Regression
MAE	0.0162	0.0117	0.012
RMSE	0.02	0.0147	0.0167
MAE in training set	0.01	0.0029	0.0072
RMSE in training set	0.012	0.0041	0.0091

u_opt2	Multivariable Linear Regression	Random Forest Regression	Neural Network Regression
MAE	0.021	0.0197	0.0178
RMSE	0.025	0.0243	0.022
MAE in training set	0.013	0.429	0.426
RMSE in training set	0.016	0.431	0.428

Table 4: Statistical Metrics in the measurement set 2.

From the data analysis section, we mention that the data in this measurement is less linear. Therefore, we expect that the linear model does not work well as the other two. Indeed, the MAE and RMSE score of linear model is worse than Random Forest and Neural Network algorithm. When data is not linear, these two algorithms begin to take advantage.
However, the generalization capability of the linear model is still better because it has the same values of MAE and RMSE. As the measurement set 1, Random Forest and Neural Network still give the precise prediction of u_{opt1}, but have large errors in u_{opt2}.
Before coming to closed loop analysis, we believe that the generalized problem could be solved by re-tuning the model in these algorithms. The main point that Machine Learning algorithms are potential in optimal operation prediction was proof. Furthermore, they could be effective when we have a nonlinear dataset.

4.3 Closed loop analysis

4.3.1 Measurement set 1:$[T_0, T_1, T_{h1}, T_2, T_{h2}, T_3, T_{h3}]$

The prediction of u_{opt1} and u_{opt2}:

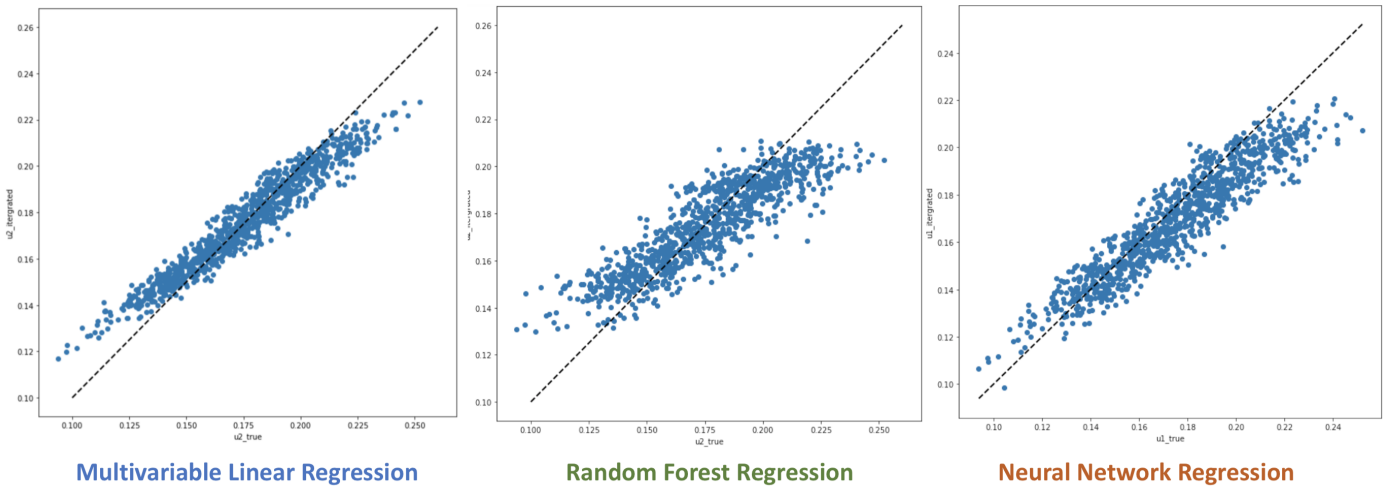

Figure 17: The prediction of u_{opt1} in measurement set 1.

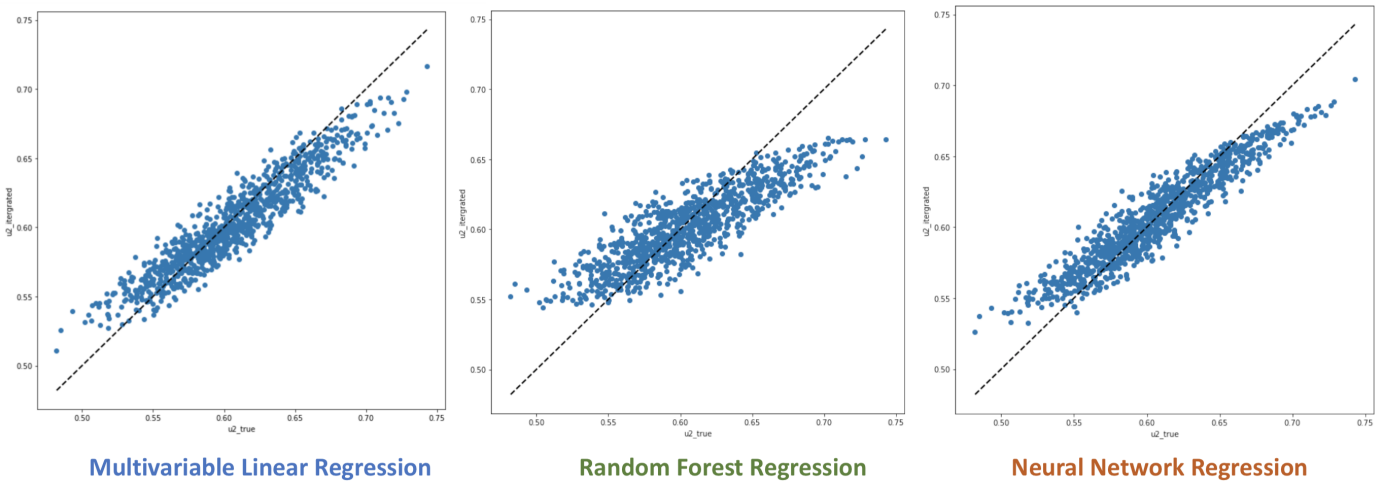

Figure 18: The prediction of u_{opt2} in measurement set 1.

Generally, we all have good results in all three chosen models. With previous analysis, we expect that Random Forest is more sensitive to outliers and these two response better to outliers. Figure 17 and 18 clearly agree with this conclusion. Random Forest does not have the wide range of the extrapolation and this makes sense if there is a outlier which is out of predicted range of Random Forest algorithm.

Results of J_{u1} and J_{u2}

The gradient J_{u1} and J_{u2} show how closed we are to the optimal point. Below figures show J_{u1} and J_{u2} results of three models. The gradient J_{u1} in the linear model is relatively better than other models. J_{u1} has the smallest variance, then Neural Network is the second variant. Random Forest is the widest range of variance in J_{u1}.
Linear model still performs well in the gradient J_{u2}. Random Forest still has the worst result in J_{u2}. Although Neural Neural has better performance, it has a small bias in J_{u2}. The performance of linear model is expected because it has the best result in opened loop analysis.

Therefore, it makes sense that linear model performs better in the feedback loop.

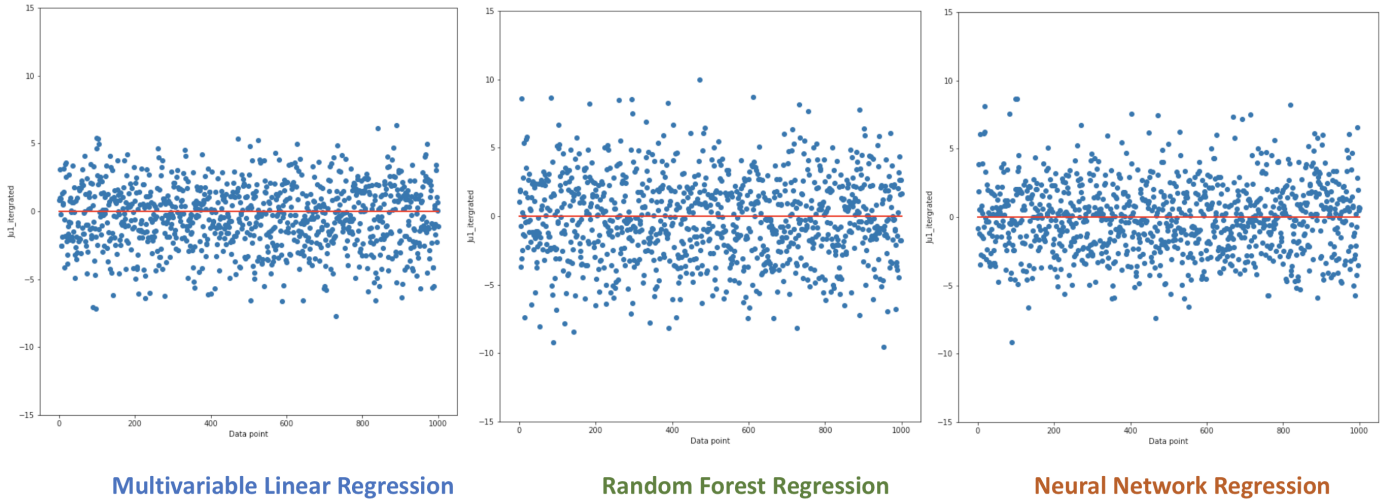

Figure 19: The gradient J_{u1} in measurement set 1.

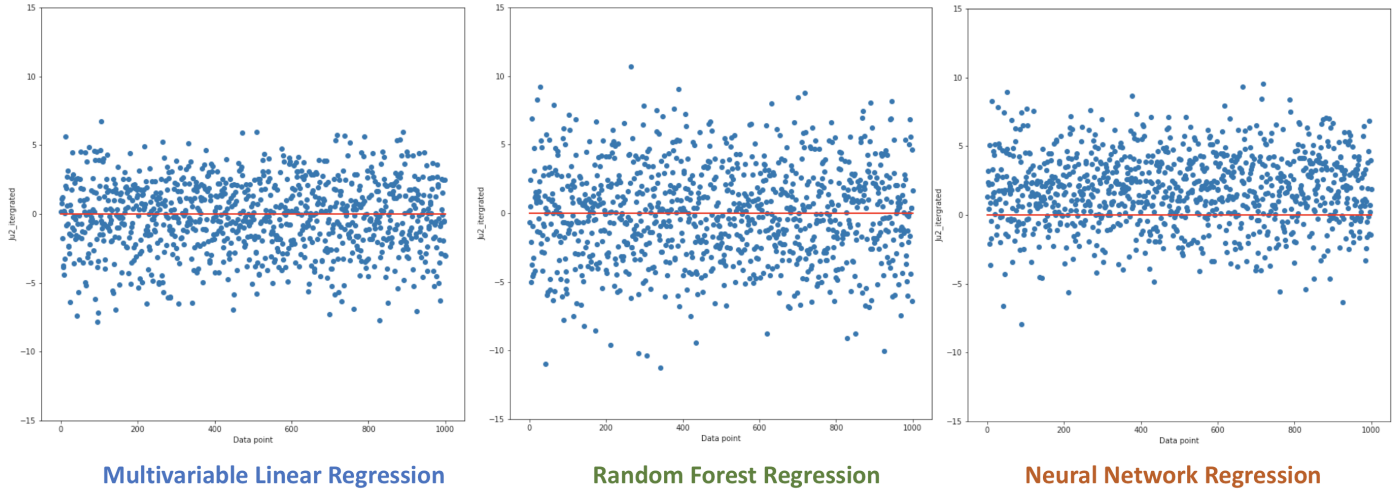

Figure 20: The gradient J_{u2} in measurement set 1.

Results of Cost

Metrics	Multivariable Linear Regression	Random Forest Regression	Neural Network Regression
MAE	0.0278	0.096	0.089
RMSE	0.0393	0.14	0.104

Because the difference in cost is small, it is better if we compare the predicted by using the MAE and RMSE metrics. In this measuremet set, the best result is the linear model.

Results of Loss

Figure 21 show results in temperature loss in each model. Due to the performance in J_{u1} and J_{u2}, we can easily predict that linear model give smallest loss in all three model. Indeed, the result in Figure 21 proof that. The loss is significantly lower in linear model because of the low variance in both J_{u1} and J_{u2}.

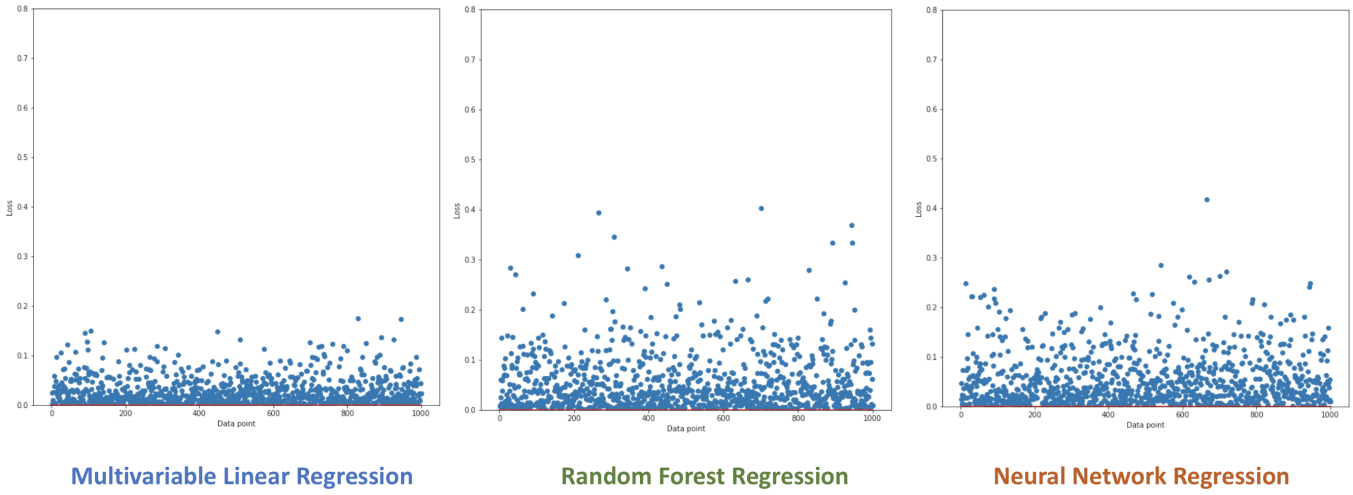

Figure 21: The Loss in measurement set 1.

4.3.2 Measurement set 2 : $[T_0, T_{h1}, T_{h2}, T_{h3}, T_{h1e}, T_{h2e}, T_{h3e}]$

The prediction of u_{opt1} and u_{opt2}

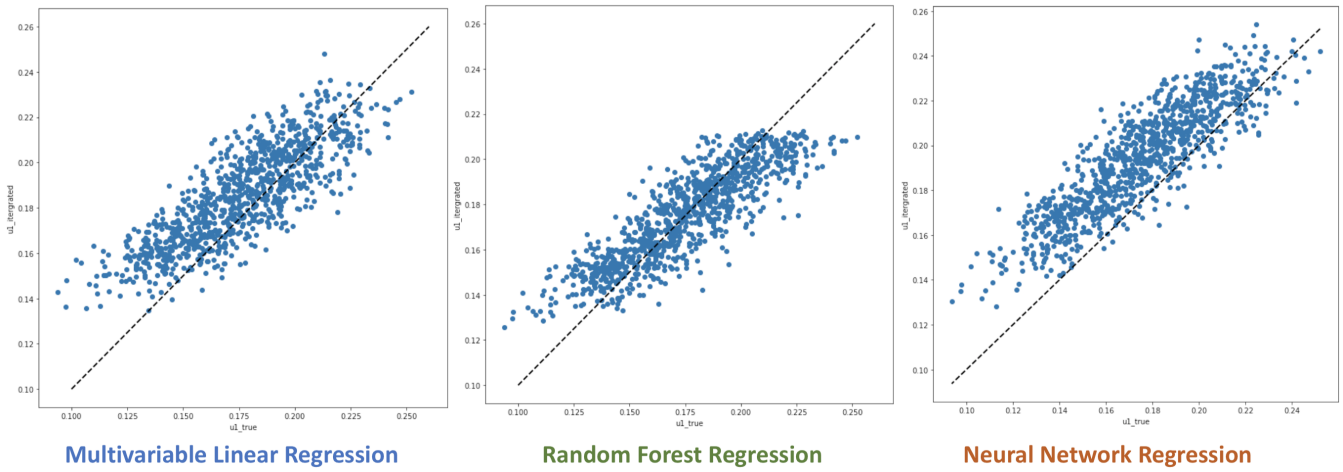

Figure 22: The prediction of u_{opt2} in measurement set 2.

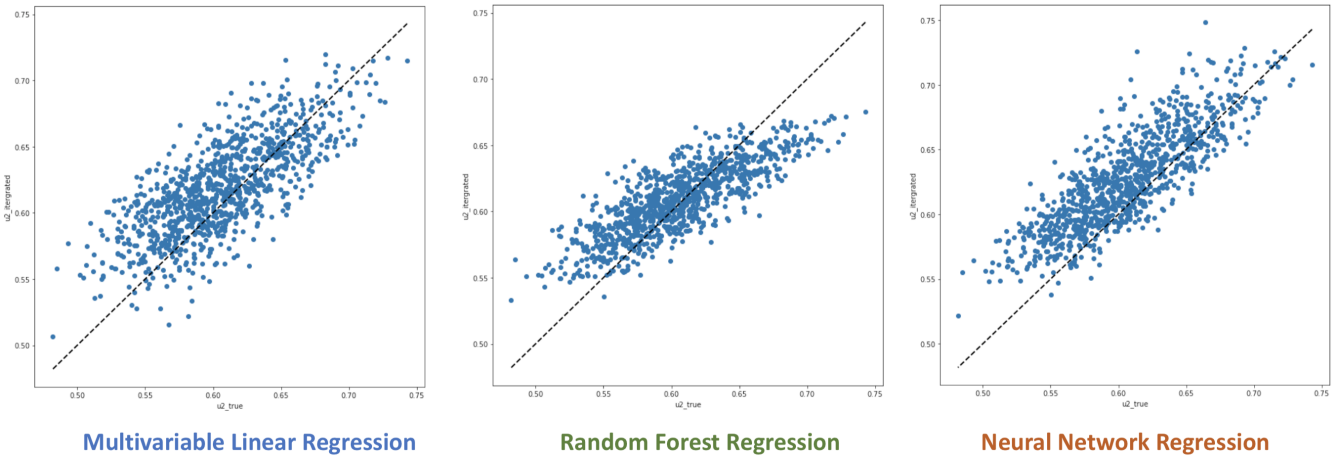

Figure 23: The prediction of u_{opt2} in measurement set 2.

In the measurement set 2, results in the u_{opt1} and u_{opt2} prediction are not good as the measurement set 1. This is expected because measurement set 2 is less linear than measurement set 1. In addition, linear model begins to lose its advantage in this case. The performance of linear model has wide range of variance in both u_{opt1} and u_{opt2}. The result in Random Forest is quite similar to measurement set 1 and Neural Network gives the bias in the prediction. Based on results on u_{opt1} and u_{opt2}, Random Forest and Neural Network are expected to perform better in finding the optimal cost.

Results of J_{u1} and J_{u2}

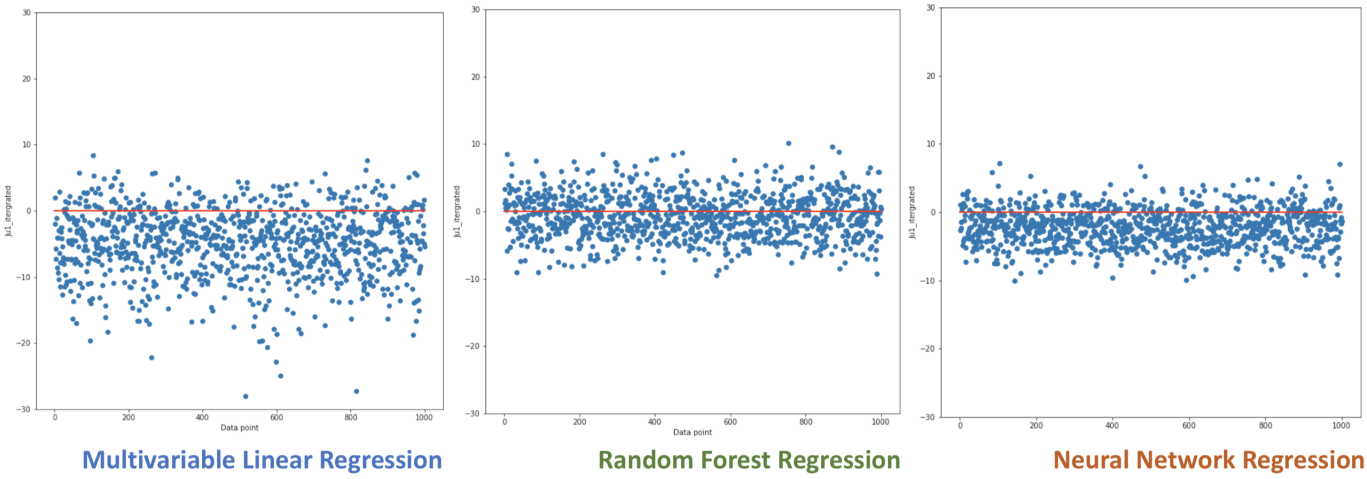

Figure 24: The gradient J_{u1} in measurement set 2.

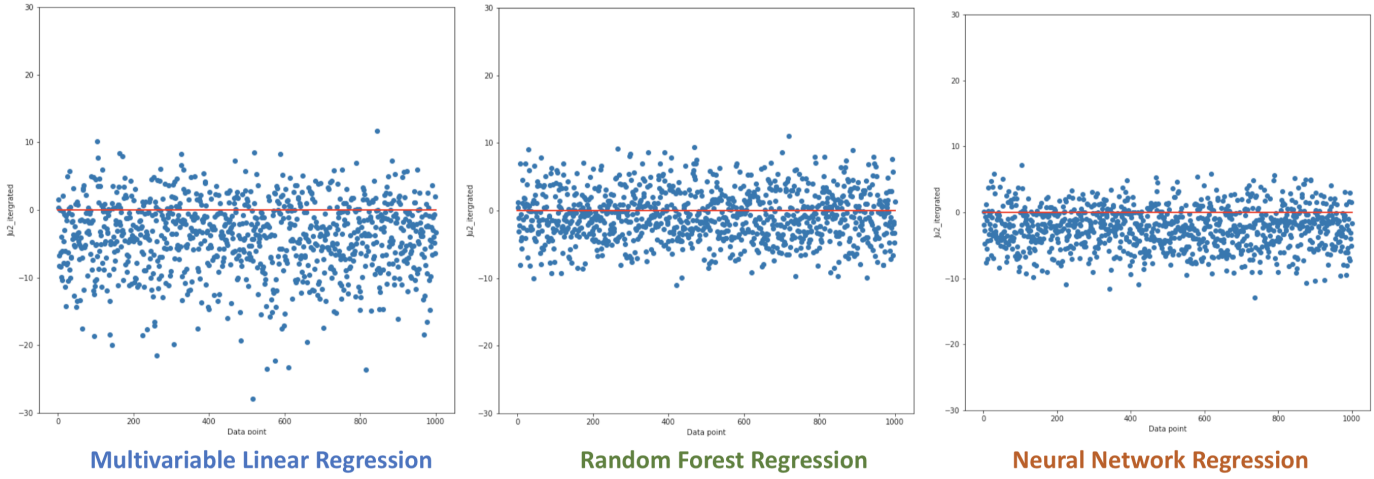

Figure 25: The gradient J_{u2} in measurement set 2.

Indeed, figure 24 and 25 show that the performance of linear model is worst. It gives the bias and high variant results in gradient J_{u1} and J_{u2}. On the other hand, Random Forest still has the same quality of prediction as to the previous measurement. Neural Network does not converge to the optimal point in most of the cases, but it has the smallest variant range. Based on these results, the linear model may give a higher loss in comparison to others.

Results of Cost

Metrics	Multivariable Linear Regression	Random Forest Regression	Neural Network Regression
MAE	0.139	0.064	0.091
RMSE	0.044	0.092	0.14

Random forest and Neural Network have the better result of the predicted cost in this measurement. This agree with these previour analysis. Even with the larger margin of error, these prediction algorithms still are acceptable.

Results of Loss

As a result, the loss in the linear model is highest in all three models. The variance is up to around 1.5 degrees Celsius. On the contrary, Random Forest and Neural Network keep the variance that a maximum of 0.4 degree Celsius. As we expected, Random Forest and Neural Network give better results when we implement them in closed feedback loop.

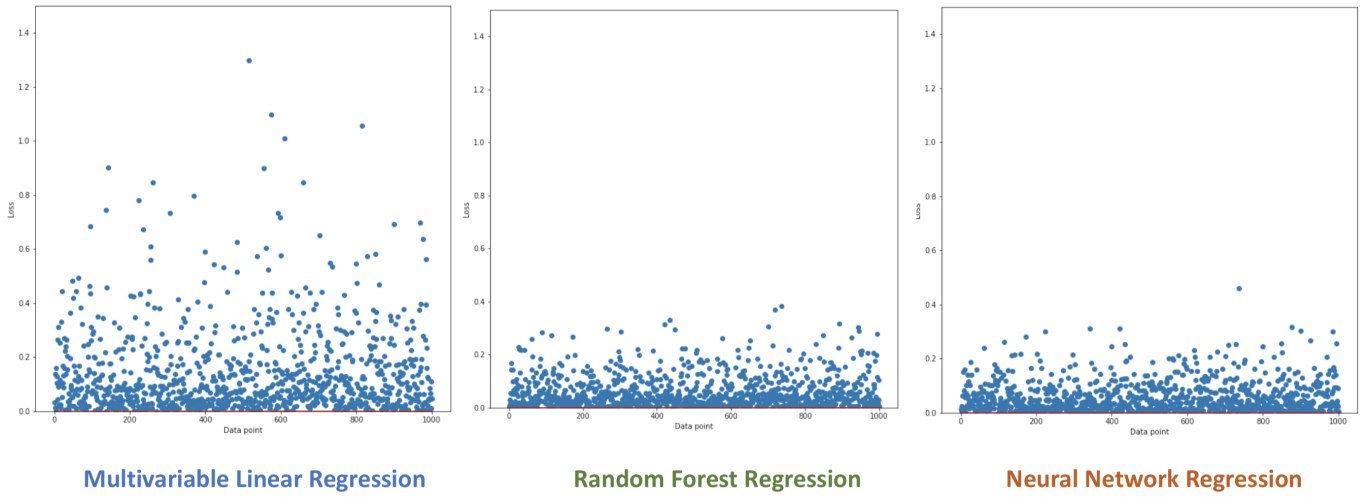

Figure 26: The Loss in measurement set 2.

5 Discussion

5.1 General discussion

Multivariable linear model
Not only in data science but also chemical engineering linear models are widely chosen because of the efficiency of it. However, it requires to do some data analysis before modelling. In this project, to find out linearity, we had to do many data visualizations and plots. This method is strongly dependent on the linearity of data. As previous results, the linear model performs worse in the second measurement which has less linear data. In contrast, the performance of the linear model is quite impressive if the data set is good enough. Therefore, due to the simplicity in model structure and parameter searching, it is strongly recommended that we used the linear model when we have linear data.

Random Forest Algorithm
Random Forest works well with both cases of measurements. Accordingly, Random Forest remains an acceptable result. Besides that, the model tuning takes fewer efforts and the complexity is low. Therefore, Random Forest requires less computational calculation than Neural Network. However, some disadvantages existed. The first one is that RF algorithm is sensitive to outliers and gives the wrong prediction. This problem is not good for a controlled algorithm because it should not give a non-sense controlled signal. Secondly, Random Forest algorithm has a small range of extrapolation. It means that the algorithm could not handle the large variance of disturbances. In brief, Random Forest is a potential Machine Learning algorithm that could be used as a controller algorithm.

Neural Network Algorithm
Neural Network is the algorithm that has the most complex model structure. Its number of parameters also is the greatest. Thus, it is expected to be the most generalized model. Consequently, it has the best result in all three chosen models. It has a small range of variance and trivial bias. Having the same results as Random Forest, it has a wide range of extrapolation. So, it has a better ability to handle the disturbance. If we evaluate these models in many engineering aspects, Neural Network is the best performer in all three. In addition, it requires less data analysis than the linear model. All things considered, many pieces of evidence prove that Neural Network could a promising controlled algorithm. Besides these advantages, this algorithm still faces many problems. Firstly, due to a large number of hyperparameter, it takes a lot of effort to tune the model. In particular, the tuning time of Neural Network is twice of Random Forest and it required the same effort as doing the data analysis of the linear model. Secondly, it has a high level of complexity, therefore, the calculation requires more computational resources.

Tuning Method
In this project, the tuning method is done manually and mostly based on personal experiences. The main purpose of the project is to understand different types of ML algorithms and to explore the controlled ability of chosen algorithms. Thus, the result is not optimal. In reality, many auto-tuning tools are available. This not only helps us to have a better machine learning model but also saves time on the model tuning step. With these tools, the result will be far better. Auto-tuning tools, in simpler terms, are these tools could search an optimal set of hyperparameter and they are based on the numerical optimal searching method.

5.2 Control ability

Convergence problem
The Machine Learning model is the modelling method based on data exploration. This type of model does not contain any information about the plant and cannot logically handle the disturbance. The controlled signal is given by the information of measurements which mostly contain the noise. For this reason, the predicted output signal could be oscillated and does not converge the controlled signal to a specific point. That is not good for performing a controlled task. However, this problem could be solved by partly implement the controlled signal.

Optimal Operation
Due to the oscillation of the controlled signal, the cost also is fluctuated. Thus, the total profit of the process is not optimal. Even if we partly implement the controlled signal, the algorithms do not converge the controlled signal to the optimal point. So, we do not achieve the optimal operating condition. In our case study, we have an uncomplicated surface of the objective function. We can easily converge to the point that is closed to the optimal condition. In real problems, we may have to face more complicated problems.

The choice of the measurement set
In the first principle model, sometimes we have a specific measurement set to fit the model. However, in the data-based method, we can freely choose the measurement set. It is beneficial that if the cost of measurement is expensive or the process is soft-sensing.

6 Conclusion

Advantages of Machine Learning algorithms
There are several advantages of using Machine Learning as controlled algorithms. First of all, ML model requires less computational resources than solving real time optimization. ML model takes effort to train the model and data generation, but it consumes less resources because after training, ML model is just a mathematical function. Secondly, results in this project show that ML model could perform a controlled task, even if the data is non-linear, the ML model still has a good performance. Controlling the system in an acceptable loss is the evident of a potential algorithm. Last but not least, many existed tools support the model tuning such as Sklearn and Tensorflow. Based on the optimization method, results from auto-tuning tools is guaranteed that optimal results. For those reasons, Machine Learning is a great topic to research in chemical engineering.

Problems with Machine Learning algorithms
Besides these advantages, ML algorithm still remain problems that is crucial if used them in the plant. The first problem is that sometimes the algorithm does not converge the controller signal. The controlled signal will be fluctuated because of the presentation of the measurement noise. We can solve this problem by partly implement signal to the plant. However, it may be not the optimal operation. The second problem is out-of-range problems. As the role of controlled, the algorithm should have the ability of extrapolation. However, if there is a strong outlier existed (the large disturbance), Machine Learning model may have a wrong prediction. The last problem is overfitting problem. We always have to care about this problem because if the model is overfitting, it will not work in the real plant.

Future works
As mentioned in section 2, we have the measurement set 3 and 4. The data in these measurement sets is more nonlinear and we expect that the ML model can also work with them. In addition, as the analysis on the objective function, we have a smooth surface, therefore, it does not required high level of accuracy. If it is possible, we will try with more complex systems with a non-smooth searching surface.
We are possible to compute the gradient of the cost, therefore, we can compute a searching algorithm to the optimal point. This could be beneficial because sometimes it takes effort to generate the optimal data in the real plant.
In this project, we tune the model manually, therefore, we aim to use some auto-tuning tools to observe the difference. We expect that we have better results.

Bibliography

[1] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, In Press, 2018.

[2] F. Chollet et al. Keras. `https://keras.io`, 2015.

[3] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[4] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[6] R. Rahman, S. R. Dhruba, S. Ghosh, and R. Pal. Functional random forest with applications in dose-response predictions. *Scientific Reports*, 9(1):1628, 2019.

[7] C. Shang, F. Yang, D. Huang, and W. Lyu. Data-driven soft sensor development based on deep learning technique. *Journal of Process Control*, 24:223–233, 2014.

[8] V. Venkatasubramanian. The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal*, 65(2):466–478, 2019.

[9] A. Verikas, E. Vaiciukynas, A. Gelzinis, J. Parker, and M. C. Olsson. Electromyographic patterns during golf swing: Activation sequence profiling and prediction of shot effectiveness. *Sensors*, 16:592, 04 2016.

[10] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization, 2017.