University of South-Eastern Norway

www.usn.no

FMH606 Master's Thesis 2022

Process Technology

Development of machine learning model for CO_2 capture plants to predict solvent degradation

Sam Narimani

Faculty of Technology, Natural sciences and Maritime Sciences
Campus Porsgrunn

University of South-Eastern Norway

www.usn.no

Course: FMH606 Master's Thesis, 2022

Title: Development of machine learning model for CO_2 capture plants to predict solvent degradation

Number of pages: 72

Keywords: Solvent degradation, Machine Learning, CO_2 capture plant

Student: Sam Narimani

Supervisor: Leila Ben Saad, Ru Yan

External partner: Technology Centre Mongstad (TCM) / Rune Teigland

Summary:

Increasing average temperature of the earth has significantly influenced human's life that many efforts have been carried out to cut the major sources of rising temperature. Among all, emissions and specifically carbon plays a key role as a type of greenhouse gasses in this area. Therefore, capturing CO_2 has become an area of interest for researchers to deal with in the recent years. Carbon capture can be performed by employing several methods, but the most common method is post-combustion since it can be installed for the plants constructed before and there is no need to change any structure in the previous plant. However, there are many problems in the post-combustion carbon plant, such as solvent degradation, which is the most important phenomena as its cost is counted for about 20% of the total operational cost of a carbon capture plant. Therefore, finding the causes can significantly affect the performance of a carbon capture plant. In this study, the focus is on behavior of carbon capture plant in the subject of solvent degradation by using machine learning methods. As this phenomenon has a complex behavior and there are no certain traditional methods or software to simulate the solvent degradation of a carbon capture plant, machine learning would be an efficient method to predict this phenomenon. Although there are many methods in machine learning, several methods such as Artificial Neural Network, Random Forest and Support Vector Regression have been so popular in the literature review. Therefore, these methods are employed to model the solvent degradation in the carbon capture plant. First, lab and online data provided by Technology Centre Mongstad, were cleaned. Secondly, to enhance the performance of the model, feature selection methods like Spearman's and Pearson's correlation methods were applied to decrease the number of features. There are many sources of solvent degradation in a carbon capture plant, but in the current one information about ten sources of solvent degradation is available. Therefore, ten various models with different hyperparameters and features were utilized to reach the best results. Results show that ANN and Random Forest are the best and most promising models to predict the solvent degradation behavior in the plant as compared to the R^2 score that was mainly more than 0.90 for 483 datasets received from TCM.

Preface

This report was written to fulfil a part of master program in Process Technology at the Department of Process, Energy and Environment at University of South-Eastern Norway.

The thesis focus is on prediction of solvent degradation in the carbon capture plant at Technology Center Mongstad by utilizing machine learning methods. The main purpose is finding the best model that can be fitted to the dataset provided by the Technology Center Mongstad from 2017 campaign test.

I would like to express my gratitude to my supervisors, Leila Ben Saad and Ru Yan, for their guidance, support and supervision during this project. Besides, I want to thank Rune Teigland and Blair MacMaster from Technology Center Mongstad for providing data and helping me to better understand solvent degradation phenomenon and data.

Finally, I would like to thank my wife for her patience, support and help me throughout this challenging time of my life.

Porsgrunn, 17.05.2022

Sam Narimani

Contents

Nomenclature

Abbreviation	Description
AI	Absorber Inlet
ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Network
BNN	Biological Neural Network
CCS	Carbon Capture and Storage
CHP	Combined Heat and Power
DG	Depleted Gas
FOLU	Forestry and Other Land Use
GEP	Gene Expression Programming
GHG	Green House Gas
GMDH	Group Method of Data Handling
HEI	1H-Imidazole-1-ethanol
HEPO	4-(2-hydroxyethyl)-2-piperazinone
ILs	Ionic liquids
lAAI	lean Amine Absorber Inlet
MAD	Mean Absolute Difference
MEA	Monoethanolamine
ML	Machine Learning
MLP	Multilayer Perceptron
MLP-NN	Multi-Layer Perceptron Neural Network
NLP	Natural language processing
PG	Product Gas
QSPR	Quantitative Structure-Property Relationship
RBFNN	Radial Basis Function Neural Network
RF	Random Forest
RFCC	Residual Fluidized Catalytic Cracker
RMSE	Root Mean Square Error

SL	Sample Location
SVC	Support Vector Classification
SVM	Support Vector Machine
SVR	Support Vector Regression
TCM	Technology Center of Mongstad
wt%	Weight percent

1 Introduction

1.1 Background and objectives

One of the most major issues in the last decades has been global average temperature increase that is mainly due to Green House Gas (GHG) emissions [1]. GHG mainly consists of water vapor, carbon dioxide, methane, nitrogen and sulfur compounds. Increasing CO_2 avoids enough solar absorption. Therefore, the average temperature goes up and resulting in extinction of many species, ice melting and rising sea level [1].

Total annual GHG emissions from 1970 to 2010 are shown in Figure 1.1. It can be noticed from the figure that the most major emission is CO_2 from fossil fuel and industrial processes counting averagely 60% of total annual GHG. On the other hand, CO_2 Forestry and Other Land Use (FOLU) was decreased over the years. A large part of these emissions belongs to energy sector which stands on 25% of total annual GHG emissions [2].

Increasing GHG will result in increasing average temperature that can lead to many problems like environmental issues. Figure 1.2 shows how the temperature has changed in the period of 1880 to 2018 in the ocean and land. As it can be seen, the temperature anomaly experienced a high increase in both land and ocean over the period since GHG has increased [3]. One of the solutions to prevent from increasing temperature is carbon capture. It has been reported by International Energy Agency that Carbon Capture and Storage (CCS) might reduce up to 17% of the emitted CO_2 by 2050 [1].

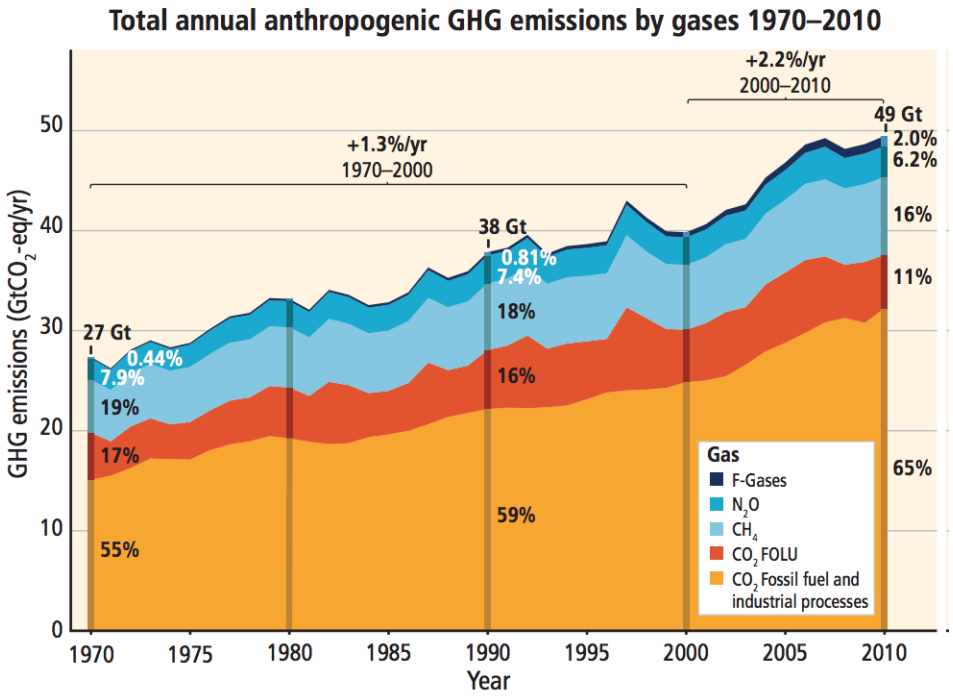


Figure 1.1: Total annual GHG emission between 1970 and 2010 [2].

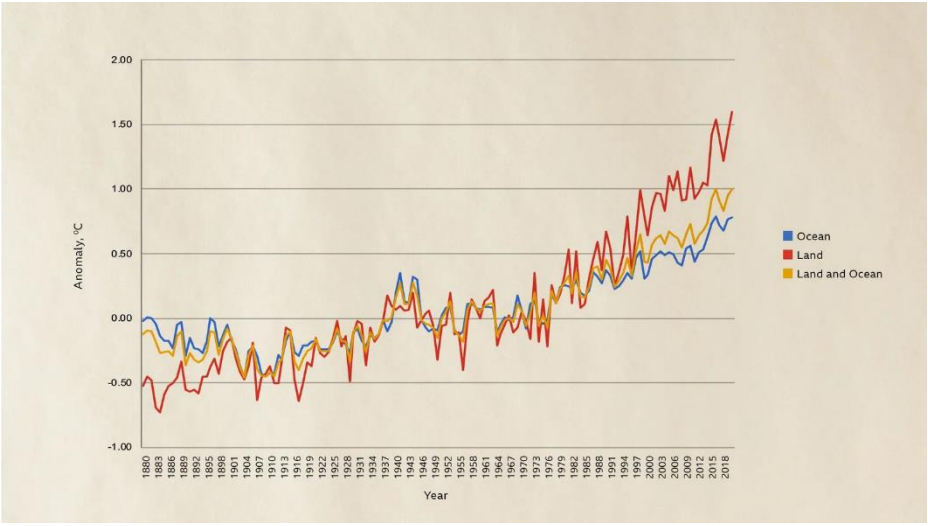


Figure 1.2: Temperature anomaly during the years [3].

Understanding the phenomena of carbon capture plant needs more complex models rather than traditional solutions. Artificial intelligence and machine learning (ML) methods have been utilized to predict many intricate phenomena such as carbon capture plant cases. The present study is predicting solvent degradation in the carbon capture with absorption at Technology Center of Mongstad (TCM). TCM is the largest center in the world to test carbon capture phenomenon. TCM started its journey 2006 when Norwegian government and Equinor agreed to establish the largest test plant for carbon capture and has been operating since 2012 in industrial scale [4]. A wide range of data has been collected by 1000 online instruments in the amine plant [4]. There are also several sampling points in the plant which analyze liquid components for further data [4].

In this report, the focus is on solvent degradation as a major problem in the carbon capture plants. Since this phenomenon has a very complicated behavior and there is no exact formula or classic regression model to fit on, the use of machine learning methods to predict the phenomenon over a campaign test is investigated. The purpose is finding a better model to look for a pattern in degradation of Monoethanolamine (MEA) solvent. Three methods of Support Vector Regression (SVR), Random Forest (RF) and Artificial Neural Network (ANN) are used. The results based on machine learning metrics are shown, discussed and the suitable models are chosen. Task description is also presented in the Appendix A.

1.2 Thesis outline

The rest of this report is structure as follows. Chapter 2 presents an overview of different carbon capture methods and solvent degradation in the plant as well as machine learning concepts. Literature review is also investigated to explain the former research in solvent degradation and machine learning applications in capturing carbon plants. In chapter 3, the system description and data pre-processing are explained. The following chapter deals with the results of all methods and present some discussions and possible improvements. The last chapter concludes this study and presents recommendations for future research studies.

2 Literature

In this chapter, carbon capture methods including pre-combustion, oxy-fuel combustion and post-combustion will be investigated. In addition, solvent degradation phenomena in post-combustion carbon capture, machine learning methods used in the models and feature selection are presented in detail. Finally, literature related to the solvent degradation and machine learning in the carbon capture plant will be introduced.

2.1 Carbon capture methods

There are three main methods to capture carbon namely pre-combustion, oxy-fuel and post-combustion capturing. In the following, a brief explanation of each method is presented [5].

2.1.1 Pre-combustion

In pre-combustion carbon capture, the carbon is taken from combustible gases before combustion [6]. In this method, the carbon is captured by applying three reactions between methane, water and pure oxygen. First, in steam reforming, the methane reacts with high pressure steam to produce hydrogen and CO. Then, the pure oxygen separated from air reacts with methane gas to form hydrogen and CO again. Finally, CO is converted to CO_2 by passing over water and CO_2 is captured [5]. All three reactions are described in the relation 2.1.

$$\begin{aligned} &CH_4 + H_2O \rightarrow CO + 3H_2 \\ &CH_4 + 0.5O_2 \rightarrow CO + 2H_2 \\ &CO + H_2O \rightleftharpoons CO_2 + H_2 \end{aligned} \tag{2.1}$$

The hydrogen produced in these reactions is used for power generation. Although this method removes carbon before combusting the gas, it has significant energy demand to separate oxygen from air. Many research studies have been carried out in increasing the efficiency of pre-combustion carbon capture and separation techniques, however, high energy demand in the method still exists. Figure 2.1 shows an integrated gasification combined cycle which used pre-combustion carbon capture [5, 7].

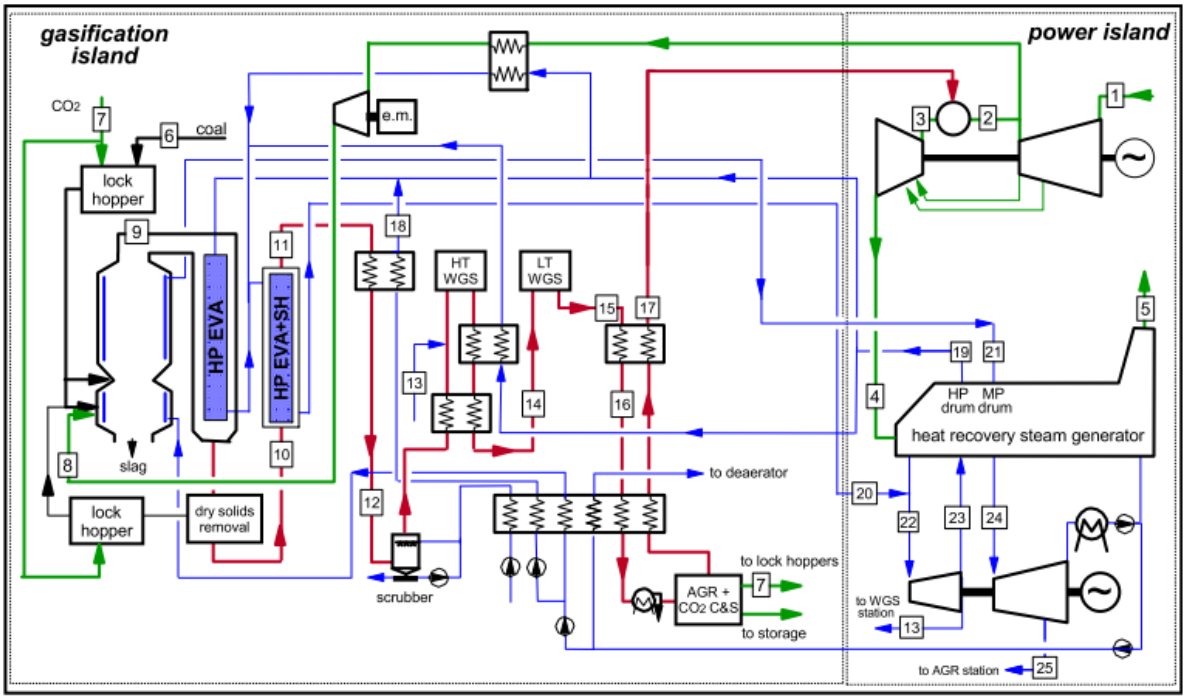


Figure 2.1: Integrated gasification combined cycle with pre-combustion capturing method [7].

2.1.2 Oxy-fuel combustion

Oxy-fuel combustion is the process of the fuel combustion in pure oxygen. As shown in Figure 2.2, the separated oxygen from air is added to fuel and reaction happens in the boiler to produce CO_2. In this method, purity of produced CO_2 is significantly high and emissions such as NO_x (either NO or NO_2) reduce since the burning is nitrogen free. The only disadvantage of this method is production cost of oxygen and carbon compression expenses which are expensive [5, 6].

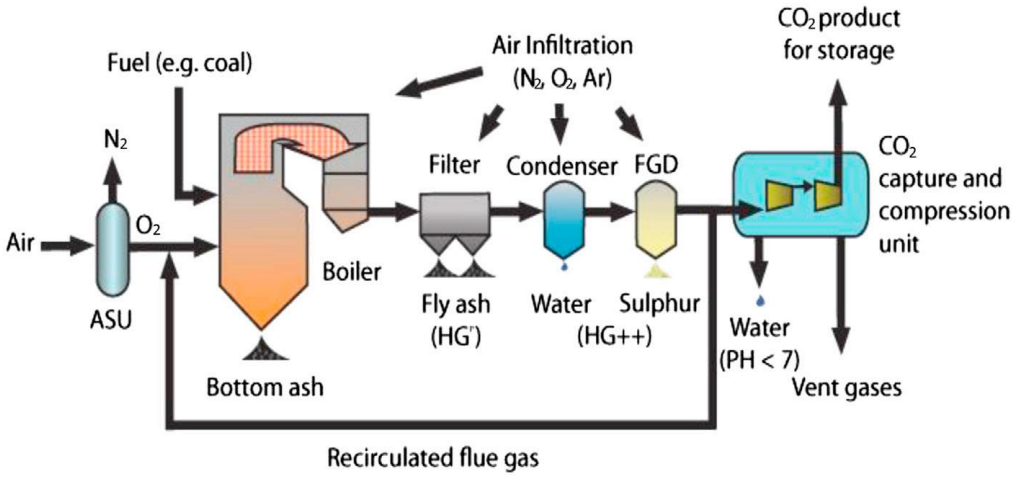


Figure 2.2: Oxy-fuel combustion system [6]

2.1.3 Post-combustion

As its name shows, this method is used after combustion meaning that it can be used for the existing plants [8]. This is the most important advantage of post-combustion rather than the other two carbon capture methods [5]. As shown in Figure 2.3, the flue gas is entered to the absorber and after crossing with solvent, rich solvent leaves the absorber. The rich solvent gains some heat to increase its temperature by passing a heat exchanger and finally, it enters the stripper. Rich solvent is heated to release CO_2 and then leaves the stripper to come back to the absorber and the cycle is done for the operational time of the plant [5, 9].

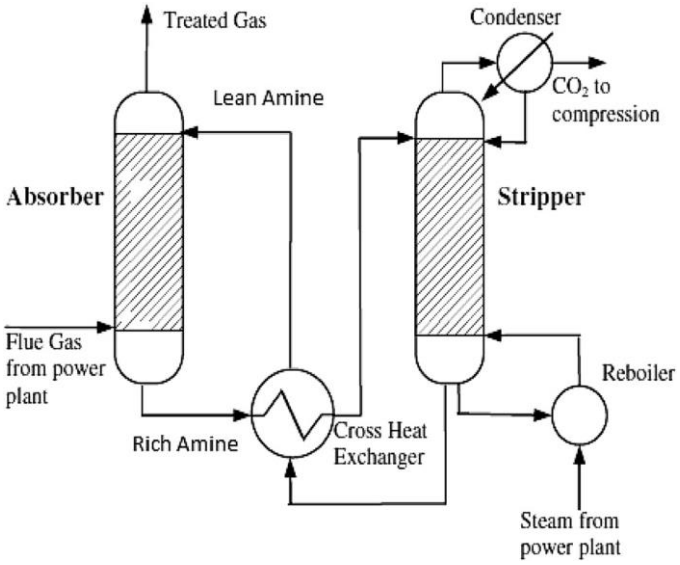


Figure 2.3: Simplified process in post-combustion carbon capture [9].

CO_2 concentration in the flue gas is approximately 4-20 percent which is relatively low. Variation in percentage of carbon dioxide depends on what kind of fuel has been used so that if coal is burning fuel, flue gas can contain about 14% carbon dioxide [5].

In amine based post-combustion carbon capture, the overall reaction between MEA, as solvent, and CO_2 can be shown in the reaction 2.2 [1].

$$CO_2 + 2RNH_2 \rightleftharpoons RNHCO_2^- + RNH_3^+ \tag{2.2}$$

The reaction 2.2 shows that to end up the reaction, one mole CO_2 needs 2 MEA moles. The ratio between CO_2 and MEA moles is described as loading in the Equation 2.3. The Loading is usually between 0.2 to 0.5 as the absorbed CO_2 into the MEA [1].

$$\alpha = \frac{\dot{n}_{CO_2}}{\dot{n}_{MEA}} \tag{2.3}$$

2.2 Solvent degradation

Solvent degradation is one of the common issues in carbon capture plants. Several research studies have proved that it mainly occurs due to presence of oxygen, high temperature and impurities such as NO_x and SO_x [1]. Degradation has multiple disadvantages such as lower solvent capacity, corrosion, emissions and foaming [1]. The largest amount of degradation is

due to the volatility of solvent in the absorber [1]. However, this issue has been almost solved by using water wash [1]. Water wash is a unit that absorbs the degraded solvent in absorber and sends volatile MEA to the absorber to again be used in the process [1].

2.2.1 Oxidative degradation

Oxidative degradation mainly happens in the presence of oxygen in flue gas. One of the factors increasing this kind of degradation is diffusivity of oxygen in the solvent [1]. Temperature and oxygen rate are the effective factors in oxygen diffusivity into the solvent [1]. As temperature or oxygen rate in flue gas increases, oxidative degradation rate goes up [1]. For instance, natural gas-based power plant can have more oxidation degradation rather than coal based as natural gas power plants needs more air meaning higher oxygen rate [1].

As discussed earlier, oxidative degradation mainly occurs in the absorber. However, it might exist small portion of dissolved oxygen in the rich solvent exiting the absorber. Reactions 2.4 and 2.5 show the main reactions resulting oxidative degradation in the absorber [1].

$$MEACOO^- + MEAH^+ + O_2 \rightarrow 2CH_2O + CO_2 + 2NH_3 \quad (2.4)$$

$$MEACOO^- + MEAH^+ + 2O_2 \rightarrow 2C_2H_2O_2 + 2H_2O + CO_2 + 2NH_3 \quad (2.5)$$

As these reactions demonstrate, ammonia is one of the most important products of oxidative degradation.

2.2.2 Decomposition or thermal degradation

Thermal degradation takes place due to reaction between CO_2 and the solvent at high temperature [1]. This type of degradation as mentioned happened in high temperature sections such as stripper, hot lines, and heat exchanger [1]. Therefore, it is more probable to find thermal degradation in stripper rather than other parts of carbon capture plant. In addition, when loading factor α increases, thermal degradation increases as the system gains more CO_2 [1].

2.2.3 Degradation due to impurities

The main impurities causing solvent degradation are SO_x and NO_x [1]. In most of carbon capture plants, the amount of these impurities is low and therefore, it is not considered in this study [1].

Solvent degradation will be more complex when several types of degradation happen simultaneously or depend on each other. Therefore, the interaction between two main groups of degradation, oxidative and thermal, should be considered.

2.3 Machine learning

Machine learning can be defined as learning of the system from past to forecast the future [10]. To find the relationship between the data, many algorithms have been suggested such as ANN, decision tree, Adaptive Neuro Fuzzy Inference System (ANFIS) and Support Vector Machine (SVM).

Machine learning and artificial intelligence are the most important tools used in the field of data analysis and technology. Many ML applications such as weather forecasting and face recognition are used in different industries to achieve the best and effective results [10].

Machine learning can be divided into three main categories supervised, unsupervised and reinforcement learning. In the following, these three classes are briefly described [10].

2.3.1 Supervised learning

In this type of ML method, algorithms show the relationship between the features and targets that are the observations [10]. Targets or outputs are also referred to labeled data [10]. Supervised learning can also be divided into two groups of classification and regression problems [10]. In classification problems, dataset is broken to limit number of classes. On the other hand, in regression problems, features have a numeric relationship with the targets that is in a continuous spectrum [10].

2.3.2 Unsupervised learning

In unsupervised learning, there is no need to supervise the model and it allows the model to find the pattern in data. One example is density estimation and finding the concentration of similar variables in dataset [10].

2.3.3 Reinforcement learning

Reinforcement learning is considered as one of the best methods as it tries to reach the best output by self-optimization [5].

Figure 2.4 illustrates these three categories with their applications in more detail.

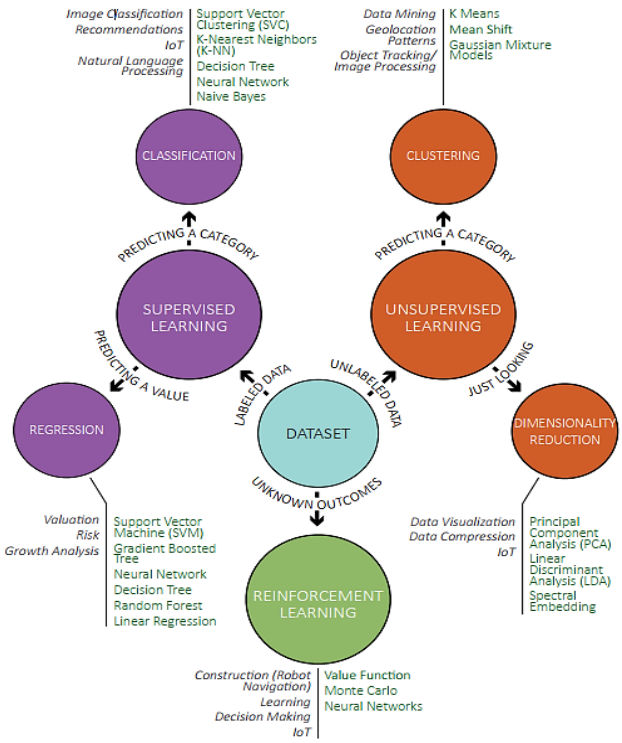


Figure 2.4: Different types of ML categories [10].

2.3.4 Machine learning methods

Machine learning methods used in this study are ANN, RF and SVR. This selection is as a result of their popularity and desirable accuracy in former research studies based on literature. In the following sections, these methods are briefly explained.

2.3.4.1 Support Vector Machine (SVM)

Support Vector Machine or SVM is a kind of machine learning method to generalize nonlinear problems by minimizing the error bound to reach better performances [11].

It has two main categories namely SVR and Support Vector Classification (SVC). This method has been primarily presented by Vapnik and is capable to be utilized when the problem has many features [11, 12]. The most important feature of SVR is that the model is only dependent on a subset of training set as the model cost function does not accept any training data being close to the prediction of the model [11]. SVR can be referred to the common and practical form of SVM [12].

Suppose there is a set of training data as $\{(x_1, y_1), (x_1, y_2), ..., (x_n, y_n)\}$. Figure 2.5 illustrates a scheme of SVR problem. The target is finding a function like f(x) so that the error between f(x) and y_i would be less than epsilon. Therefore, the problem is minimizing the norm of parameters that satisfy the conditions of the model which is model error. Equation 2.6 shows that f(x) can be written as a summation of parameter product and inputs, and a bias. Relation 2.7 also shows the basic minimizing problem that SVR model solves to achieve minimum error [11].

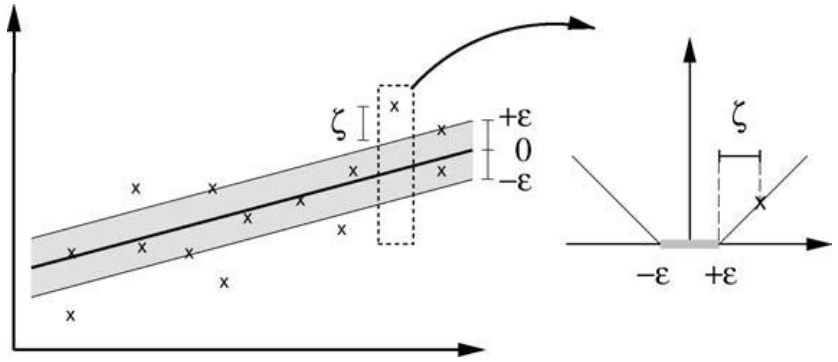


Figure 2.5: The margin loss for SVR [11].

$$f(x) = wx_i + b \tag{2.6}$$

$$\min : \frac{1}{2}\|w\|^2$$
$$\text{Subject to}: \begin{cases} y_i - wx_i - b \le \varepsilon \\ y_i - wx_i - b \ge \varepsilon \end{cases} \tag{2.7}$$

2.3.4.2 Random Forest (RF)

Random Forest is a set of tree predictors depending on random vector values with the same distribution for trees [13]. Generalization of Random Forest relies on each tree strength and relation between trees [13]. This method was firstly developed by Breiman in 1996 with application for both regression and classification problems [14]. To better understand the RF model, consider the dataset in the Table 2.. there are four features namely Outlook, HWDone, Weekend and Play along with eight datasets [14].

Table 2.1: Training data [14].

Outlook	HWDone	Weekend	Play
Sunny	True	True	Yes
Sunny	True	False	Yes
Sunny	False	True	Yes
Sunny	False	False	No
Rainy	True	True	Yes
Rainy	True	False	No
Rainy	False	True	Yes
Rainy	False	False	No

As it is seen in the Figure 2.6, there are three trees, which will be referred to n_estimator lately, to predict feature Play which is the target. This is based on Yes and No conditions. For instance, in tree B, if it is rainy the prediction should show Don't Play, otherwise the tree will investigate other conditions such as HWDone and so on [14]. Random Forest models follow randomization in each tree and split the best node to break that can improve the accuracy of the model. It has been reported that RF accuracy is mostly better than decision tree and SVM models in the application of carbon capture [13, 14].

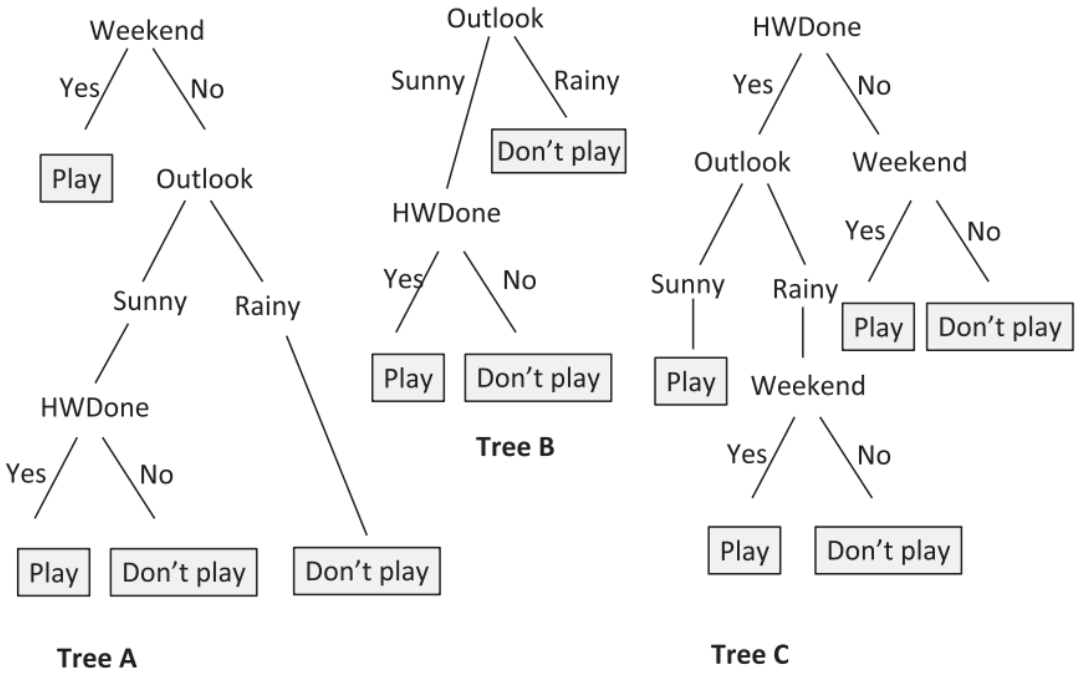


Figure 2.6: Random Forest example with three trees [14].

Several hyperparameters affect efficiency of RF method such as number of trees (n_estimator), the maximum depth of tree (max_depth), the number of the features for best splitting (max_features), minimum sample number for splitting internal node (min_samples_split) and minimum sample number at one node (min_samples_leaf) [15]. Therefore, hyperparameters should be tuned to reach a better result. The meaning of hyperparameters is shown in the Figure 2.7 to Figure 2.9.

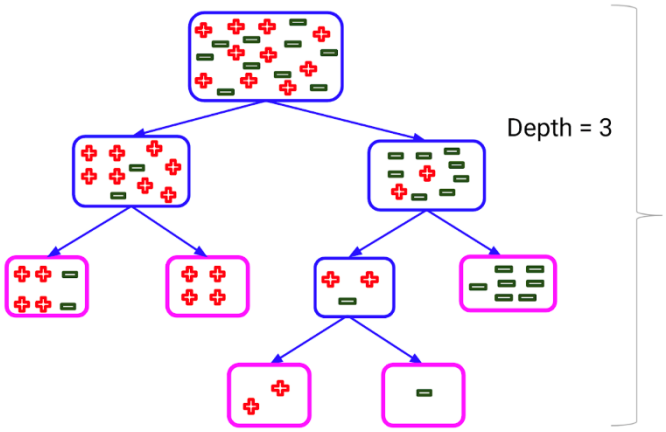


Figure 2.7: max_depth in RF method [15].

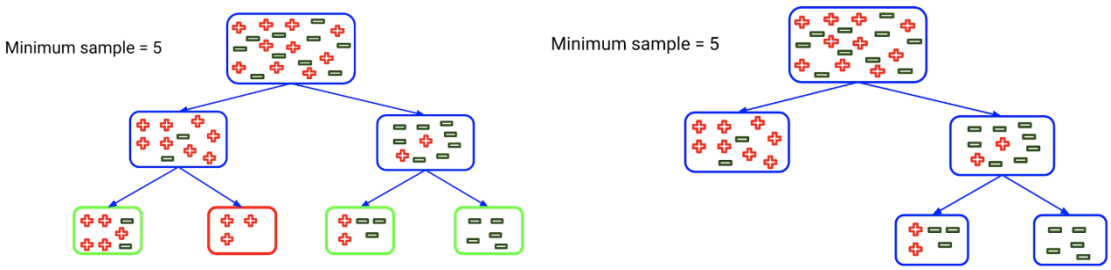


Figure 2.8: min_samples_leaf in RF method [15].

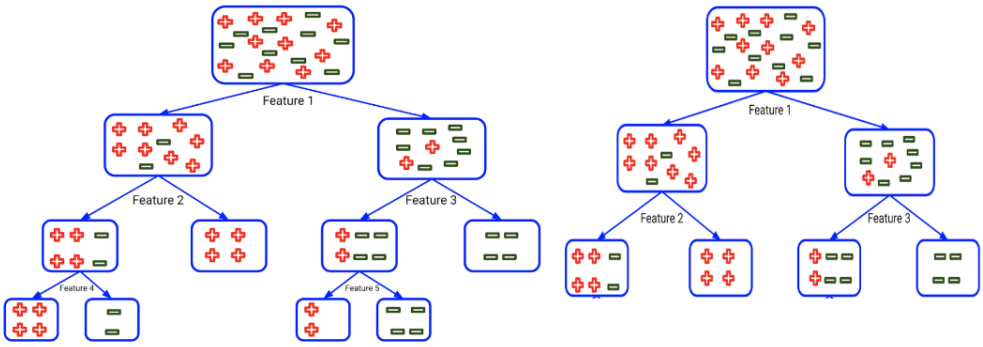


Figure 2.9: min_sample_split and max_features in RF method [15].

2.3.4.3 Artificial Neural Network (ANN)

ANN has become one of the most popular models in ML during the last decade thanks to its high-speed processing provided for big data. ANN has many applications such as predicting value, face recognition and Natural language processing (NLP) in different industries. Although ANN is utilized in various fields, there is major necessity to address some problems and solve them before generalizing models. For example, selection of data set and features, data accuracy, data pre-processing and cleaning, validation of data can be mentioned [16].

The name of neural network comes from a biological phenomenon. The brain of the human contains many neurons which each one performs a special function based on received information from its environment. Figure 2.10 illustrates a neuron consisting of soma, axon and dendrites. Axons and dendrites are like input and output in neural network while soma reflects the neuron. Synapses connect axons to dendrites and when a signal is given to a neuron, synapses can grow or lessen the electrical potential which are similar to weights in neural network [17]. Although there are many similarities between ANN and Biological Neural Network (BNN), there are also several differences. The processing speed is the one of major differences between ANN and BNN which is faster in ANN. Moreover, the process can carry out big data in parallel way in BNN, though it operates sequentially [18].

Figure 2.11 illustrates an ANN includes input, hidden and output layers. Input layer consists of given data for the model while output layer is the model targets. Hidden layer is the layer that connect inputs to output. Each layer consists of some nodes which perform a special function based on received information. Output of each layer is specific activation function that should be defined for every application separately as there in no certain way to examine. There are also some weights in each layer that should be optimized in the process of fitting the model [16, 17].

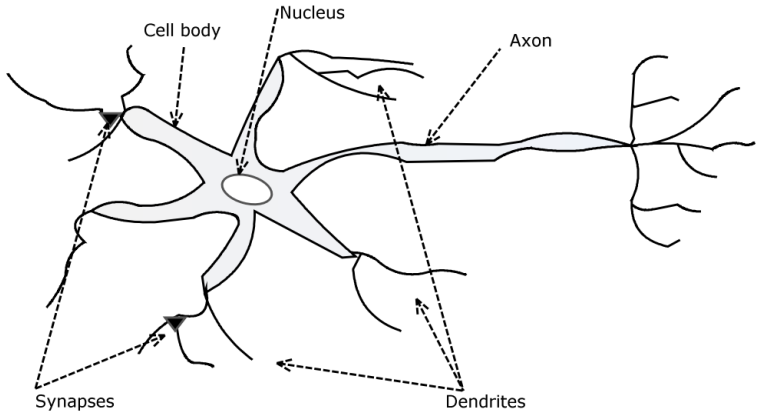


Figure 2.10: Structure of a neuron in brain [17].

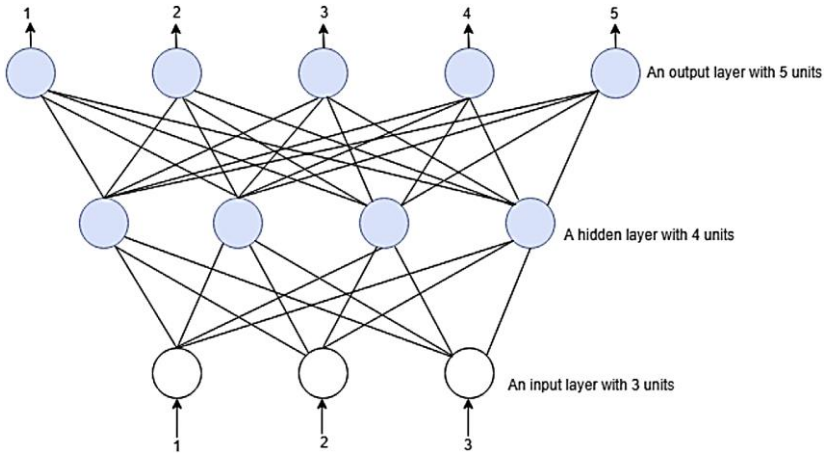


Figure 2.11: ANN architecture including input, hidden and output layers with different nodes [16].

To better understand the performance of ANN, consider a simple ANN with just one neuron shown in Figure 2.12. In the figure, x_i represents input while y is output. Besides, w_i's and b are weights and bias in ANN model. Relation between input, output, weights and bias are

described in the relation 2.8. There are several activation functions that should be tested for each problem and find the suitable one, though Rectified Linear Units (ReLU) and tanh are the common activation functions [17].

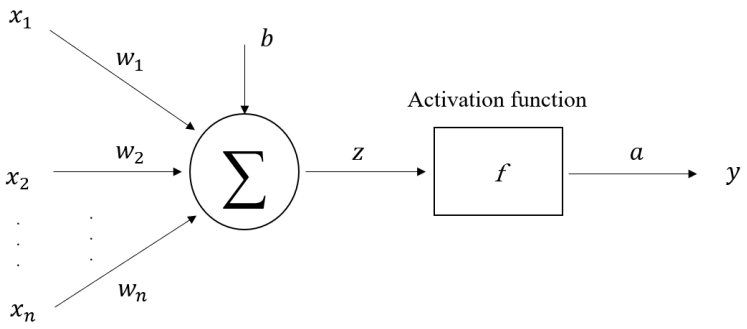


Figure 2.12: Simple ANN performance with one neuron

$$z = \sum_{i=1}^{n} w_i x_i + b$$
$$a = f(z) \tag{2.8}$$
$$y = a$$

2.4 Feature selection

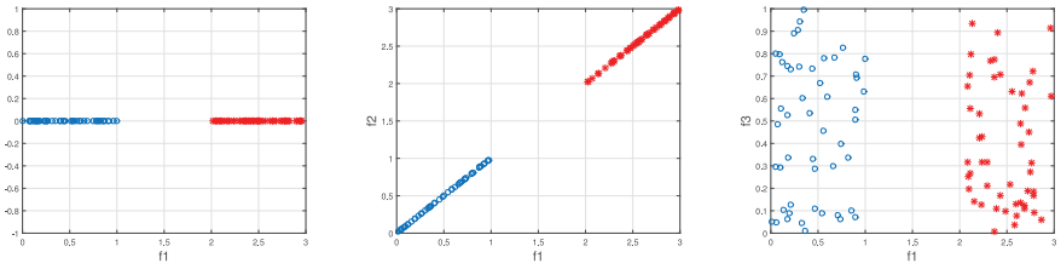
Feature selection is a kind of pre-processing method to efficiently prepare data for machine learning problems [19]. Its target is creating simpler models and enhancing model performance. In the case of high dimensional data, there is a well-known issue as dimensionality curse causing data sparser [19]. On one hand, increasing number of features can lead the model to overfit that may affect model performance to predict unseen data. One the other hand, high dimensional feature need memory storage requirement and consequently computational cost increases [19].

Dimensionality reduction is a crucial method and tool to overcome the mentioned problems. It can be classified into two main parts of feature extraction and feature selection. Feature extraction represents original dimensional features in a lower dimensional feature where the interesting parts of data are captured [20]. In feature selection, the most relevant features as a subset of data are chosen to construct the model [19].

Both techniques are utilized to enhance the performance, reducing memory storage and constructing more effective model. Therefore, these methods are referred to as efficient dimensionality reduction methods. When comprehensible features are not included in raw data to an ML algorithm, feature extraction is elected. As the feature extraction creates new features, latter analysis is difficult as we are not able to sustain the physical meaning of these new features. Therefore, feature selection retains physical meanings of the primary features by holding some of the main features. Then, many researchers prefer to employ feature selection in their models instead of feature extraction [19].

In the world of the real data, many improper, redundant and noisy data can be seen. Removing these features enhances model performance and decreases the computational cost. As Figure 2.13 Figure (a) shows that feature f_1 is a relevant feature and can separate two clusters while Figure 2.13 (b) illustrates redundant parameters of f_1 and f_2 because they are correlated.

Moreover, in Figure 2.13 (c), f_1 is an irrelevant feature since it cannot discriminate two clusters. Therefore, the model performance would not be negatively affected in case of removing features f_2 and f_3 [19].



(a) relevant feature f_1 (b) redundant feature f_2 (c) irrelevant feature f_3

Figure 2.13: Different types of features (relevant, redundant and irrelevant) [19].

Feature selection can be classified in three categories in strategy point of view namely filter, wrapped and embedded [19]. In the following, these methods will be briefly explained.

In filter strategy, feature selection is independent of any learning algorithm and is just based on data characteristics. This method is very computationally efficient while due to lack of learning algorithms evaluating the feature selection process, the selected feature may not be the optimal target for the model. There are several techniques in filter methods like Chi-square Test, Fisher's Score, Correlation Coefficient, Variance Threshold, Mean Absolute Difference (MAD) and Dispersion ratio which Correlation Coefficient is one of the most popular ones [19].

Wrapped method is based on an ML algorithm and tries to find the possible correlation between features until the optimal features are obtained. Forward and backward propagation are the most common methods used in the wrapped method. Wrapped method process is shown in the Figure 2.14 [21].

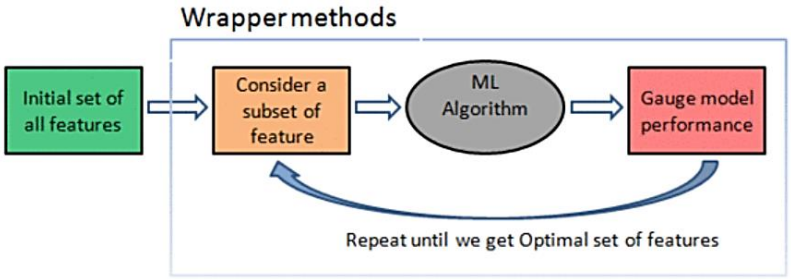


Figure 2.14: Wrapped method algorithm [21].

Embedded strategy is a combination of filter and wrapped method. Elastic Net and Ridge regression are two methods in this category of feature selection. In this study, filter method is utilized as feature selection strategy for further process [19].

The correlation coefficient, which is originally one of the techniques in filter strategy, is a statistical technique measuring the relation between two variables. The most usable correlation methods are Pearson's product moment correlation coefficient and Spearman's rank correlation coefficient. Pearson's correlation coefficient shows a linear relationship between two variables while Spearman's rank correlation coefficient is a non-linear rank coefficient [22].

The produced coefficient of both methods is between -1 and 1 which negative and positive signs show reverse and direct relationship between variables. As the absolute number becomes closer to zero, there is less relation between two variables [22].

2.5 Literature review on solvent degradation and machine learning approaches

In this section, previous studies are presented in both solvent degradation and machine leaning methods used in the carbon capture plants.

2.5.1 Solvent degradation

Léonard et al. studied the effect of solvent degradation in a post-combustion capture plant and introduced a model to increase solvent degradation. They showed that the model has a potential to reduce solvent loss about 11% rather than primary case. Their results also proved that the major solvent loss happened in absorber that is oxidative degradation [23].

In 2021, Seo et al. proposed Ionic liquids (ILs) as the major solvent instead of MEA to reduce thermal solvent loss. ILs are more expensive than MEA, but they showed the effect of thermal degradation is considerably more important than using traditional solvent as the system worked in lower temperatures. They also showed that residence time of the solvent, as an important factor in solvent degradation, significantly decreased by using ILs [24].

Solvent and emission behavior were monitored to estimate solvent degradation and emissions sources at TCM in long operational period. This research showed that Ammonia emission is a type of emission indicating solvent loss in the carbon capture plant as well as other types of degradation such as heat stable salt and organic loss [25].

A thorough investigation was carried out by Anne K Morken et al. to show the source of solvent degradation in the carbon capture plant at TCM. They showed that solvent can be degraded by emitting via absorber, stripper, in Ammonia formation in product flue gas, wash water and due to leakage. Figure 2.15 shows the solvent degradation types seen at TCM for 1960 hours operational time. In this research, they also showed that the color of the solvent is an indication for the solvent degradation that solvent color changed from colorless to yellow during the time due to solvent loss. Figure 2.16 illustrates the result of color and solvent degradation dependency during the time. Finally, a table was introduced to show the fraction of product produced per unit of degraded solvent (MEA). Table 2.2 shows that Ammonia is the most source of degradation for carbon capture plant with MEA solvent [26].

Table 2.2: Fraction of different types of degradation per solvent loss [26].

Product	Mole produced/mole amine lost	Mole produced/mole amine lost [a)]
Ammonia	0.67	0.67
Total formate + HEF	0.03	0.12
Oxalate + oxylamide	0.003 [b)]	0.01
Nitrate	0.005	0.01
HEI	0.01	0.06
HeGly	0.04	0.05
HEHEAA + HeGly + HEPO	0.12	-

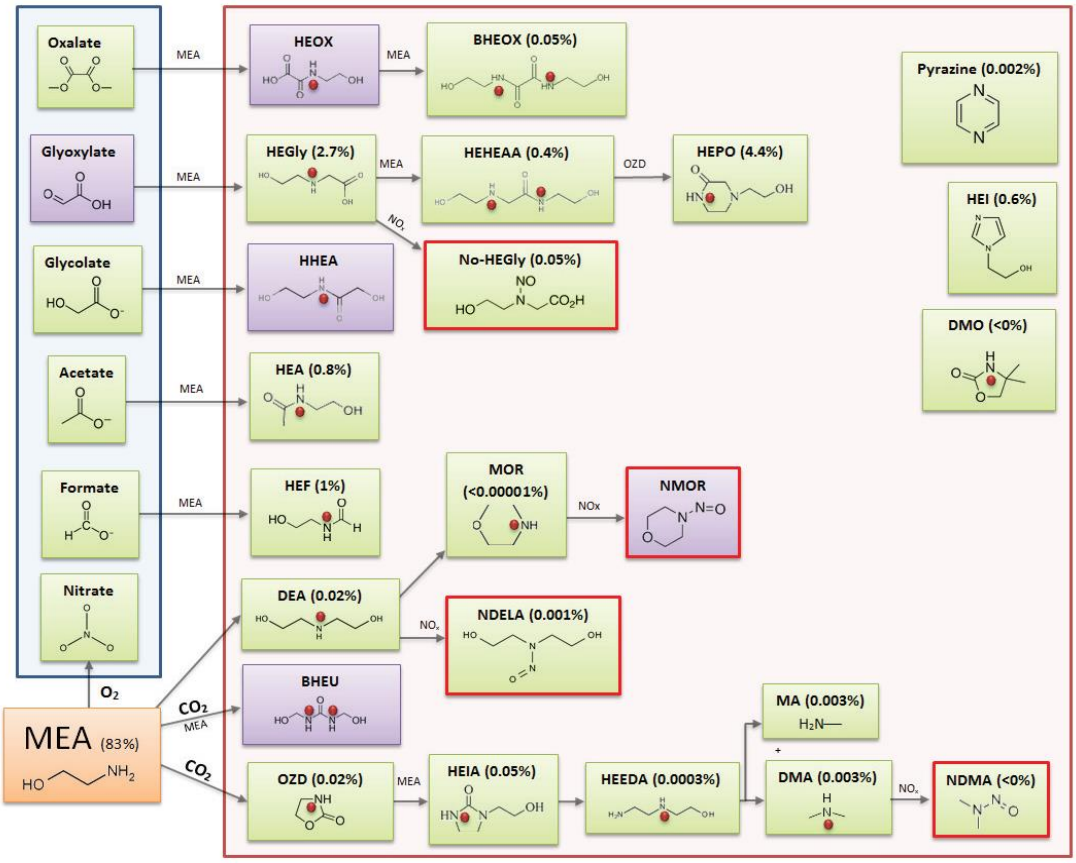


Figure 2.15: Different types of solvent degradation at TCM carbon capture plant [26].

Figure 2.16: Solvent color form colorless to yellow over the time [26].

Fløet al. studied the thermal reclaiming to reduce accumulated solvent loss at TCM. They showed that some physical characteristics of solvent can be changed during the solvent degradation. For instance, Figure 2.17 shows that viscosity of the lean solvent recorded by different samples increases during the time as solvent is degraded [27].

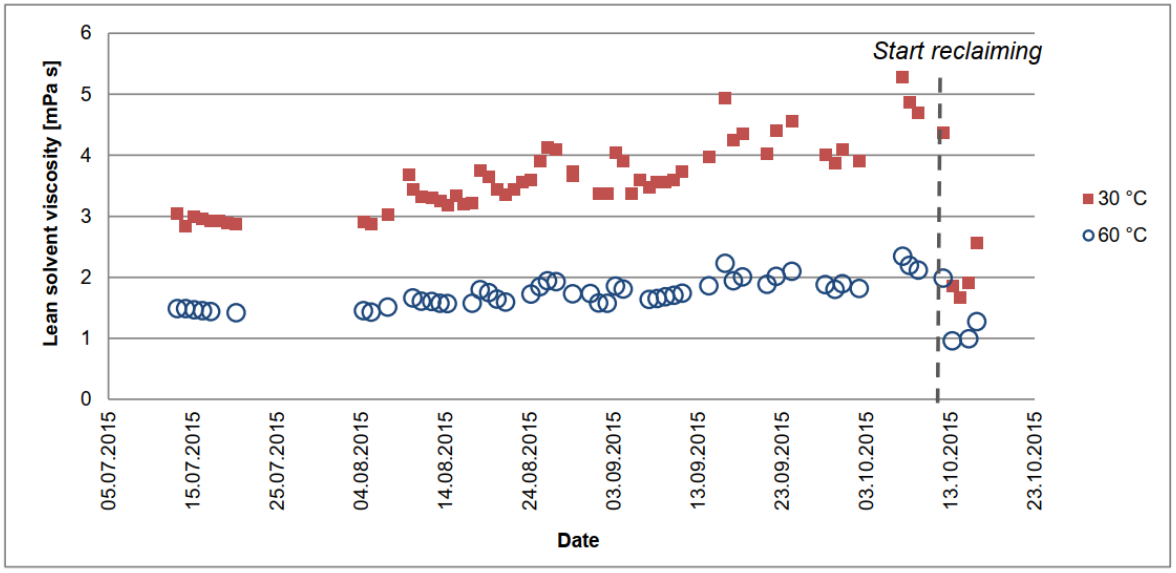


Figure 2.17: Viscosity change in lean solvent over the time [27].

Cuccia et al. presented a review on solvent degradation and showed that a major part of degradation is in form of aldehydes and organic acids [28].

The blend 1MPZ-PZ-Water was presented as a substitute for MEA in a lab scale carbon capture by Cuccia et al. This experiment showed that the degradation of new solvent is around 22% for 900-hour operational time [29].

In 2019, Flø et al. investigated compatibility of the metal materials used in the carbon capture plant to avoid corrosion due to MEA solution. All material except CS235 was introduced as compatible material with MEA solvent during the long operational period [30].

A new lab scale CO_2 capture benchmark was presented by Bontemps et al. to measure solvent degradation. Results were shown acceptable agreement with solvent degradation in industrial application [31].

In 2017, a study in solvent degradation was carried out on three CO_2 capture plants namely TNO, EnBW and ENEL located in Netherlands, Germany and Italy, respectively. The results showed that Ammonia emissions is a cause of solvent degradation. Time residency was also introduced as an effective parameter in high temperature parts of carbon capture plant [32].

In 2015, impact of operational parameters on solvent degradation was studied in a lab scale with MEA solvent. Results revealed that oxidative degradation is the main solvent loss in a capture plant [33].

2.5.2 Machine learning

Rahimi et al. presented a ML implementation in carbon capture plant to reach a smart plant in the future. They showed that ML methods can find the optimal energy demand of a carbon capture in case data acquisition has been effectively carried out. As the development of the model also depends on the data, selecting the practical data demonstrating relation between the parameters is also vital. Figure 2.18 shows the suggested ML method using ANN for carbon capture plant. In this model, inputs are flue gas temperature entering absorber, fraction of CO_2,

flow rate of the flue gas and the absorbent. On the other hand, the targets are predicting carbon capture rate and duty of reboiler [34].

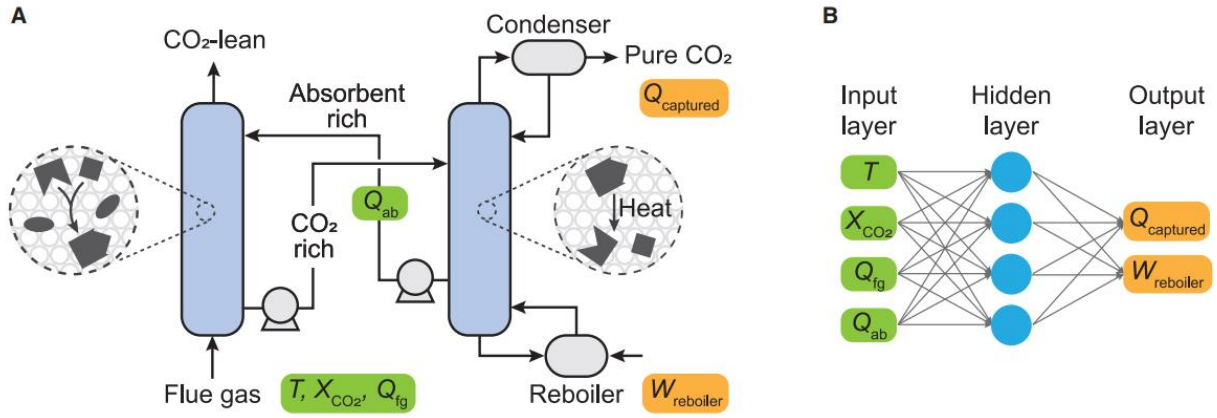


Figure 2.18: An example for ML methods implementing in carbon capture plant [34].

Amar et al. utilized three ML methods namely Multilayer Perceptron (MLP), Gene Expression Programming (GEP) and Group Method of Data Handling (GMDH), to predict CO_2 viscosity in high temperature. In this research, 1124 experimental data was used, and inputs were temperature and density of CO_2. Results showed that MLP is the most effective model to predict the viscosity so that R^2 and Root Mean Square Error (RMSE) showing 0.9999 and 0.0012 mPa, respectively [35].

In 2019, Dureckova et al. studied the use of ML methods in pre-combustion carbon capture plants to predict CO_2/H_2 separation features and CO_2 working capacities. They proposed Quantitative Structure−Property Relationship (QSPR) model with R^2 values 0.872 and 0.944 for working capacity of CO_2 and CO_2/H_2 [36].

Menad et al. modeled solubility of CO_2 with 570 data sets by using MLP and Radial Basis Function Neural Network (RBFNN) methods. The results proved that RBFNN model had higher accuracy rather than MLP [37].

Mesbah et al. also presented Multi-Layer Perceptron Neural Network (MLP-NN) model to predict the CO_2 solubility with a wide range of temperature and pressure data as features. They showed that accuracy of the model is so high and therefore, the model is feasible to be generalized [38].

Another approach was carried out to optimize the pressure oscillation in carbon capture plant with adsorption. In this research, ANN was applied, and the results indicated that the average mean square error was around 10^{-8} [39].

Previous research studies in the literature have proved neural network models are promising models to predict targets in carbon capture plant. However, there is also a limitation in the neural network models that can result poor accuracy. The reason stands on trapping this kind of model in local minimum when the data and algorithm are trained, and the model can overfit the noise in the process of training data [40]. Many techniques have been suggested to solve this problem such as early stopping [41], other types of regularization like dropout [42] and Bayesian learning [43].

Since there is no study about prediction of solvent degradation by utilizing ML methods, the main purpose of this study is to predict a continuous value for all types of degradation appearing in the carbon capture plant by using three ML methods and finding a better model.

3 System description and data pre-processing

In this chapter, a brief description of the carbon capture plant is presented. In addition, machine learning steps for solvent degradation and data pre-processing are investigated.

3.1 System Description

This study focuses on carbon capture plant at TCM located in Mongstad next to the Equinor refinery. Figure 3.1 indicates the process flow diagram for this plant. As shown in the figure, there are two sources of flue gas feeding this plant namely Combined Heat and Power (CHP) and Residual Fluidized Catalytic Cracker (RFCC). The CHP flue gas has about 4% CO_2 while the RFCC contains around 14% CO_2. Both flue gas sources cool down before entering absorber. CO_2 in flue gas is removed by passing over lean solvent in the absorber. Besides, in the top of the absorber, there are columns of water washes to absorb evaporated amine and ammonia and then, feed to the bottom of the absorber where the rich solvent is. Finally, the depleted flue gas, which contains small emissions, is released from top of the absorber. On the other side, rich solvent containing CO_2 heats up by passing from a heat exchanger to reach the temperature of the stripper. In stripper, trapped CO_2 in rich solvent is released after gaining some heat from reboiler and therefore, rich solvent is converted to lean solvent. CO_2 is collected in top of the stripper and cooled down for further storage [26].

In this study, one of the TCM campaign test data for carbon capture with MEA solvent is used. This campaign test was begun from the first of July 2017 and lasted to the first of March 2018 (approximately eight months). Total operational days during this campaign was 162.5 based on flue gas circulation. Figure 3.2 shows operational time of plant in each day over the time period. In y axis, one and zero present working and stopping of the process in the plant, respectively. In addition, Figure 3.3 illustrates total working days in the plant during the time.

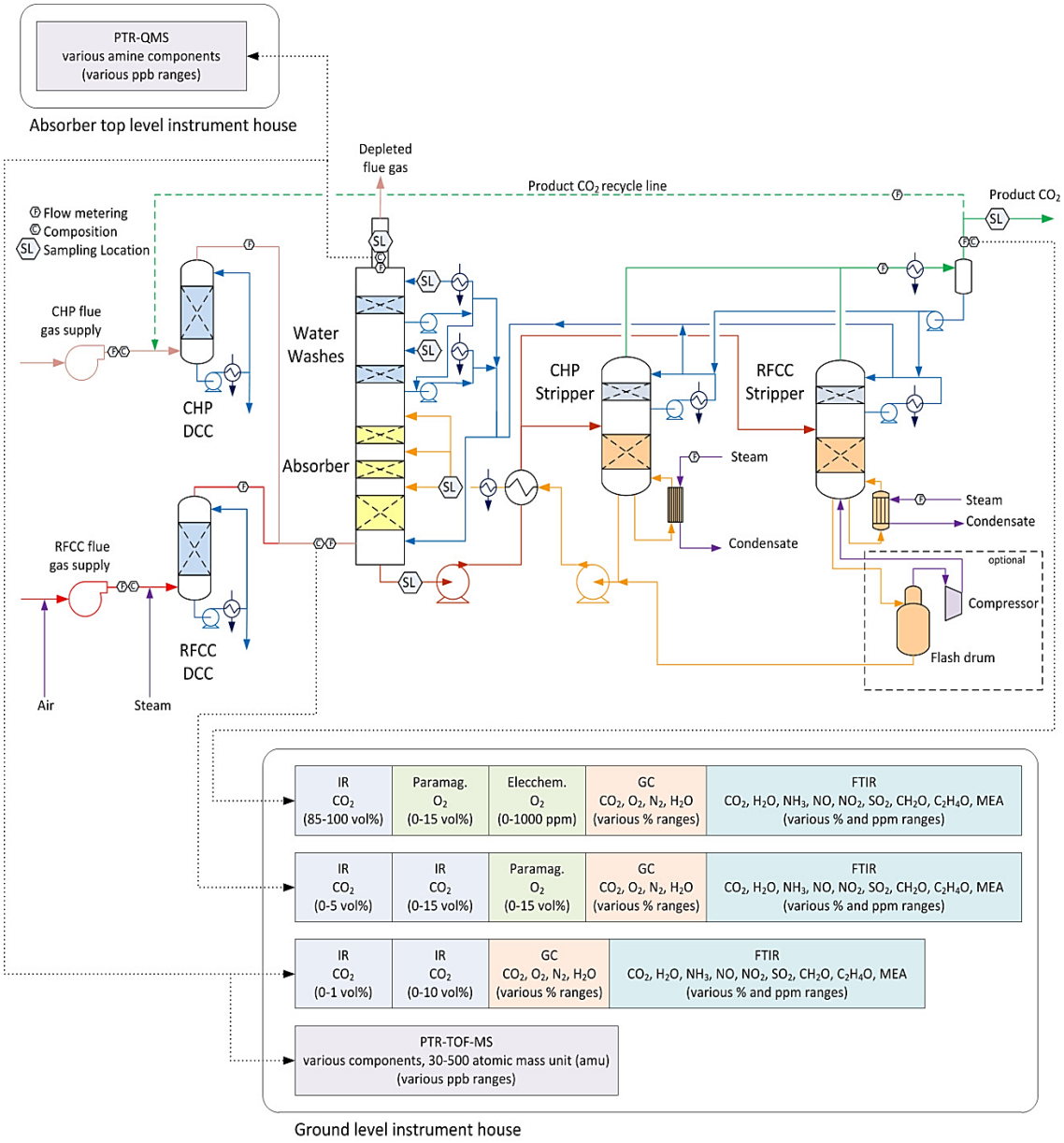


Figure 3.1: Carbon capture plant overview [26].

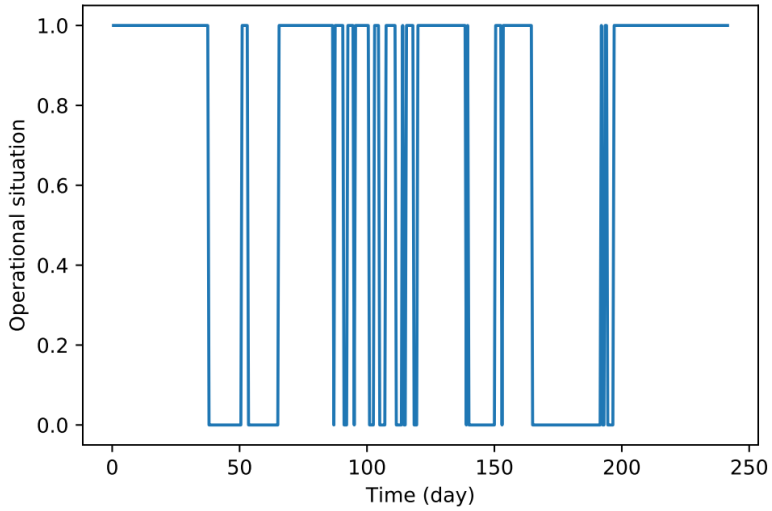


Figure 3.2: Operational time in carbon capture plant over the time

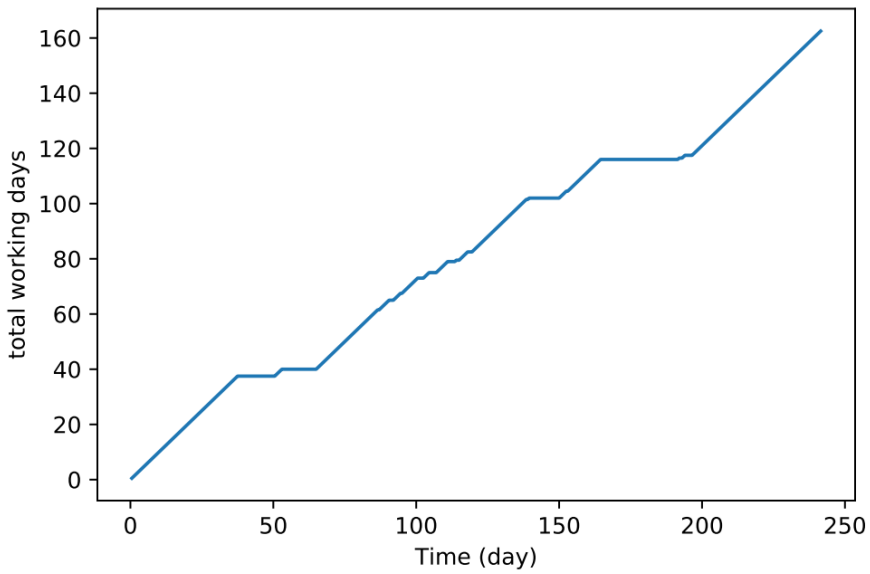


Figure 3.3: Total number of working days in the plant over the period

3.2 Machine learning process

Machine learning steps used in TCM data are shown in the Figure 3.4. As it can be seen in the figure, data should be firstly collected and then pre-processed. Pre-processing step can consist of several steps like cleaning, manipulation of data and feature selection. After pre-processing, model should be chosen and trained. Performance of the model is evaluated by test model step. Finally, improvement can be carried out by comparing test and train result in case test results are not satisfied.

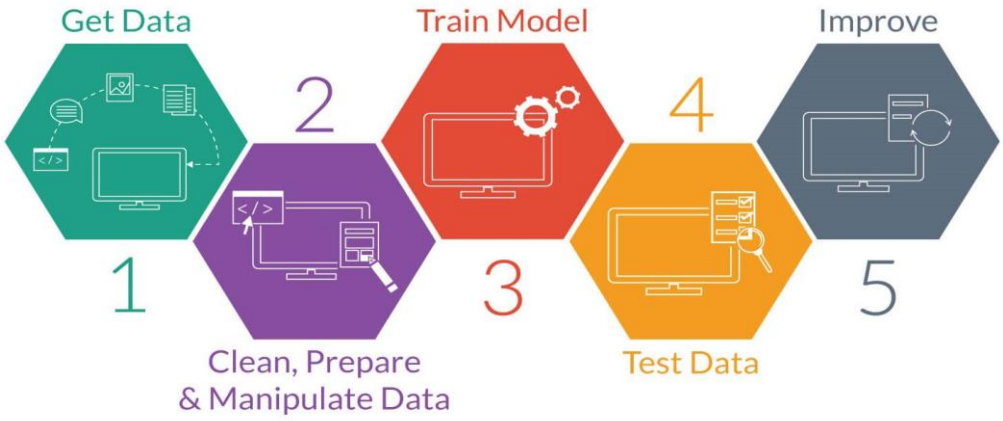


Figure 3.4: Machine learning steps for TCM data [44].

3.3 Data collection and pre-processing

There are available online and lab data from different parts of the plant. According to the literature and former research studies at TCM, there are some components that are very important in solvent degradation prediction. Ammonia, Formaldehydes, Acetaldehydes, 1H-Imidazole-1-ethanol (HEI) and 4-(2-hydroxyethyl)-2-piperazinone (HEPO) which can be seen in the Table 2.2 or Figure 2.15 in detail. Data availability is one of the concerns that can limit

the prediction of some components. For instance, available HEPO data over the period is just 20 datasets which is very low for further analysis. In the following, the process of the data selection and pre-processing will be explained in detail.

3.3.1 Online data

To measure parameters like temperature, pressure, flue gas decomposition and flow rate, several instruments have been installed at TCM to record the information continuously. This data referred to online data, has high resolution with frequency of 12 hours or even better, as they are continuously recorded. Therefore, a vast majority of data needed can be obtained from online sources. According to the literature and as Figure 3.1 shows, source of solvent loss can be followed in the exit of the plant which is depleted gas and product gas. Ammonia, Formaldehydes and Acetaldehydes can represent oxidative degradation in the absorber and stripper containing a high percentage of total solvent degradation. Besides, stripper temperature and data regarding the absorber inlet should be extracted. Stripper temperature plays a role in thermal degradation while fraction of components in the absorber inlet influences the oxidative degradation. Flue gas in the inlet of the absorber, depleted flue gas and product flue gas are also of interest because they represent flow rate of each component after multiplying into their component fraction. Finally, to measure volatility of the solvent in absorber, flow rate of water in the water wash sections have been extracted.

Data pre-processing has been executed in Excel. All Nan values have been omitted and zero value has been replaced for flue gas rate less than 2000 Sm^3/h in the absorber inlet as it was proposed by TCM. Appendix B shows an overview of the final and cleaned online data for further implementation.

3.3.2 Lab data

As it can be seen in the Figure 3.1, there are some points in the plant shown in sample location (SL). There are several places in the plant that samples should be taken to evaluate their components in the laboratory. TCM provided more than 10 data sources for lab data over the period. Each lab data source corresponds to a specific period of time that in total would be around six months which is far less than available period time for online data. In addition, data frequency in each data source is different that needs further process. Ammonium and MEA content in both water wash sections, and viscosity are also essential data for further implementations based on the literature. Therefore, this data has been extracted form available lab data sources. Appendix D shows the source code used to extract the needed data from lab data sources. Excel was employed for further data pre-processing. Cleaned lab data is also presented in the Appendix C.

One of the issues in lab data was various data frequency which should be solved. One of the solutions is considering constant value in each interval. Figure 3.5 to Figure 3.8 show almost the same number of data sets (about 105 datasets) for each parameter while Figure 3.9 has a lower data frequency (76 datasets). However, this data has been extended by assuming constant behavior in each interval. This can be easily seen in Figure 3.9 where the figure has a shape like the step function. This assumption is shown in the Figure 3.10 for more explanation.

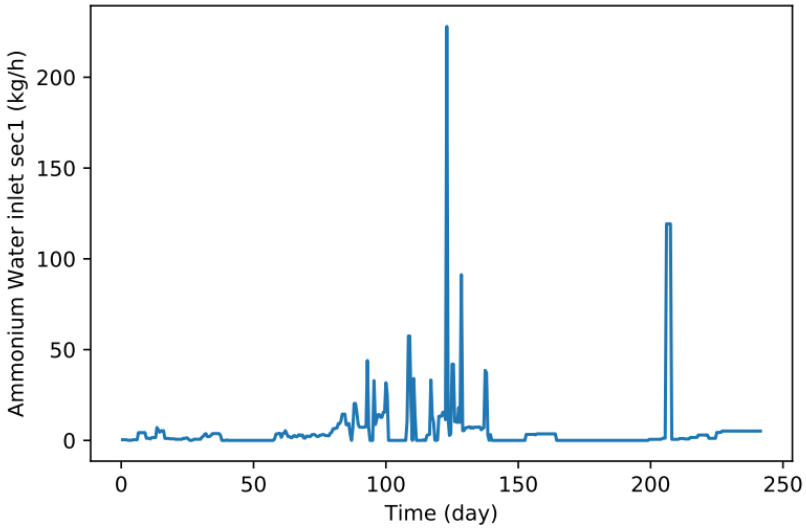


Figure 3.5: Ammonium content in the water wash inlet section 1 over the time

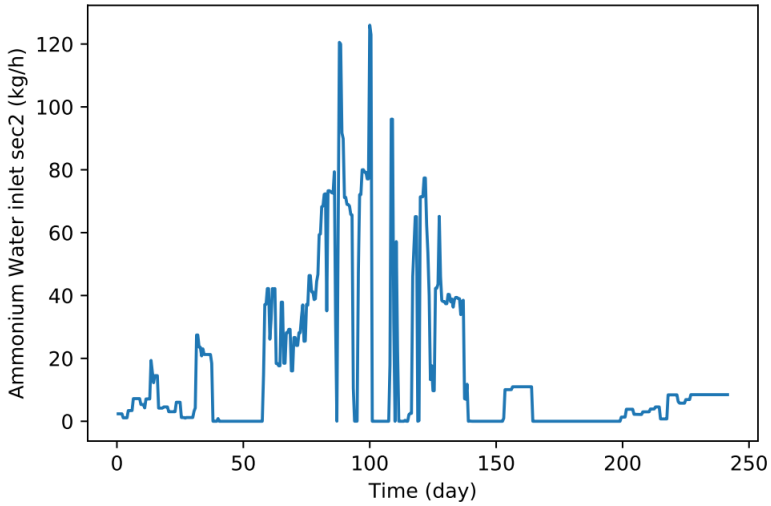


Figure 3.6: Ammonium content in the water wash inlet section 2 over the time

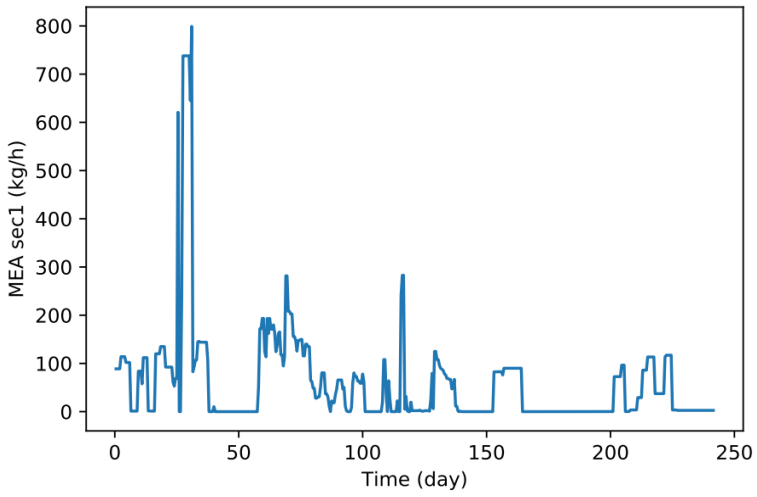


Figure 3.7: MEA content in the water wash inlet section 1 over the time

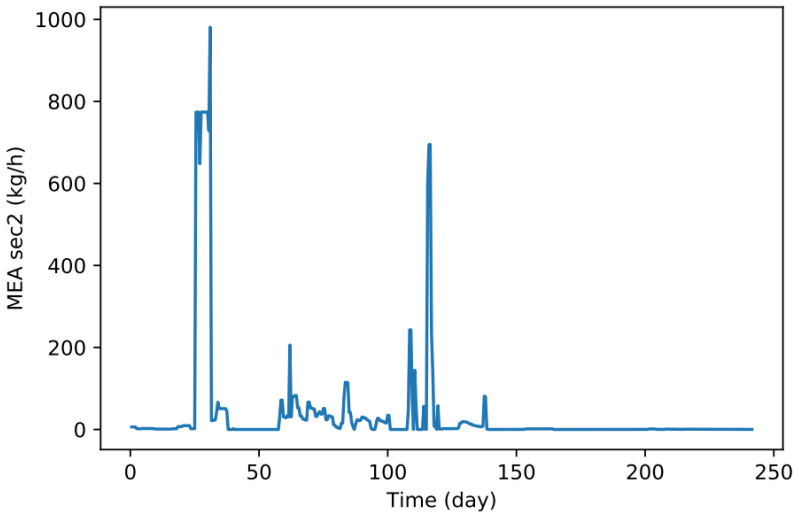


Figure 3.8: MEA content in the water wash inlet section 2 over the time

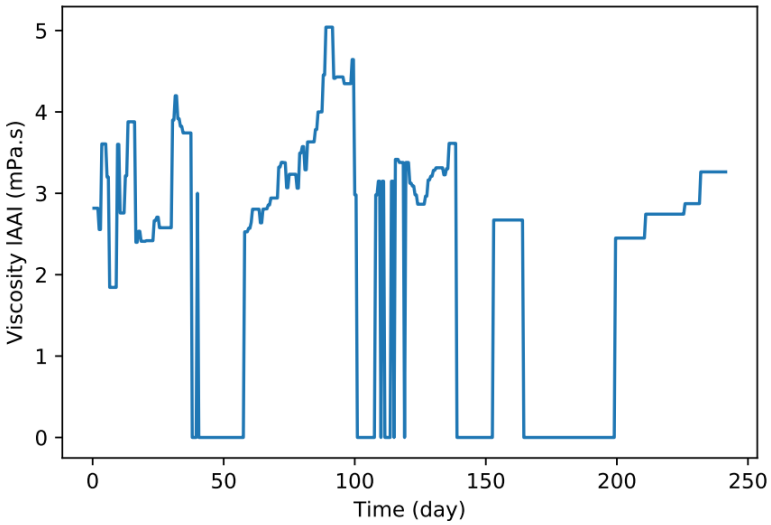


Figure 3.9: Viscosity of the lean solvent over the time

As shown in the right plot of Figure 3.10 , some intervals have constant value extended before combination with online data. Real data is also presented in the left plot of the figure.

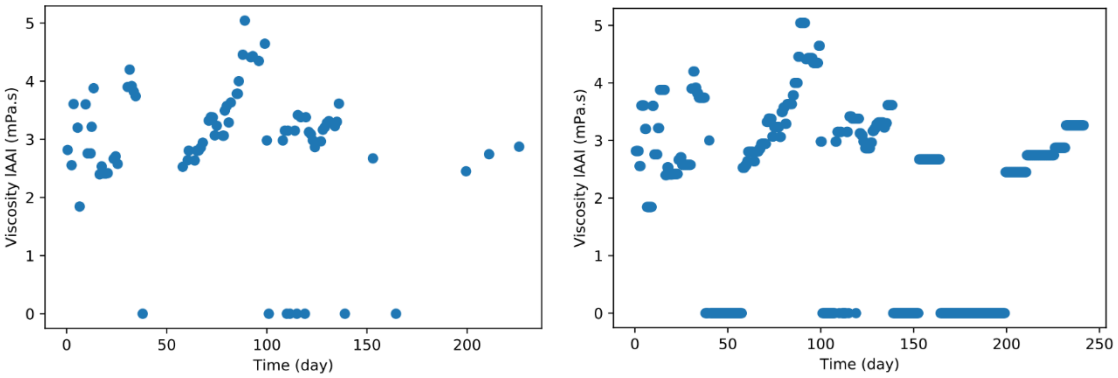


Figure 3.10: Viscosity of the lean solvent (right plot: extended data, left plot: original data)

By combining lab and online data, the whole data is used for the further pre-processing. After combination of online and lab, 483 datasets are ready further pre-processing and modelling. In the following chapter, feature selection process and results are presented.

4 Results and discussion

In this chapter, Pearson's and Spearman's coefficients for variables are firstly presented. Then, the results of all machine learning methods are introduced and discussed.

4.1 Feature selection

By combining lab and online data with the same frequency of data, Spearman's and Pearson's heatmap can be shown as Figure 4.1 and Figure 4.2, respectively. Besides, Table 4.1 and

Table 4.2 show the coefficients clearly for the further process. As the nature of solvent degradation is nonlinear, therefore, Spearman's coefficient would be more beneficial.

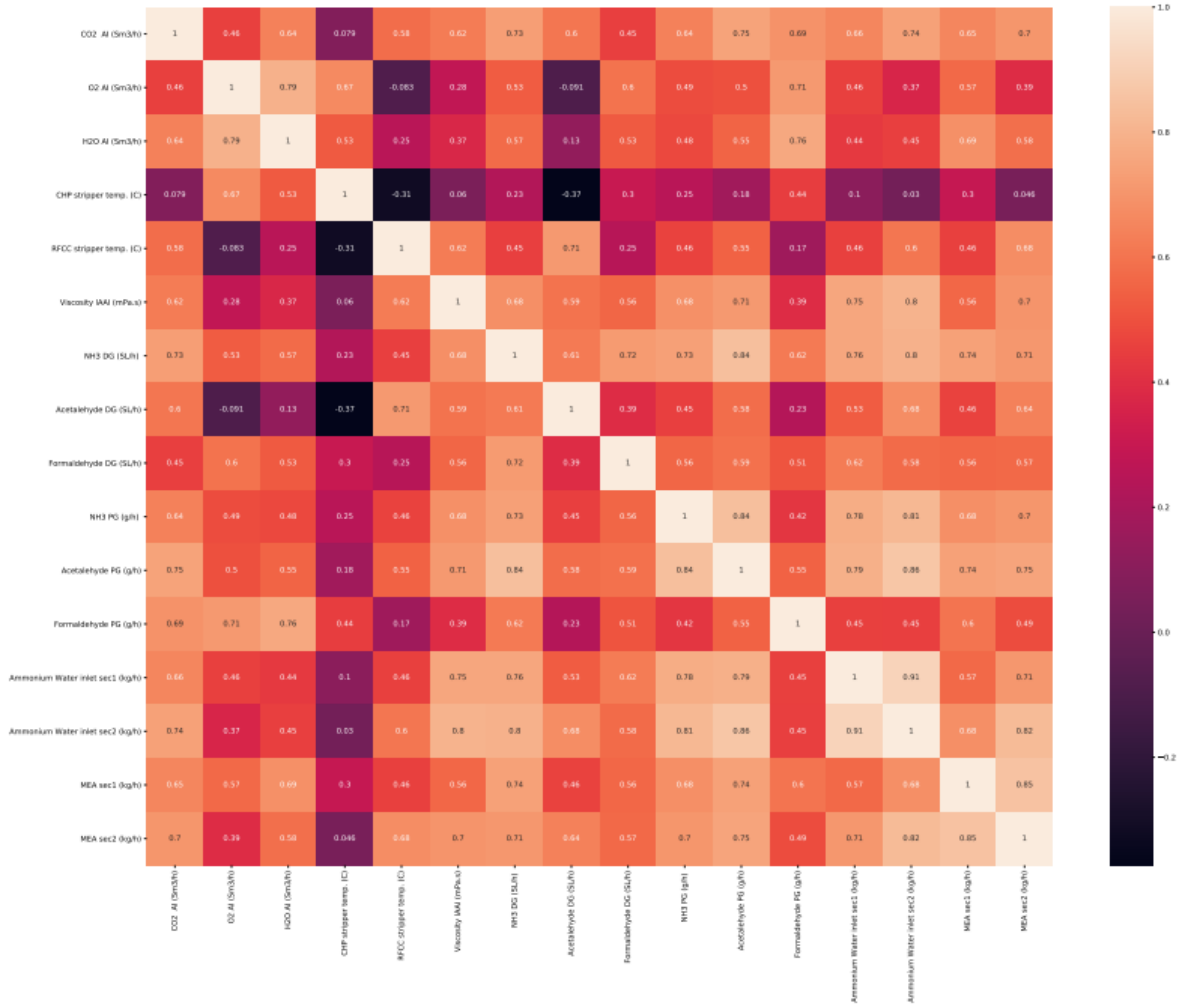


Figure 4.1: Spearman's heatmap for variables

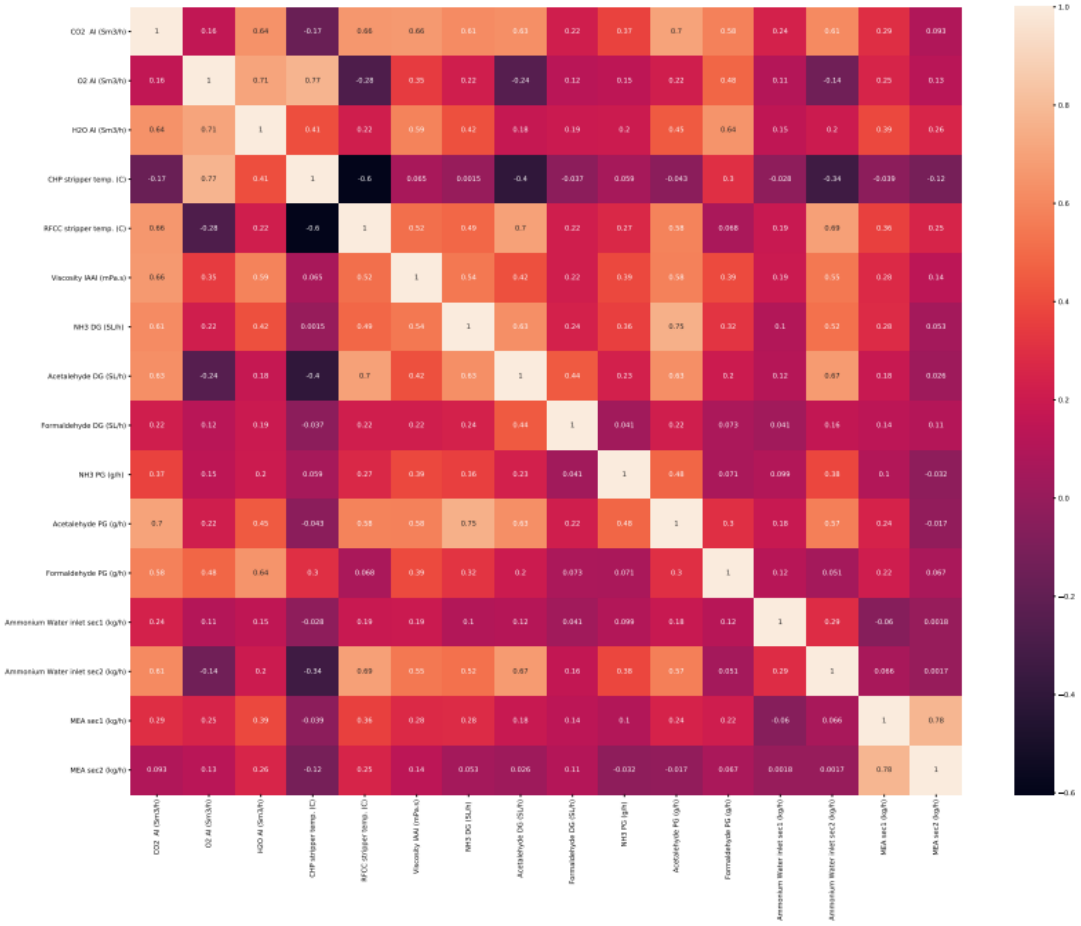


Figure 4.2: Pearson's heatmap for variables

Table 4.1: Spearman's coefficient results for variables

Variables	CO2 AI (Sm3/h)	O2 AI (Sm3/h)	H2O AI (Sm3/h)	CHP stripper temp. (C)	RFCC stripper temp. (C)	Viscosity lAAI (mPa.s)	NH3 DG (SL/h)	Acetaldehyde DG (SL/h)	Formaldehyde DG (SL/h)	NH3 PG (g/h)	Acetaldehyde PG (g/h)	Formaldehyde PG (g/h)	Ammonium Water inlet sec1 (kg/h)	Ammonium Water inlet sec2 (kg/h)	MEA sec1 (kg/h)	MEA sec2 (kg/h)
CO2 AI (Sm3/h)	1.00	0.46	0.64	0.08	0.58	0.62	0.73	0.60	0.45	0.64	0.75	0.69	0.66	0.74	0.65	0.70
O2 AI (Sm3/h)	0.46	1.00	0.79	0.67	-0.08	0.28	0.53	-0.09	0.60	0.49	0.50	0.71	0.46	0.37	0.57	0.39
H2O AI (Sm3/h)	0.64	0.79	1.00	0.53	0.25	0.37	0.57	0.13	0.53	0.48	0.55	0.76	0.44	0.45	0.69	0.58
CHP stripper temp. (C)	0.08	0.67	0.53	1.00	-0.31	0.06	0.23	-0.37	0.30	0.25	0.18	0.44	0.10	0.03	0.30	0.05
RFCC stripper temp. (C)	0.58	-0.08	0.25	-0.31	1.00	0.62	0.45	0.71	0.25	0.46	0.55	0.17	0.46	0.60	0.46	0.68
Viscosity lAAI (mPa.s)	0.62	0.28	0.37	0.06	0.62	1.00	0.68	0.59	0.56	0.68	0.71	0.39	0.75	0.80	0.56	0.70
NH3 DG (SL/h)	0.73	0.53	0.57	0.23	0.45	0.68	1.00	0.61	0.72	0.73	0.84	0.62	0.76	0.80	0.74	0.71
Acetalehyde DG (SL/h)	0.60	-0.09	0.13	-0.37	0.71	0.59	0.61	1.00	0.39	0.45	0.58	0.23	0.53	0.68	0.46	0.64
Formaldehyde DG (SL/h)	0.45	0.60	0.53	0.30	0.25	0.56	0.72	0.39	1.00	0.56	0.59	0.51	0.62	0.58	0.56	0.57
NH3 PG (g/h)	0.64	0.49	0.48	0.25	0.46	0.68	0.73	0.45	0.56	1.00	0.84	0.42	0.78	0.81	0.68	0.70
Acetalehyde PG (g/h)	0.75	0.50	0.55	0.18	0.55	0.71	0.84	0.58	0.59	0.84	1.00	0.55	0.79	0.86	0.74	0.75
Formaldehyde PG (g/h)	0.69	0.71	0.76	0.44	0.17	0.39	0.62	0.23	0.51	0.42	0.55	1.00	0.45	0.45	0.60	0.49
Ammonium Water inlet sec1 (kg/h)	0.66	0.46	0.44	0.10	0.46	0.75	0.76	0.53	0.62	0.78	0.79	0.45	1.00	0.91	0.57	0.71
Ammonium Water inlet sec2 (kg/h)	0.74	0.37	0.45	0.03	0.60	0.80	0.80	0.68	0.58	0.81	0.86	0.45	0.91	1.00	0.68	0.82
MEA sec1 (kg/h)	0.65	0.57	0.69	0.30	0.46	0.56	0.74	0.46	0.56	0.68	0.74	0.60	0.57	0.68	1.00	0.85
MEA sec2 (kg/h)	0.70	0.39	0.58	0.05	0.68	0.70	0.71	0.64	0.57	0.70	0.75	0.49	0.71	0.82	0.85	1.00

Table 4.2: Pearson's coefficient results for variables

Variables	CO2 AI (Sm3/h)	O2 AI (Sm3/h)	H2O AI (Sm3/h)	CHP stripper temp. (C)	RFCC stripper temp. (C)	Viscosity lAAI (mPa.s)	NH3 DG (SL/h)	Acetaldehyde DG (SL/h)	Formaldehyde DG (SL/h)	NH3 PG (g/h)	Acetaldehyde PG (g/h)	Formaldehyde PG (g/h)	Ammonium Water inlet sec1 (kg/h)	Ammonium Water inlet sec2 (kg/h)	MEA sec1 (kg/h)	MEA sec2 (kg/h)
CO2 AI (Sm3/h)	1.00	0.16	0.64	-0.17	0.66	0.66	0.61	0.63	0.22	0.37	0.70	0.58	0.24	0.61	0.29	0.09
O2 AI (Sm3/h)	0.16	1.00	0.71	0.77	-0.28	0.35	0.22	-0.24	0.12	0.15	0.22	0.48	0.11	-0.14	0.25	0.13
H2O AI (Sm3/h)	0.64	0.71	1.00	0.41	0.22	0.59	0.42	0.18	0.19	0.20	0.45	0.64	0.15	0.20	0.39	0.26
CHP stripper temp. (C)	-0.17	0.77	0.41	1.00	-0.60	0.07	0.00	-0.40	-0.04	0.06	-0.04	0.30	-0.03	-0.34	-0.04	-0.12
RFCC stripper temp. (C)	0.66	-0.28	0.22	-0.60	1.00	0.52	0.49	0.70	0.22	0.27	0.58	0.07	0.19	0.69	0.36	0.25
Viscosity lAAI (mPa.s)	0.66	0.35	0.59	0.07	0.52	1.00	0.54	0.42	0.22	0.39	0.58	0.39	0.19	0.55	0.28	0.14
NH3 DG (SL/h)	0.61	0.22	0.42	0.00	0.49	0.54	1.00	0.63	0.24	0.36	0.75	0.32	0.10	0.52	0.28	0.05
Acetalehyde DG (SL/h)	0.63	-0.24	0.18	-0.40	0.70	0.42	0.63	1.00	0.44	0.23	0.63	0.20	0.12	0.67	0.18	0.03
Formaldehyde DG (SL/h)	0.22	0.12	0.19	-0.04	0.22	0.22	0.24	0.44	1.00	0.04	0.22	0.07	0.04	0.16	0.14	0.11
NH3 PG (g/h)	0.37	0.15	0.20	0.06	0.27	0.39	0.36	0.23	0.04	1.00	0.48	0.07	0.10	0.38	0.10	-0.03
Acetalehyde PG (g/h)	0.70	0.22	0.45	-0.04	0.58	0.58	0.75	0.63	0.22	0.48	1.00	0.30	0.18	0.57	0.24	-0.02
Formaldehyde PG (g/h)	0.58	0.48	0.64	0.30	0.07	0.39	0.32	0.20	0.07	0.07	0.30	1.00	0.12	0.05	0.22	0.07
Ammonium Water inlet sec1 (kg/h)	0.24	0.11	0.15	-0.03	0.19	0.19	0.10	0.12	0.04	0.10	0.18	0.12	1.00	0.29	-0.06	0.00
Ammonium Water inlet sec2 (kg/h)	0.61	-0.14	0.20	-0.34	0.69	0.55	0.52	0.67	0.16	0.38	0.57	0.05	0.29	1.00	0.07	0.00
MEA sec1 (kg/h)	0.29	0.25	0.39	-0.04	0.36	0.28	0.28	0.18	0.14	0.10	0.24	0.22	-0.06	0.07	1.00	0.78
MEA sec2 (kg/h)	0.09	0.13	0.26	-0.12	0.25	0.14	0.05	0.03	0.11	-0.03	-0.02	0.07	0.00	0.00	0.78	1.00

AI, lAAI, DG, PG, sec and temp. in the tables represent Absorber Inlet, lean Amine Absorber Inlet, Depleted Gas, Product Gas, section and temperature, respectively.

The corresponding source code to create the coefficients and heatmap can be found in Appendix E.

To select the suitable features, 0.5 is used as threshold to choose the effective parameter for predicting all sources of the solvent degradation namely NH3 DG and PG, Acetaldehyde DG and PG, Formaldehyde DG and PG, Ammonium Water inlet sections 1 and 2, MEA in the sections 1 and 2. According to Table 4.1, there is no relation between all targets and CHP stripper temperature since its absolute coefficients are less than 0.5. Therefore, this variable can be neglected for further modelling. In addition, all variables except for RFCC stripper temperature have coefficient more than 0.5 for NH3 DG prediction. Regarding the Acetaldehyde DG, variables CO_2 AI, RFCC stripper temp., NH3 DG, Acetaldehyde PG, Ammonium Water inlet sec1 and Ammonium Water inlet sec2, and MEA sec2 have coefficients less than threshold 0.5. Therefore, the rest of variables would be considered to forecast Acetaldehyde DG. This process should be carried out on all target variables to reach the minimum features for modelling. Table 4.3 shows the result for feature selection so that coefficient in the yellow cells are the final features for each target.

Table 4.3: Effective variables based on Spearman's correlation method

Variables	CO2 AI (Sm3/h)	O2 AI (Sm3/h)	H2O AI (Sm3/h)	RFCC stripper temp. (C)	Viscosity lAAI (mPa.s)	NH3 DG (SL/h)	Acetaldehyde DG (SL/h)	Formaldehyde DG (SL/h)	NH3 PG (g/h)	Acetaldehyde PG (g/h)	Formaldehyde PG (g/h)	Ammonium Water inlet sec1 (kg/h)	Ammonium Water inlet sec2 (kg/h)	MEA sec1 (kg/h)	MEA sec2 (kg/h)
NH3 DG (SL/h)	0.73	0.53	0.57	0.45	0.68	1.00	0.61	0.72	0.73	0.84	0.62	0.76	0.80	0.74	0.71
Acetalehyde DG (SL/h)	0.60	-0.09	0.13	0.71	0.59	0.61	1.00	0.39	0.45	0.58	0.23	0.53	0.68	0.46	0.64
Formaldehyde DG (SL/h)	0.45	0.60	0.53	0.25	0.56	0.72	0.39	1.00	0.56	0.59	0.51	0.62	0.58	0.56	0.57
NH3 PG (g/h)	0.64	0.49	0.48	0.46	0.68	0.73	0.45	0.56	1.00	0.84	0.42	0.78	0.81	0.68	0.70
Acetalehyde PG (g/h)	0.75	0.50	0.55	0.55	0.71	0.84	0.58	0.59	0.84	1.00	0.55	0.79	0.86	0.74	0.75
Formaldehyde PG (g/h)	0.69	0.71	0.76	0.17	0.39	0.62	0.23	0.51	0.42	0.55	1.00	0.45	0.45	0.60	0.49
Ammonium Water inlet sec1 (kg/h)	0.66	0.46	0.44	0.46	0.75	0.76	0.53	0.62	0.78	0.79	0.45	1.00	0.91	0.57	0.71
Ammonium Water inlet sec2 (kg/h)	0.74	0.37	0.45	0.60	0.80	0.80	0.68	0.58	0.81	0.86	0.45	0.91	1.00	0.68	0.82
MEA sec1 (kg/h)	0.65	0.57	0.69	0.46	0.56	0.74	0.46	0.56	0.68	0.74	0.60	0.57	0.68	1.00	0.85
MEA sec2 (kg/h)	0.70	0.39	0.58	0.68	0.70	0.71	0.64	0.57	0.70	0.75	0.49	0.71	0.82	0.85	1.00

4.2 Support Vector Regression (SVR)

There are several hyperparameters in SVR method that should be tuned before further processing. Regularization parameter (C), epsilon (ε), kernel function, tolerance and gamma (kernel coefficient) can be named as the most important hyperparameters in SVR method. Regularization parameter should be a positive number and the regularization technique is based on L2 regularization. Epsilon also represents the tube area with ε radius that loss function applies no penalty for the training data. There are several kernel functions such as linear, poly, radial based function (rbf) and sigmoid. Kernel function rbf is the most common function used in SVR and therefore, it is used for current SVR models. Besides, tolerance and gamma are set to 10^{-5} and 'scale' to reach better results. Therefore, regularization parameter and epsilon should be tuned in each type of solvent degradation in the carbon capture plant.

Grid search optimization method is employed to tune regularization parameter and epsilon in SVR models. 20 and 5 points are chosen for regularization parameter and epsilon, respectively, to search for the best results. As shown in the Figure 4.3, grid search has been executed in the range of 0 to 37000 and 0.02 to 0.1 for regularization parameter and epsilon, respectively, for MEA sec1. As shown in the figure, tuned regularization parameter and epsilon are chosen 30000 and 0.02, respectively, to reach 0.9670 and 0.9861 for the train and test R^2 score. Table 4.4 shows the results for each type of solvent degradation based on the optimal regularization parameter and epsilon for two splitting type of 80/20, 70/30 for train/test datasets. The results show that there is an excellent validation in this method for most cases while R^2 is low for training and test dataset in some cases. In fact, SVR predicts well in NH3 DG, Acetaldehyde PG, Formaldehyde PG, Ammonium Water inlet sec2, MEA sec1 and MEA sec2 as the train and test R^2 is more than 0.9. In other cases, the models need more improvement to reach better result. As a general rule and based on the results of SVR methods shown in the table, the model can appropriately predict when data is split 80/20 in train/ test datasets rather than 70/30.

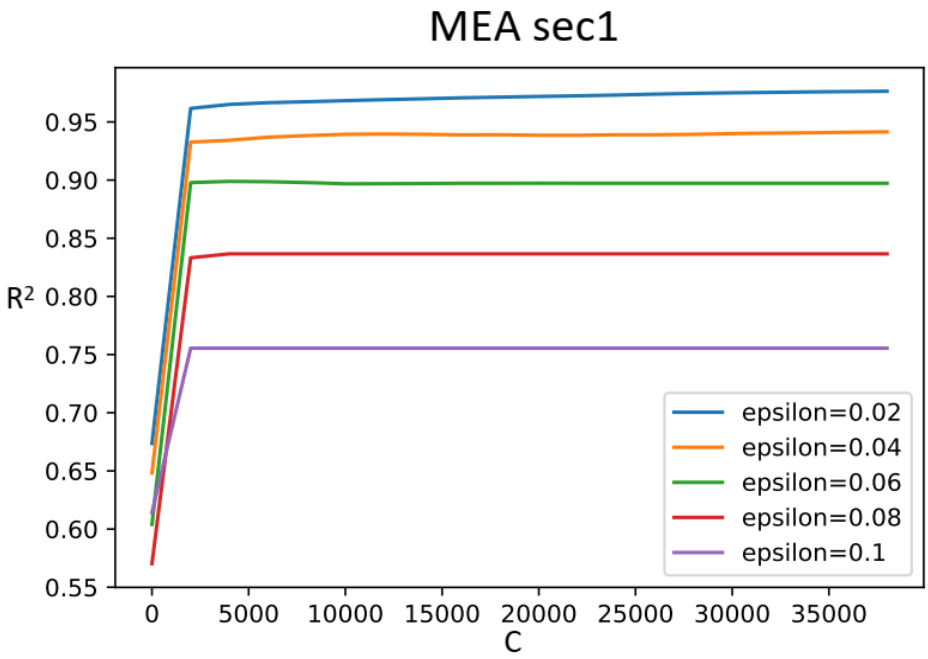


Figure 4.3: Results of tuning regularization parameter and epsilon in SVR model for MEA sec1.

Table 4.4: Results for SVR method with tuned hyperparameter.

Splitting type	**Train set (80%), Test set (20%)**		**Train set (70%), Test set (30%)**		**Regularization parameter**	**Epsilon**
Outputs	**Train R^2**	**Test R^2**	**Train R^2**	**Test R^2**		
NH3 DG	0.919	0.902	0.919	0.906	47000	0.02
Acetaldehyde DG	0.554	0.502	0.566	0.485	45000	0.04
Formaldehyde DG	0.449	0.443	0.482	0.362	55000	0.05
NH3 PG	0.453	0.446	0.458	0.436	52000	0.04
Acetaldehyde PG	0.970	0.973	0.968	0.976	56000	0.02
Formaldehyde PG	0.873	0.856	0.876	0.852	50000	0.02
Ammonium Water inlet sec1	0.305	0.149	0.292	0.175	12000	0.02
Ammonium Water inlet sec2	0.947	0.887	0.947	0.915	60000	0.04
MEA sec1	0.9670	0.9861	0.963	0.983	30000	0.02
MEA sec2	0.980	0.984	0.975	0.988	15000	0.02

Appendix F shows the source code for SVR methods.

4.3 Random Forest (RF)

There are also several hyperparameters in RF method that should be also tuned before further implementation. These hyperparameters are listed as n_estimators, min_sample_split, min_sample_leaf, max_depth and bootstrap. To tune hyperparameters in RF models, randomized search cross validation is used. Therefore, hyperparameters can be selected in the range described in the Table 4.5. As shown in the table, several points are produced linearly for max_depth and n_estimator by using linear function in python.

Table 4.5: Results for tuned hyperparameter in RF method for splitting 80/20 train/test dataset.

Hyperparameter	**Hyperparameter choices**
n_estimator	10 numbers between 20 and 1000 (linear function)
min_sample_split	2, 5, 8
min_sample_leaf	1, 2, 5
max_depth	11 number between 1 and 110 (linear function), None
bootstrap	False, True

By implementing randomized search cross validation for splitting 80/20 in train/ test data sets, tuned hyperparameter in RF model are described as Table 4.6. None in the max_depth demonstrates that the extension of nodes continues until number of samples in all leaves are less than min_samples_split.

Table 4.6: Results for tuned hyperparameter in RF method for splitting 80/20 in train/test dataset

Type of degradation	**Hyperparameters**				
	n_estimators	min_sample_split	min_sample_leaf	max_depth	bootstrap
NH3 DG	237	5	1	60	False
Acetaldehyde DG	128	2	1	none	False
Formaldehyde DG	20	2	1	50	True
NH3 PG	673	5	2	none	False
Acetaldehyde PG	346	5	1	90	False
Formaldehyde PG	128	2	1	none	False
Ammonium Water inlet sec1	128	2	1	none	False
Ammonium Water inlet sec2	455	8	1	80	False
MEA sec1	237	2	2	110	False
MEA sec2	128	2	1	none	False

Figure 4.4 illustrates a view of RF model belonging to Acetaldehyde DG.

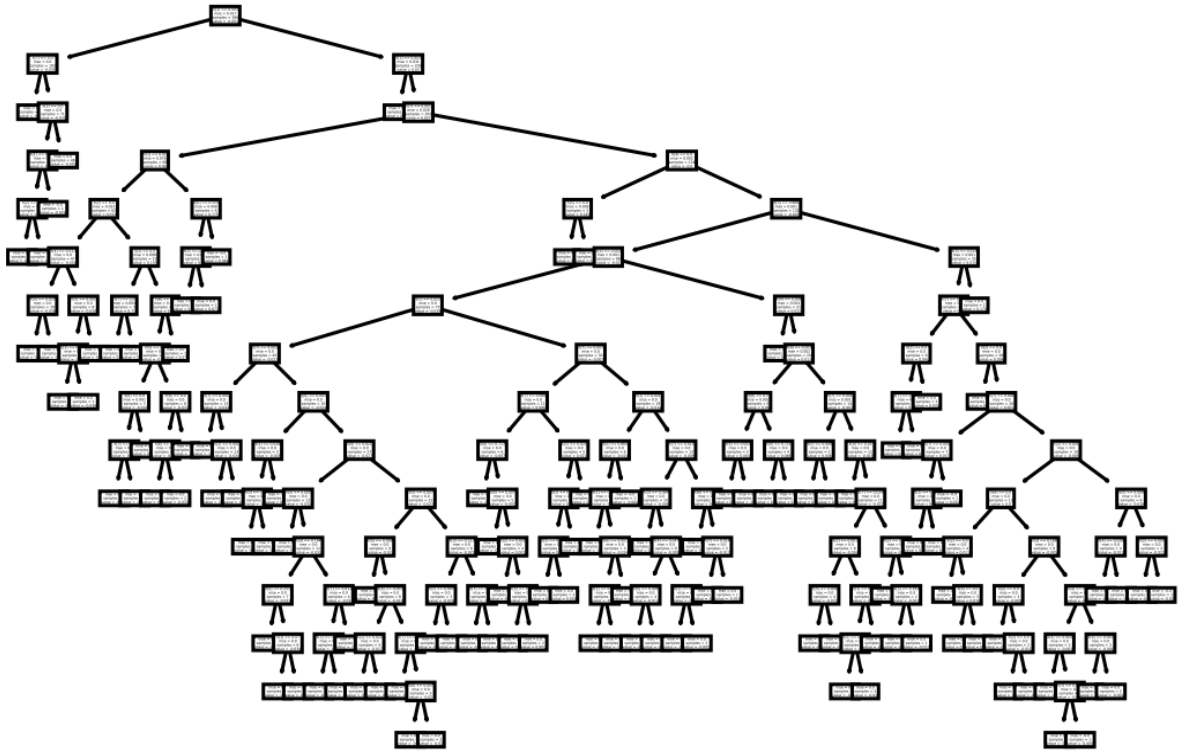


Figure 4.4: RF architecture for Acetaldehyde DG with tuned hyperparameter.

Table 4.7: Results for RF method by using cross validation random search.

Splitting type	**Train set (80%), Test set (20%)**		**Train set (70%), Test set (30%)**	
Type of degradation	**Train R^2**	**Test R^2**	**Train R^2**	**Test R^2**
NH3 DG	0.9951	0.9099	0.9999	0.8917
Acetaldehyde DG	0.9878	0.8334	0.9999	0.7684
Formaldehyde DG	0.9218	0.4841	0.5848	0.2866
NH3 PG	0.9250	0.5059	0.9414	0.5434
Acetaldehyde PG	0.9999	0.9368	0.9982	0.9146
Formaldehyde PG	0.9999	0.7695	0.9919	0.7257
Ammonium Water inlet sec1	0.9489	0.1384	0.9411	0.1594
Ammonium Water inlet sec2	0.9999	0.8467	0.9999	0.8143
MEA sec1	0.9911	0.8830	0.9999	0.8944
MEA sec2	0.9756	0.9841	0.9602	0.9585

Table 4.7 shows the results of RF model for different types of solvent degradation in the plant. As it can be seen in the table, R^2 in the training data is very good (more than 0.95). However, there is a gap between R^2 training and testing dataset in a few models. For instance, RF model in Formaldehyde DG has experienced overfitting since there is a considerable gap between R^2 in train and test datasets. To avoid overfitting in this model, max_depth and min_sample_split was limited but no significant changes occurred. For example, by repeating the tuning process with limiting max_depth and min_sample_split to 20 and 4, respectively, train and test R^2 became 0.8927 and 0.5790. This process was also carried out for NH3 PG and the results were not satisfying. In Ammonium water inlet sec1, there is a huge gap between train and test R^2 which can be due to data quality.

Appendix G displays the source code for Random Forest with randomized search cross validation.

4.4 Artificial Neural Network (ANN)

Hyperparameters in ANN model that should be tuned, are learning rate, number of neurons in each hidden layer, number of hidden layers, epochs, batch size and activation function. Several experiments were carried out before tuning all hyperparameters and seen that the best results are regarding to the models having activation functions of ReLU and tanh.

Besides, one hidden layer was not observed as efficient as two hidden layers in ANN models, though increasing more hidden layers could increase the probability of overfitting. Therefore, two hidden layers are used in the further implementations. ReLU and tanh are considered as the first and second hidden layer activation function, respectively, while activation function for the output layer was chosen ReLU. ReLU was selected as the activation function for the output layer since targets or outputs are positive in models. Batch size was also assumed to be 40 after checking several choices. In addition, epochs should be checked before finalizing the ANN model for each type of solvent degradation since it can trap the model into overfitting. Finally, the rest of the hyperparameters should be tuned by one of the optimization methods. Grid search is used to optimize learning rate in each model for different number of neurons in each hidden layer. 120, 70 and 40 number of neurons for the first hidden layer and 100, 60 and 20 for the second hidden layers are considered. These numbers were chosen based on several trial implementations and checking train and test R^2 in the ANN model. In fact, a grid search including 20 points of different learning rate is examined and the best learning rate based on R^2 is chosen. Besides, all implementations are carried out with splitting data 80/20 in train/test. An overview of ANN architecture with two hidden layers is shown in the Figure 4.5. As shown in the figure, this model contains two hidden layers with 40 and 20 neurons in the first and second hidden layers, respectively. Input and output layer are also the first and last layer of this ANN model.

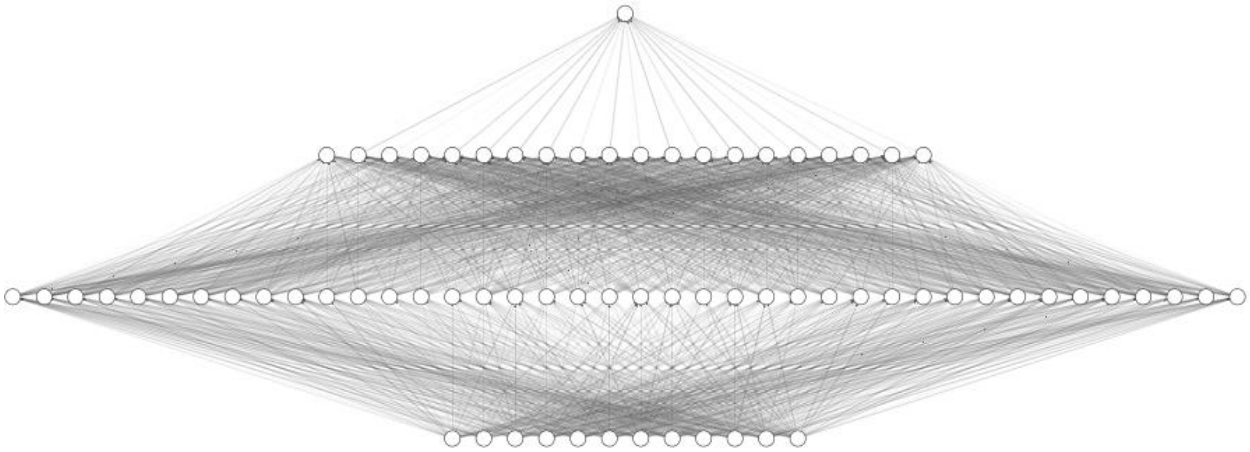


Figure 4.5: ANN architecture for one of the models.

4.4.1 Performance of ANN for different types of solvent degradation

Results for the NH3 DG are shown in Table 4.7. As it is seen, train and test R^2 scores have very excellent results for different number of neurons. Since an ANN model with 120 and 100 neurons in the first and second hidden layers have the best results among others, tuned hyperparameter relevant to this model is chosen for further improvement.

Table 4.8: Results for NH3 DG with ANN method

# Neuron in first hidden layer	# Neuron in second hidden layer	Train R^2	Test R^2	Learning rate
120	100	0.9981	0.9231	0.0015
	60	0.9833	0.9081	0.0019
	20	0.9916	0.9002	0.0035
70	100	0.9783	0.9161	0.0035
	60	0.9821	0.9196	0.0021
	20	0.9776	0.9224	0.0023
40	100	0.9800	0.9170	0.0021
	60	0.9707	0.9154	0.0021
	20	0.9819	0.9107	0.0031

loss function versus epochs is plotted to find suitable epochs number for NH3 DG. As shown in Figure 4.6, validation and train dataset approximately experience no change after 1600 epochs. Therefore, final ANN hyperparameters with corresponding R^2 results for NH3 DG are described in the Table 4.9.

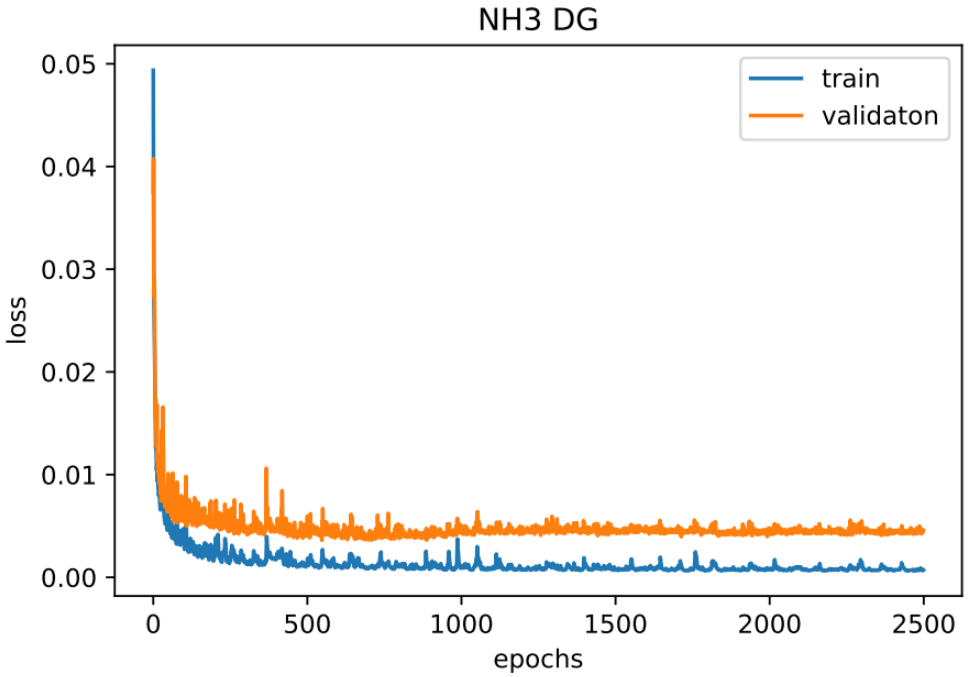


Figure 4.6: loss versus epochs for NH3 DG

Table 4.9: Final hyperparameter of ANN model for NH3 DG

Degradation type	**#Hidden layer**	**#Nodes in the first hidden layer**	**#Nodes in the second hidden layer**	**Batch size**	**Epochs**	**Learning rate**	**Train R^2**	**Test R^2**
NH3 DG	2	120	100	40	1600	0.0015	0.9902	0.9150

Same as NH3 DG, optimized learning rate for Acetaldehyde DG is obtained. Results shown in Table 4.10, indicate that there is a gap between R^2 in training and testing dataset which is probably due to overfitting. Figure 4.7 demonstrates that the ANN model experience overfitting after approximately 100 epochs as validation loss increases. Therefore, regularization methods are used to prevent overfitting in this model. Early stopping and L2 regularization method are applied to improve the results. As shown in Figure 4.8, regularization methods improved the model properly. After implementing all models shown in the Table 4.10, the best ANN hyperparameter is described in the Table 4.11. As it can be seen in Table 4.11, a model with 120 and 60 neurons in the first and second hidden layers is the best ANN model.

Table 4.10: Results for Acetaldehyde DG with ANN method

# Neuron in first hidden layer	# Neuron in second hidden layer	Train R^2	Test R^2	Learning rate
120	100	0.9242	0.6894	0.00023
	60	0.8920	0.7254	0.00016
	20	0.9035	0.6460	0.00024
70	100	0.8600	0.7007	0.00021
	60	0.9152	0.6444	0.00028
	20	0.8922	0.6613	0.00023
40	100	0.8609	0.6202	0.00022
	60	0.9195	0.6917	0.0008
	20	0.8435	0.7429	0.00029

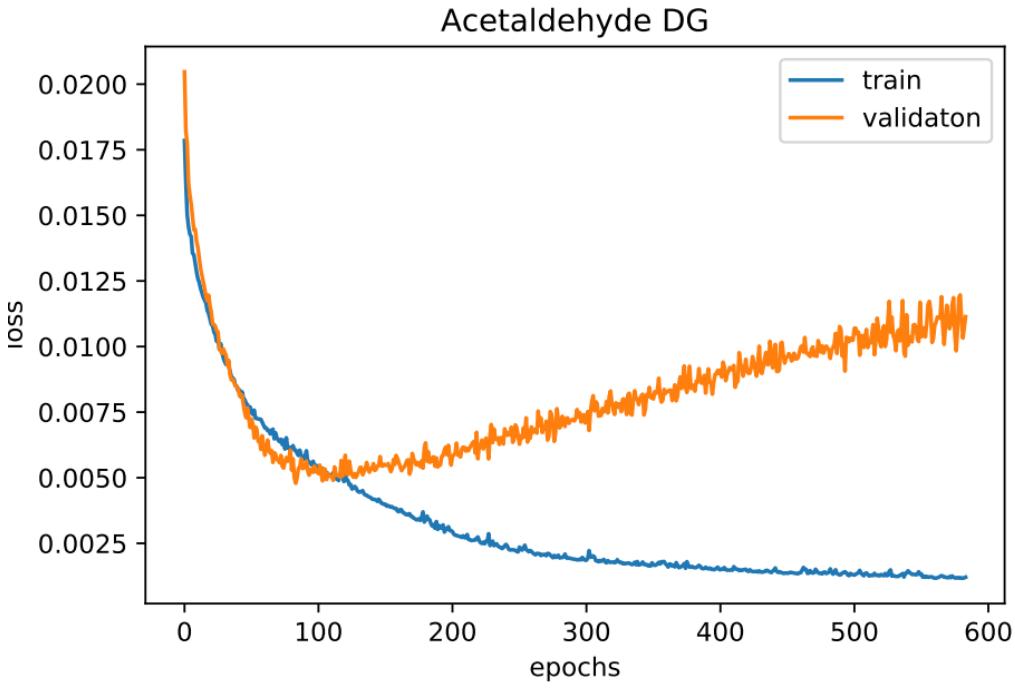


Figure 4.7: Overfitting in Acetaldehyde DG ANN model

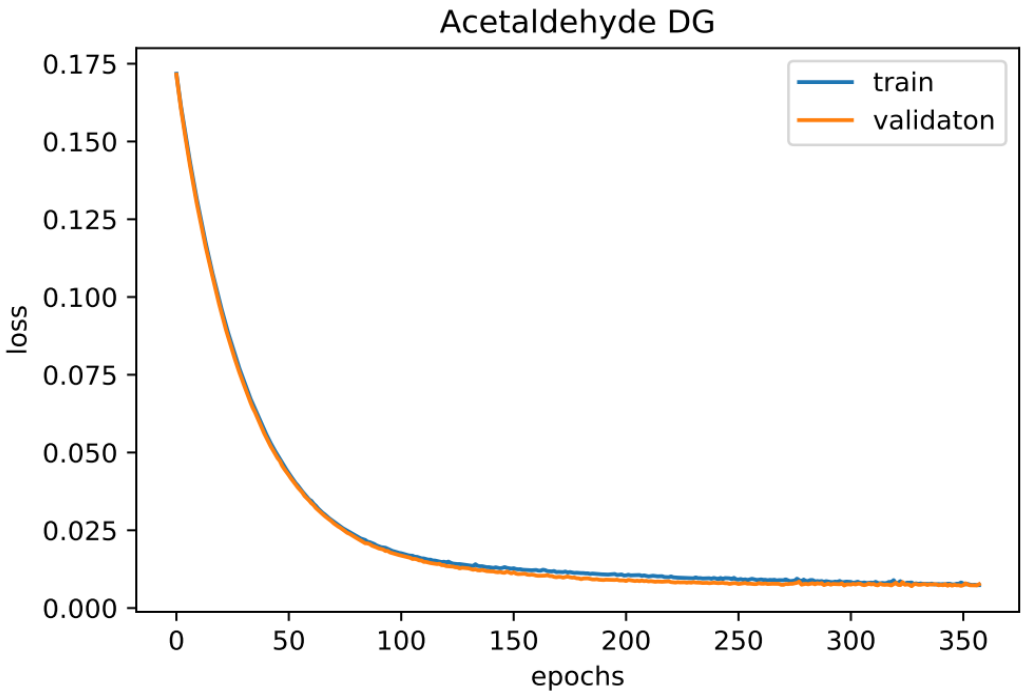


Figure 4.8: Loss versus epochs after using regularization methods

Table 4.11: Final hyperparameter of ANN model for Acetaldehyde DG

Degradation type	**#Hidden layer**	**#Nodes in first hidden layer**	**#Nodes in second hidden layer**	**Batch size**	**Epochs**	**Learning rate**	**Train R^2**	**Test R^2**
Acetaldehyde DG	2	120	60	40	332	0.00016	0.7809	0.7791

The process of tuning is also executed for Formaldehyde DG. Table 4.12 displays the corresponding results for Formaldehyde DG. As it can be seen, the gap between train and test R^2 can be due to overfitting. Therefore, overfitting should be investigated in Formaldehyde DG as well. As shown in Figure 4.9, overfitting has occurred, and regularization methods should be utilized. After implementing early stopping and L2 regularization method in all cases, an ANN model with 40 and 60 neurons in the first and second hidden layers, respectively, was the best model. Figure 4.10 shows the loss versus epochs after applying regularization methods on the model. As indicated in the figure, ANN model experience improvement after regularization. In addition, Table 4.13 shows the hyperparameter and R^2 results for the tuned ANN model for Formaldehyde DG.

Table 4.12: Results for Formaldehyde DG with ANN method

# Neuron in first hidden layer	# Neuron in second hidden layer	Train R^2	Test R^2	Learning rate
120	100	0.8902	0.5581	0.00011
	60	0.7806	0.6044	0.00007
	20	0.9152	0.4980	0.00025
70	100	0.9520	0.3956	0.00037
	60	0.9234	0.02198	0.00037
	20	0.9229	0.3212	0.00033
40	100	0.8609	0.349446	0.00019
	60	0.8853	0.5716	0.00019
	20	0.8919	0.4426	0.00031

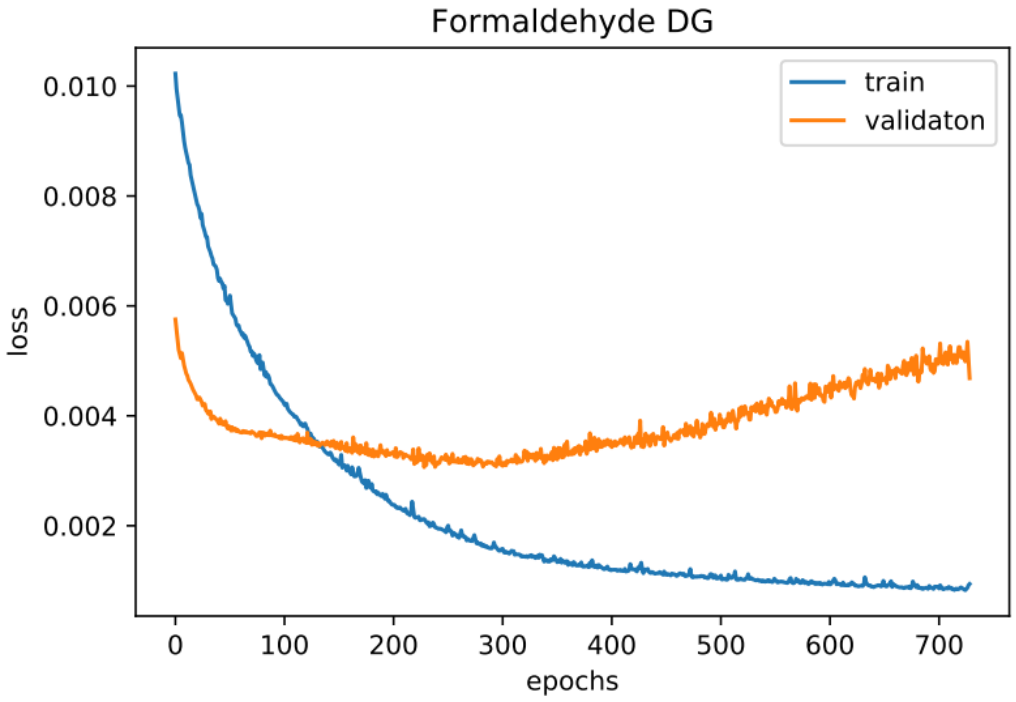


Figure 4.9: Overfitting in Formaldehyde DG ANN model

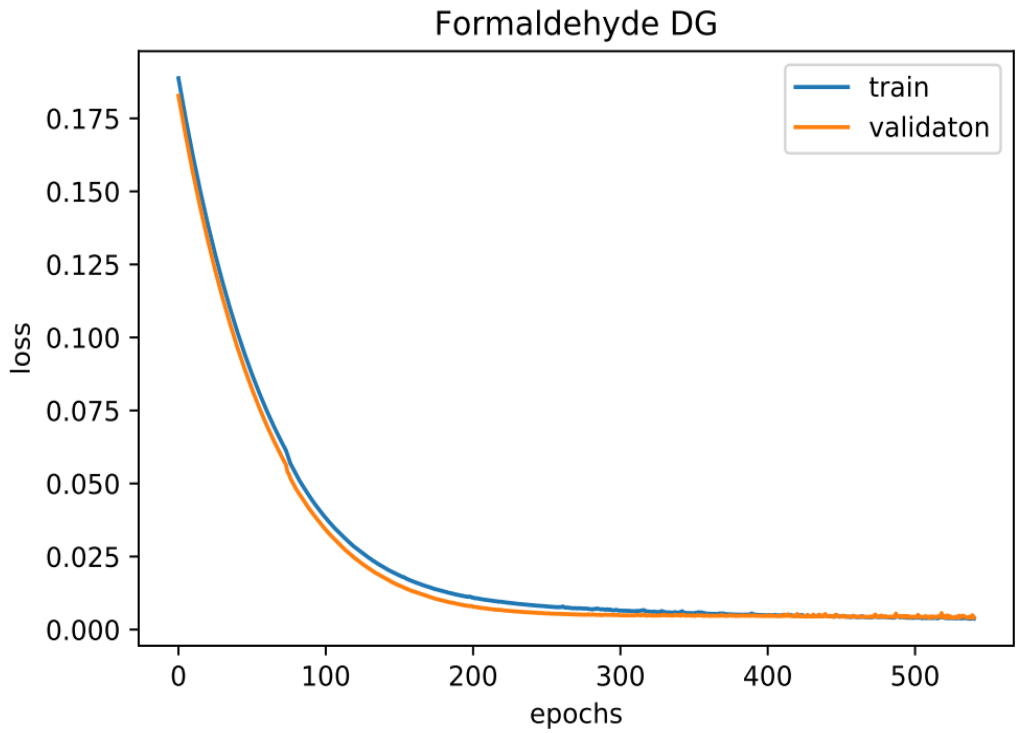


Figure 4.10: Loss versus epochs after using regularization methods

Table 4.13: Final hyperparameter of ANN model for Formaldehyde DG

Degradation type	**#Hidden layer**	**#Nodes in first hidden layer**	**#Nodes in second hidden layer**	**Batch size**	**Epochs**	**Learning rate**	**Train R^2**	**Test R^2**
Formaldehyde DG	2	40	60	40	511	0.00019	0.7656	0.7383

Tuned learning rate in the corresponding number of neurons is indicated in the Table 4.14 for NH3 PG. As it is clear from the table, there is a huge gap between R^2 in training and testing datasets. Besides, models with 20 neurons in the first layer have an unfavorable performance in test R^2 presenting negative value. In addition, models with 70 neurons in the first layer does not either perform well. Since there is a gap between train and test R^2, overfitting is firstly investigated. Figure 4.11 shows that the model experiences overfitting. After applying regularization methods (early stopping and L2), overfitting is avoided that can be seen in the Figure 4.12. Finally, after implementing all models, hyperparameters and R^2 results for the best ANN model for NH3 PG are described in Table 4.15.

Table 4.14: Results for NH3 PG with ANN method

# Neuron in first hidden layer	# Neuron in second hidden layer	Train R^2	Test R^2	Learning rate
120	100	0.8418	0.3295	0.00028
	60	0.8077	0.1878	0.00028
	20	0.8181	0.2354	0.00037
70	100	0.8028	0.2410	0.00009
	60	0.8044	0.2074	0.00033
	20	0.7383	0.1844	0.00033
40	100	0.7267	0.1322	0.00033
	60	0.7131	-0.0550	0.00035
	20	0.6427	0.0879	0.00033

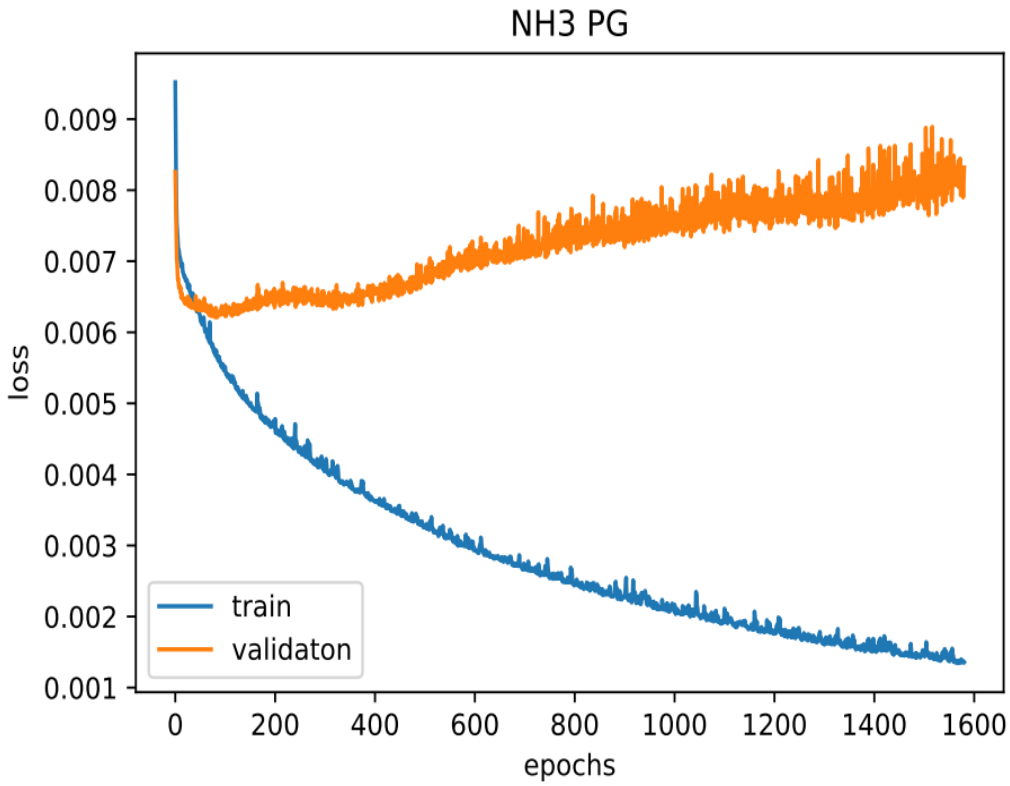


Figure 4.11: Overfitting in NH3 PG ANN model

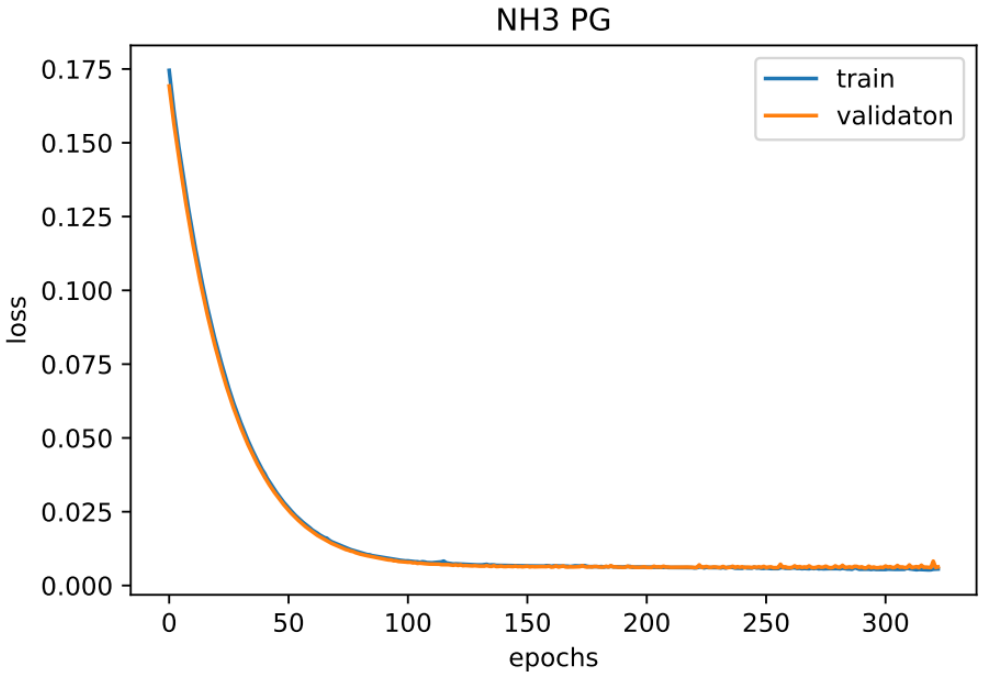


Figure 4.12: Loss versus epochs after using regularization methods

Table 4.15: Final hyperparameter of ANN model for NH3 PG

Degradation type	**#Hidden layer**	**#Nodes in first hidden layer**	**#Nodes in second hidden layer**	**Batch size**	**Epochs**	**Learning rate**	**Train R^2**	**Test R^2**
NH3 PG	2	120	100	40	293	0.00028	0.4808	0.4301

Table 4.16 indicates results for different ANN models for Acetaldehyde PG. Although all models have favorable results showing R^2 more than 0.9 in both training and test dataset, an ANN model with 70 and 100 neurons in the first and second hidden layers, respectively, presents the highest R^2. Therefore, this model is chosen for further improvement. To find the optimal epochs in the model, loss versus epochs is plotted for the validation and train dataset. As shown in Figure 4.13, 1350 epochs can be approximately suitable for epochs in the ANN model. Therefore, final results for the ANN model can be described in the Table 4.17.

Table 4.16: Results for Acetaldehyde PG with ANN method

# Neuron in first hidden layer	# Neuron in second hidden layer	Train R^2	Test R^2	Learning rate
120	100	0.9917	0.8716	0.00037
	60	0.9886	0.9163	0.0019
	20	0.9887	0.9035	0.0039
70	100	0.9900	0.9233	0.0017
	60	0.9912	0.9074	0.0029
	20	0.9923	0.8914	0.0037
40	100	0.9862	0.9001	0.0025
	60	0.9875	0.8688	0.0029
	20	0.9857	0.8995	0.0039

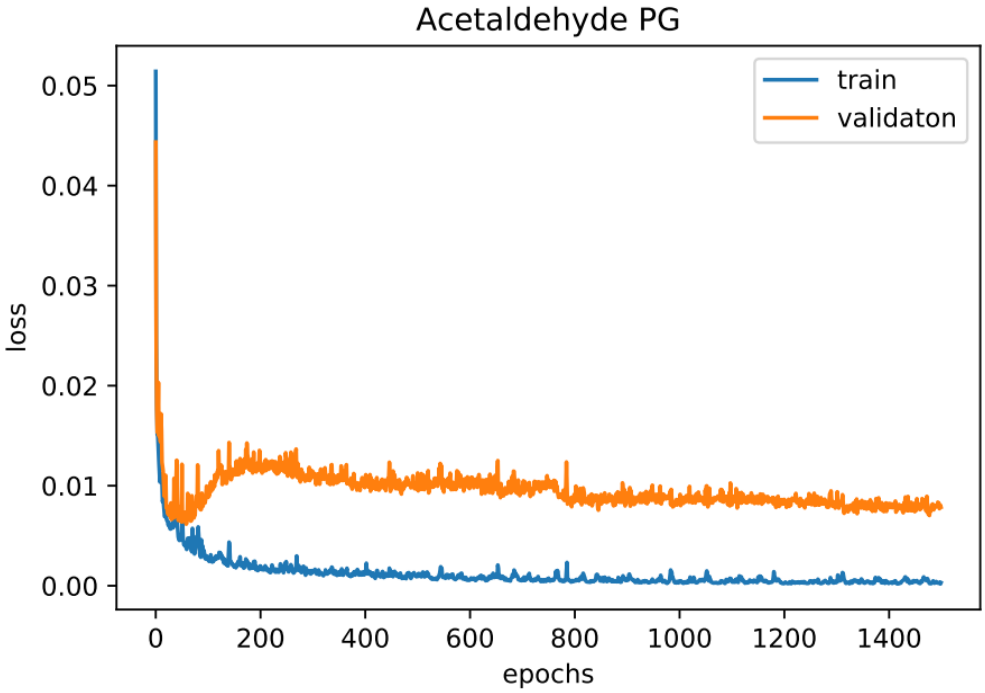


Figure 4.13: Loss versus epochs for Acetaldehyde PG

Table 4.17: Final hyperparameter of ANN model for Acetaldehyde PG

Degradation type	#Hidden layer	#Nodes in first hidden layer	#Nodes in second hidden layer	Batch size	Epochs	Learning rate	Train R^2	Test R^2
Acetaldehyde PG	2	70	100	40	1350	0.0029	0.9911	0.9013

The process of tuning is also executed for Formaldehyde PG. Table 4.18 demonstrates the corresponding results for Formaldehyde PG. As it can be seen, the gap between train and test R^2 can be due to overfitting. Therefore, overfitting is investigated in Formaldehyde PG. As indicated in the Figure 4.14, overfitting has occurred since the gap between validation and train loss increases. After implementing early stopping and L2 regularization method in all cases, an ANN model with 40 and 20 neurons in the first and second hidden layers, respectively, was the best model. Figure 4.15 shows the loss versus epochs after applying regularization methods on the model. Table 4.19 also shows the hyperparameters for the tuned ANN model for Formaldehyde DG.

Table 4.18: Results for Formaldehyde PG with ANN method

# Neuron in first hidden layer	# Neuron in second hidden layer	Train R^2	Test R^2	Learning rate
120	100	0.9521	0.6439	0.0029
	60	0.9513	0.5231	0.0033
	20	0.9529	0.5640	0.0013
70	100	0.9462	0.6097	0.0031
	60	0.9465	0.5862	0.0013
	20	0.9386	0.5829	0.0029
40	100	0.9396	0.4741	0.0035
	60	0.9117	0.7105	0.0013
	20	0.9357	0.6113	0.0029

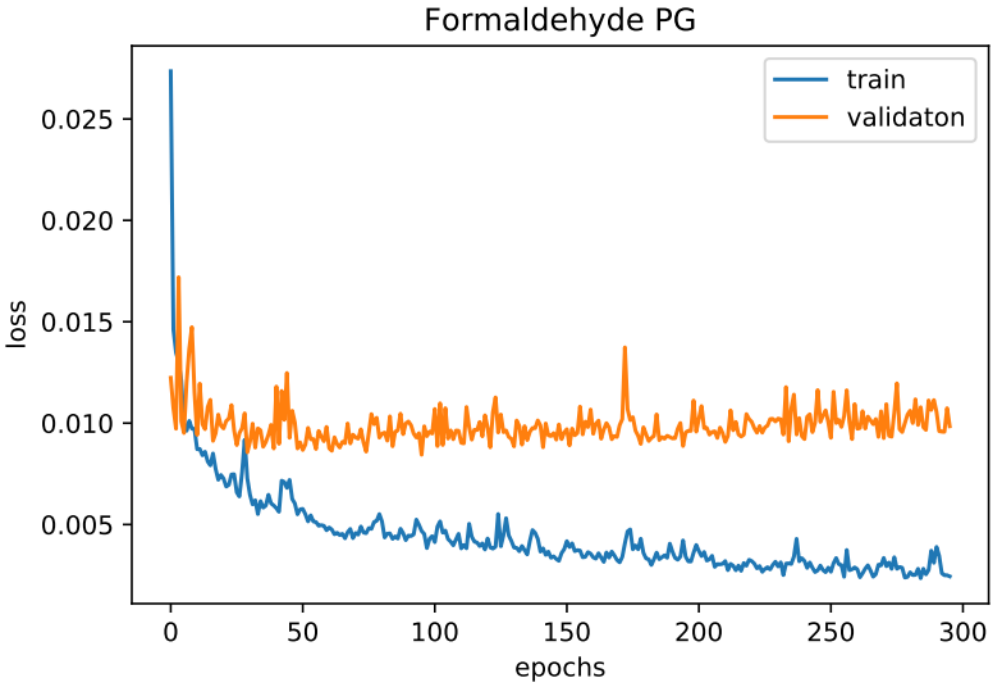


Figure 4.14: Overfitting in Formaldehyde PG ANN model

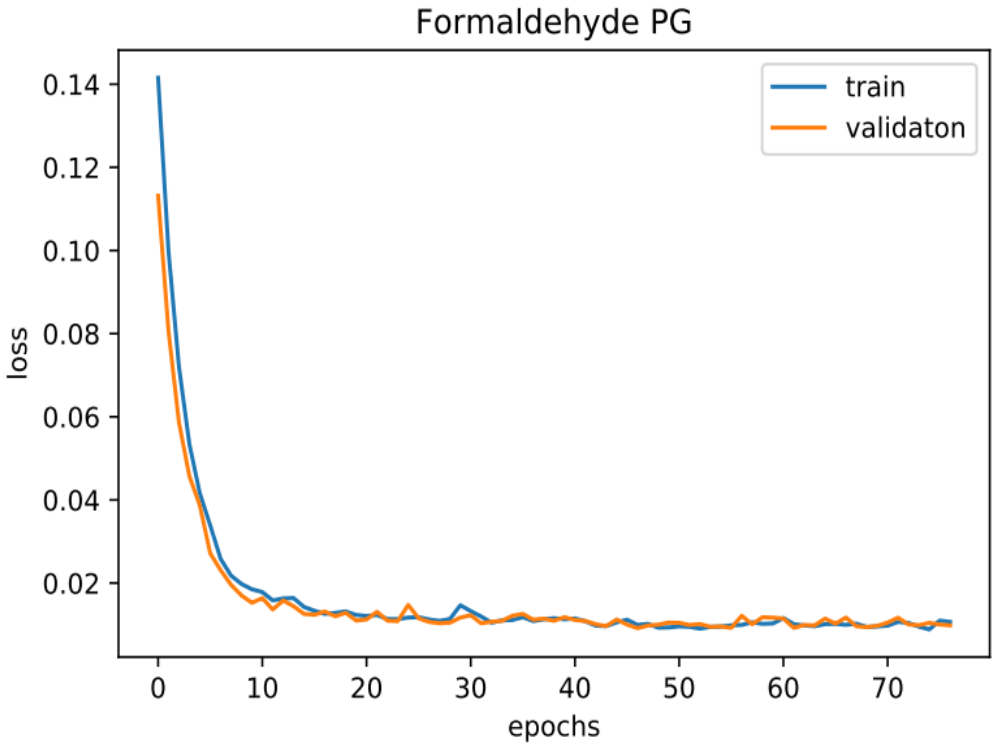


Figure 4.15: Loss vs. epochs after using regularization methods

Table 4.19: Final hyperparameter of ANN model for Formaldehyde PG

Degradation type	**#Hidden layer**	**#Nodes in first hidden layer**	**#Nodes in second hidden layer**	**Batch size**	**Epochs**	**Learning rate**	**Train R^2**	**Test R^2**
Formaldehyde PG	2	40	20	40	48	0.0029	0.7388	0.6782

As there are no acceptable results for train and test R^2 in Ammonium water inlet sec1, number of hidden layers have been increased but no significant change happened in the results. In addition, the last two methods could not predict this type of solvent degradation very well.

To find the hyperparameter of the ANN model for Ammonium in water inlet section 2, the former process is applied. Table 4.20 shows the results of R^2 for different number of neurons for the first and second hidden layers. As it is seen in the table, train and test R^2 are more than 0.98 and 0.8 for all cases. An ANN model with 120 and 100 neurons in the first and second hidden layers is chosen for further process. As shown in the Figure 4.16, 1000 epochs can be approximately suitable in the ANN model. Therefore, results for the ANN model can be described in the Table 4.21.

Table 4.20: Results for Ammonium water inlet sec2 with ANN method

# Neuron in first hidden layer	# Neuron in second hidden layer	Train R^2	Test R^2	Learning rate
120	100	0.9952	0.8310	0.0009
	60	0.9958	0.8231	0.0023
	20	0.9951	0.8264	0.0009
70	100	0.9933	0.8309	0.0033
	60	0.9934	0.8163	0.0037
	20	0.9941	0.8278	0.0033
40	100	0.9896	0.7981	0.0035
	60	0.9838	0.7944	0.0031
	20	0.9915	0.8146	0.0029

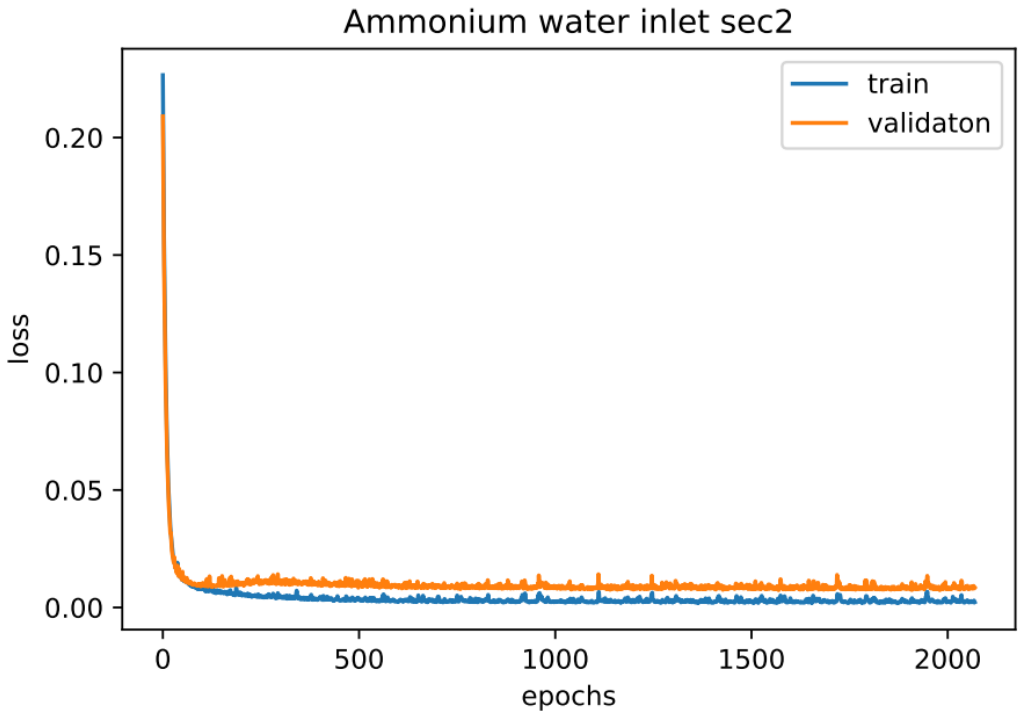


Figure 4.16: Loss versus epochs for Ammonium water inlet sec2

Table 4.21: Final hyperparameter of ANN model for Ammonium water inlet sec2

Degradation type	#Hidden layer	#Nodes in first hidden layer	#Nodes in second hidden layer	Batch size	Epochs	Learning rate	Train R^2	Test R^2
Ammonium Water inlet sec2	2	120	100	40	1000	0.0009	0.9819	0.8202

Table 4.22 demonstrates results for different ANN model for MEA sec1. Although the models have favorable results showing R^2 more than 0.93 in both training and test dataset, an ANN model with 70 and 60 neurons in the first and second hidden layers, respectively, presents the highest R^2. Therefore, this model is chosen for further improvement. To find the optimal epochs in the model, loss versus epochs is plotted for the validation and train dataset. As shown in the Figure 4.17, 300 epochs can be reasonable for the ANN model. Therefore, hyperparameters and R^2 results for MEA sec1 are described in the Table 4.23.

Table 4.22: Results for MEA sec1 with ANN method

# Neuron in first hidden layer	# Neuron in second hidden layer	Train R^2	Test R^2	Learning rate
120	100	0.9941	0.9317	0.0019
	60	0.9946	0.9369	0.0017
	20	0.9940	0.9480	0.0019
70	100	0.9940	0.9444	0.0029
	60	0.9918	0.9451	0.0019
	20	0.9929	0.9421	0.0029
40	100	0.9820	0.9392	0.0019
	60	0.9808	0.9409	0.0017
	20	0.9755	0.9544	0.0019

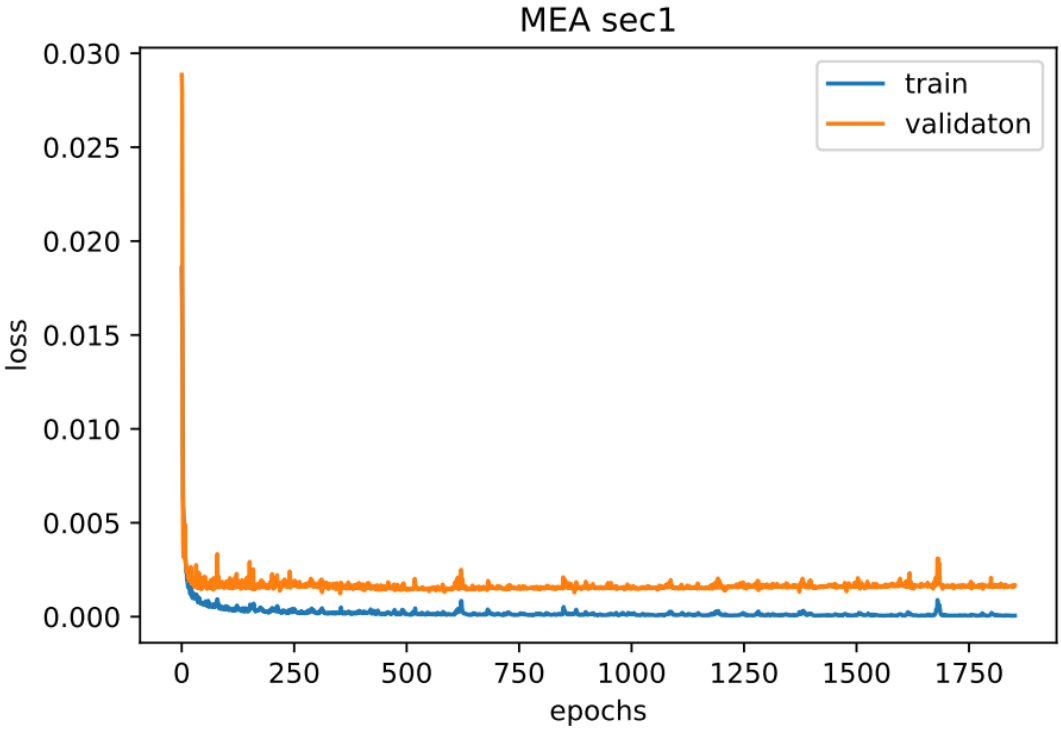


Figure 4.17: Loss versus epochs for MEA sec1

Table 4.23: Final hyperparameter of ANN model for MEA sec1

Degradation type	#Hidden layer	#Nodes in first hidden layer	#Nodes in second hidden layer	Batch size	Epochs	Learning rate	Train R^2	Test R^2
MEA sec1	2	70	60	40	300	0.0019	0.9866	0.9362

Results for the MEA sec2 is shown in Table 4.24. As it is seen, train and test R^2 scores have good results for different number of neurons. Since an ANN model with 70 and 60 neurons in the first and second hidden layers has the best results among others, the relevant tuned hyperparameter for this model is chosen for further improvement. loss function versus epochs is also plotted to find appropriate epochs number for MEA sec2. As shown in the Figure 418, validation and train data set approximately experience no change after 800 epochs. Therefore, final ANN hyperparameters with corresponding R^2 results for MEA sec2 are described in the Table 4.25.

Table 4.24: Results for MEA section 2 with ANN method

# Neuron in first hidden layer	# Neuron in second hidden layer	Train R^2	Test R^2	Learning rate
120	100	0.9799	0.8637	0.0005
	60	0.9808	0.8706	0.00044
	20	0.9830	0.8649	0.00026
70	100	0.962	0.9396	0.00034
	60	0.9819	0.9370	0.00026
	20	0.9818	0.8545	0.00044
40	100	0.9780	0.9310	0.0004
	60	0.9346	0.8989	0.0001
	20	0.9399	0.9305	0.00022

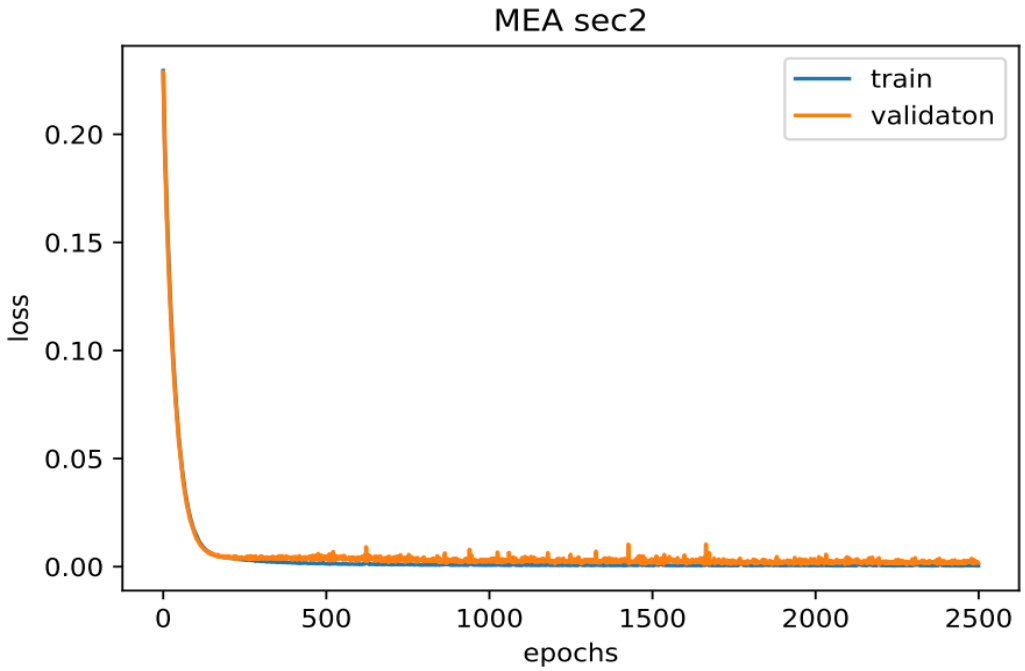


Figure 4.18: Loss versus epochs for MEA sec2

Table 4.25: Final hyperparameter of ANN model for MEA sec2

Degradation type	#Hidden layer	#Nodes in first hidden layer	#Nodes in second hidden layer	Batch size	Epochs	Learning rate	Train R^2	Test R^2
MEA sec2	2	70	60	40	800	0.00026	0.9883	0.9329

Appendix H shows the source code used for ANN models.

4.5 Discussion

In this section, results obtained from all three models for different types of solvent degradation are discussed.

In the feature selection results with Spearman's technique, feature independency was mainly neglected. In fact, the correlation coefficient between features and target was counted while features should be independent as well. The reason was that after removing those features that are related to each other, model results were dramatically unfavorable. Besides, it was seen that CHP stripper temperature has no contribution in all models since its coefficient was less than 0.5. One reason can be investigated in CHP stripper temperature data. As shown in the Figure 4.19, CHP stripper contribution in the operational time is around two months and the main carbon capture operation occurs with RFCC stripper. The other reason relates to the feature selection method used in this study as correlation methods do not perform any learning algorithm to improve the results for the feature selection.

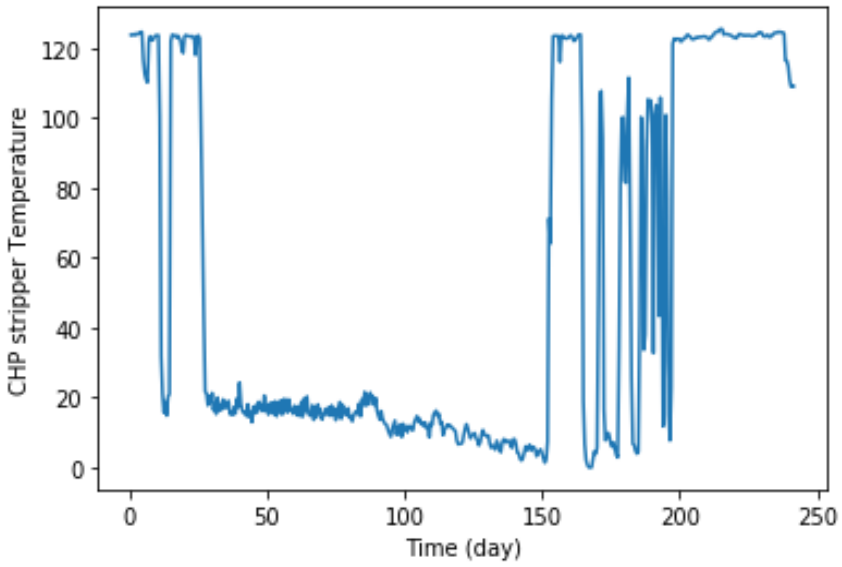


Figure 4.19: CHP stripper temperature over the time

The best model to predict NH3 DG were ANN and RF. Train and test R^2 were more than 0.90 showing very good results. However, generalizability of ANN model was better than RF since test R^2 for ANN was more than RF. SVR results were also satisfying while the first two methods were better.

Random Forest method was introduced as the best model in forecasting the behavior of the Acetaldehyde DG since its R^2 results were significantly higher than the other two methods. However, there is a demand for increasing generalizability of the model as R^2 test is less than 0.85.

In Formaldehyde DG prediction, ANN after applying regularization methods demonstrated the best results. However, there is low R^2 in testing dataset in RF. The reason can be overfitting, but this tried to be avoided by limiting the max_depth which did not affect the results. SVR presents a quite weak model for the Formaldehyde DG.

To predict the NH3 PG, Random Forest had best result in training data. However, R^2 score did not change significantly after restricting max_depth to prevent overfitting. ANN and SVR introduced unacceptable result showing R^2 less than 0.5.

To generalize the Acetaldehyde PG for further independent data, SVR was the best R^2 outcome while the best training score belonged to Random Forest model. ANN also presented acceptable results with R^2 more than 0.9.

In Formaldehyde PG, Random Forest and ANN had better results in training data while SVR presented a better generalizability to predict independent data.

None of the methods could properly forecast the Ammonium in the water wash section 1. In fact, RF demonstrated only good train R^2 while R^2 test was less than 0.15. The other two methods were also unable to give good results. Therefore, further investigation is required to improve this model.

Random Forest and ANN model presented high R^2 in training set for Ammonium in water wash section 2 while the most reliable model for test dataset was SVR with the highest R^2.

In MEA sec1 and MEA sec 2 prediction, ANN and SVR introduced favorable results which can be generalized. Random Forest also demonstrated excellent results for prediction of these types of degradation.

5 Conclusion

The objective of this study was to predict solvent degradation phenomenon by using machine learning methods in carbon capture plant at TCM. This research is consisted of pre-processing data, using different machine learning methods and evaluation of the models.

In the pre-processing step, collected lab and online data were cleaned and finally 483 datasets remained for further implementations. Besides, feature selection methods such as Pearson's and Spearman's technique were utilized to increase the performance of the model by removing redundant and irrelevant features.

Various machine learning methods were used to represent the pattern between the selected features and different types of solvent degradation. Three models of ANN, RF and SVR were implemented, and the corresponding results were demonstrated. Hyperparameters in all methods were tuned to introduce the best possible models for all types of solvent degradation. To optimize the hyperparameter, grid and randomized search optimization methods were used for all models. Results demonstrated that all models forecasted very well except for some cases for instance, NH3 PG and Ammonium water inlet sec1. ANN and RF displayed the favorable results in most cases whereas SVR also presented acceptable models in a few cases. For example, R^2 results for NH3 DG, MEA sec 1, MEA sec2, Acetaldehyde DG, Acetaldehyde PG and Ammonium water inlet sec2 were more than 0.90 in RF and ANN models which showed quite accurate results. SVR also appropriately predicted MEA sec1 and MEA sec2 with high train and test R^2 showing more than 0.97.

There are several recommendations that can be presented for further research in solvent degradation prediction with machine learning methods. Since lab and online data frequency were different and some assumptions have been used to implement all models, there is a demand for better data frequency in lab data. Besides, other feature selection methods such as wrapped and embedded strategy could be applied to reach high resolution results as these methods are based on learning algorithms. To better tune the hyperparameters in all methods, other optimization methods might be utilized to find the global minimum since grid and randomized search are possibly trapped in the local minimums. Other machine learning methods like Recurrent Neural network (RNN) or ANFIS seem to be useful for further implementation as these models were also recommended in the literature review.

References

1. Lille-Mæhlum, Ø.V.H., *Modelling Solvent Degradation in Amine-based Post-combustion Carbon Capture*. 2021, NTNU.
2. Pachauri, R.K., L. Gomez-Echeverri, and K. Riahi, *Synthesis report: summary for policy makers*. 2014.
3. Gerretsen, I. *The state of the climate in 2021*. 11.01.2021 [cited 17.05.2021; Available from:https://www.bbc.com/future/article/20210108-where-we-are-on-climate-change-in-five-charts.
4. *About TCM*. 02.05.2022]; Available from: https://tcmda.com/about-tcm/.
5. Shalaby, A., *Data Driven Modelling and Optimization of MEA Absorption Process for CO2 Capture*. 2020, University of Waterloo.
6. Carpenter, S.M. and H.A. Long III, *Integration of carbon capture in IGCC systems*, in *Integrated Gasification Combined Cycle (IGCC) Technologies*. 2017, Elsevier. p. 445-463.
7. Moioli, S., et al., *Pre-combustion CO2 capture by MDEA process in IGCC based on air-blown gasification.* Energy Procedia, 2014. **63**: p. 2045-2053.
8. Wang, M., et al., *Post-combustion CO2 capture with chemical absorption: A state-of-the-art review.* Chemical engineering research and design, 2011. **89**(9): p. 1609-1624.
9. Li, F., et al., *Modelling of a post-combustion CO2 capture process using extreme learning machine.* International Journal of Coal Science & Technology, 2017. **4**(1): p. 33-40.
10. Conway, J.J.E., *Artificial intelligence and machine learning: Current applications in real estate*. 2018, Massachusetts Institute of Technology.
11. Smola, A.J. and B. Schölkopf, *A tutorial on support vector regression.* Statistics and computing, 2004. **14**(3): p. 199-222.
12. Basak, D., *SP And, and DC Partababis,".* Support vector regression," Neural Inf. Process, 2007.
13. Breiman, L., *Random forests.* Machine learning, 2001. **45**(1): p. 5-32.
14. Fawagreh, K., M.M. Gaber, and E. Elyan, *Random forests: from early developments to recent advancements.* Systems Science & Control Engineering: An Open Access Journal, 2014. **2**(1): p. 602-609.
15. Saxena, S. *A Beginner's Guide to Random Forest Hyperparameter Tuning*. 12.03.2020 17.05.2022]; Available from: https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/.
16. Abiodun, O.I., et al., *State-of-the-art in artificial neural network applications: A survey.* Heliyon, 2018. **4**(11): p. e00938.
17. Kacprzyk, J. and L.C. Jain, *Intelligent Systems Reference Library, Volume 24.* 2012.

18. geeksforgeeks. *Difference between ANN and BNN*. 2020 14 Dec, 2020 02.05.2022]; Available from: https://www.geeksforgeeks.org/difference-between-ann-and-bnn/.
19. Li, J., et al., *Feature selection: A data perspective.* ACM computing surveys (CSUR), 2017. **50**(6): p. 1-45.
20. Liang, H., et al., *Text feature extraction based on deep learning: a review.* EURASIP journal on wireless communications and networking, 2017. **2017**(1): p. 1-12.
21. Verma, V. *A comprehensive guide to Feature Selection using Wrapper methods in Python*. 24.10.2020 17.05.2022]; Available from: https://www.analyticsvidhya.com/blog/2020/10/a-comprehensive-guide-to-feature-selection-using-wrapper-methods-in-python/.
22. Wang, Y., et al., *Efficient test for nonlinear dependence of two continuous variables.* BMC bioinformatics, 2015. **16**(1): p. 1-8.
23. Léonard, G., et al., *Influence of process operating conditions on solvent thermal and oxidative degradation in post-combustion CO2 capture.* Computers & Chemical Engineering, 2015. **83**: p. 121-130.
24. Seo, K., et al., *Modeling and optimization of ionic liquid-based carbon capture process using a thin-film unit.* Computers & Chemical Engineering, 2021. **155**: p. 107522.
25. Morken, A.K., et al., *CO2 capture with monoethanolamine: solvent management and environmental impacts during long term operation at the Technology Centre Mongstad (TCM).* International Journal of Greenhouse Gas Control, 2019. **82**: p. 175-183.
26. Morken, A.K., et al., *Emission results of amine plant operations from MEA testing at the CO2 Technology Centre Mongstad.* Energy Procedia, 2014. **63**: p. 6023-6038.
27. Fløa, N.E., et al., *Results from MEA degradation and reclaiming processes at the CO2 Technology Centre Mongstad.* Energy Procedia, 2017. **114**: p. 1307-1324.
28. Cuccia, L., et al., *Analytical methods for the monitoring of post-combustion CO2 capture process using amine solvents: A review.* International Journal of Greenhouse Gas Control, 2018. **72**: p. 138-151.
29. Cuccia, L., et al., *Monitoring of the blend 1-methylpiperazine/piperazine/water for post-combustion CO2 capture. Part 1: Identification and quantification of degradation products.* International Journal of Greenhouse Gas Control, 2018. **76**: p. 215-224.
30. Fløa, N.E., et al., *Assessment of material selection for the CO2 absorption process with aqueous MEA solution based on results from corrosion monitoring at Technology Centre Mongstad.* International Journal of Greenhouse Gas Control, 2019. **84**: p. 91-110.
31. Bontemps, D., et al., *LEMEDES-CO2: a lab for studying degradation of solvents used for CO2 capture post-combustion amine based systems.* Energy Procedia, 2014. **63**: p. 787-790.
32. Rieder, A., et al., *Understanding solvent degradation: A study from three different pilot plants within the OCTAVIUS project.* Energy Procedia, 2017. **114**: p. 1195-1209.
33. Léonard, G., D. Toye, and G. Heyen, *Relevance of accelerated conditions for the study of monoethanolamine degradation in post-combustion CO2 capture.* The Canadian Journal of Chemical Engineering, 2015. **93**(2): p. 348-355.

34. Rahimi, M., et al., *Toward smart carbon capture with machine learning.* Cell Reports Physical Science, 2021. **2**(4): p. 100396.

35. Amar, M.N., et al., *Modeling viscosity of CO2 at high temperature and pressure conditions.* Journal of Natural Gas Science and Engineering, 2020. **77**: p. 103271.

36. Dureckova, H., et al., *Robust machine learning models for predicting high CO2 working capacity and CO2/H2 selectivity of gas adsorption in metal organic frameworks for precombustion carbon capture.* The Journal of Physical Chemistry C, 2019. **123**(7): p. 4133-4139.

37. Menad, N.A., et al., *Predicting solubility of CO2 in brine by advanced machine learning systems: Application to carbon capture and sequestration.* Journal of CO2 Utilization, 2019. **33**: p. 83-95.

38. Mesbah, M., et al., *Accurate prediction of miscibility of CO2 and supercritical CO2 in ionic liquids using machine learning.* Journal of CO2 Utilization, 2018. **25**: p. 99-107.

39. Leperi, K.T., et al., *110th anniversary: surrogate models based on artificial neural networks to simulate and optimize pressure swing adsorption cycles for CO2 capture.* Industrial & Engineering Chemistry Research, 2019. **58**(39): p. 18241-18252.

40. Li, F., *Modelling and optimisation of post-combustion carbon capture process integrated with coal-fired power plant using computational intelligence techniques*. 2018, Newcastle University.

41. Caruana, R., S. Lawrence, and C. Giles, *Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping.* Advances in neural information processing systems, 2000. **13**.

42. Srivastava, N., et al., *Dropout: a simple way to prevent neural networks from overfitting.* The journal of machine learning research, 2014. **15**(1): p. 1929-1958.

43. Dikov, G. and J. Bayer. *Bayesian learning of neural network architectures*. in *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019. PMLR.

44. Fareed, H. *Machine Learning for Dummies*. 07.04.2018 17.05.2022]; Available from: https://towardsdatascience.com/wtf-is-machine-learning-a-quick-guide-39457e49c65b.

Appendices

Appendix A – Task description

Appendix B – A part of TCM online data

Appendix C – A part of TCM offline data

Appendix D – Pre-processing source code for offline data

Appendix E – Spearman's and Pearson's source code

Appendix F – Support Vector Regression (SVR) source code

Appendix G – Random Forest source code

Appendix H – Artificial Neural Network (ANN) source code

Appendix A - Task description

University of South-Eastern Norway

Faculty of Technology, Natural Sciences and Maritime Sciences, Campus Porsgrunn

FMH606 Master's Thesis

Title: **Development of machine learning model for CO2 capture plants to predict solvent degradation**

USN supervisors: Leila Ben Saad and Ru Yan

External partner: Technology Centre Mongstad (TCM) v/ Rune Teigland

Task background:

Technology Centre Mongstad (TCM) is the world's largest and most flexible test center for developing CO2 capture technologies and a leading competence center for carbon capture. TCM has been operating since autumn 2012, providing an arena for qualification of CO2 capture technologies on an industrial scale. A vast amount of data is collected from more than 1000 online instruments in the amine plant and more than 1100 in the utility plant. In addition, there are multiple sampling points for liquid sampling throughout the amine plant. Hence a vast amount of data is readily available to be exploited.
Machine Learning (ML) can be applied when you have a complex task or problem involving Big data and many variables, but you do not know the formula/equation or classic regression methods do not fit well.
Machine Learning methods has the potential to design, test and improve various aspects of the CO2 process that are computationally time consuming or experimentally time consuming and expensive. The use of ML techniques for carbon capture processes [1-4] is still emerging and most investigations have been on simplified models.
The objective of this project is to build data driven models that can enhance the understanding of relationships among key process parameters. The data available from TCM open campaigns will be used as training data sets for ANN model to be built. The aim of ANN model is to provide an analysis of the extracted rules and reveal most significant relationships between them.
In this project, the focus will be mainly on the problem of solvent degradation [5]. The use of ANN model in this context will allow to predict the solvent degradation.

Task description:

- Give an overview of CO2 capture technologies with focus post combustion carbon capture
- Understand the problem of solvent degradation and solvent and its accumulation in CO2 capture process
- Data analysis of TCM database to extract the most suitable data to deal with solvent degradation phenomena
- Review and inspect the state-of-the-art of adopting machine learning in the context of CO2 capture process.
- Describe the steps of developing data driven models using a machine learning approach.
- Analyze and investigate various approaches utilizing Artificial Neural Networks to find the most suitable models for this application.
- Develop some of these models based on data from TCM and discuss the accuracy of these models

Signatures:

Supervisor (date and signature): 01/02/2022
Leila Ben Saad

Student (write clearly in all capitalized letters):
SAM NARIMANI

Student (date and signature):
5/16/2022
Sam Narimani

Appendix B – TCM online data

Date	CO2 AI (Sm3/h)	O2 AI (Sm3/h)	H2O AI (Sm3/h)	CHP stripper temp. (C)	RFCC stripper temp. (C)	NH3 DG (SL/h)	Acetaldehyde DG (SL/h)	Formaldehyde DG (SL/h)	NH3 PG (g/h)	Acetaldehyde PG (g/h)	Formaldehyde PG (g/h)
01.07.2017 12:00	1839	6786	2049	124	24	119	0.000	0.961	25.8	41.1	11.2
02.07.2017 00:00	1853	6789	2036	124	21	75	0.000	0.415	22.9	35.1	9.8
02.07.2017 12:00	1874	6814	2054	124	20	60	0.000	0.435	23.5	29.9	9.0
03.07.2017 00:00	1966	6694	2100	124	19	44	0.000	0.125	26.3	23.2	9.8
03.07.2017 12:00	1983	6668	2120	124	20	47	0.000	0.586	32	16.2	14.7
04.07.2017 00:00	1984	6645	2075	124	18	40	0.000	0.500	29	11.4	11.7
04.07.2017 12:00	1987	6649	2076	125	20	30	0.000	0.253	32.6	4.7	13.4
05.07.2017 00:00	1990	6616	2055	125	17	90	0.000	0.017	27.8	5.8	25.1
05.07.2017 12:00	1310	4400	1374	117	19	249	0.000	2.633	40.7	32.9	6.1
06.07.2017 00:00	1043	3520	1114	113	16	306	0.000	1.346	34.3	29.6	7.3
06.07.2017 12:00	1054	3561	1004	111	18	306	0.008	2.475	41.3	27.5	13.5
07.07.2017 00:00	1131	3671	1003	110	15	269	0.000	0.869	36.6	18.6	16.7
07.07.2017 12:00	2010	6470	1954	122	15	278	0.002	0.166	34.8	32.9	12.5
08.07.2017 00:00	1935	6248	2008	123	14	391	0.000	0.296	36.1	31.4	8.9

Appendix C – TCM offline data

Date	Viscosity lAAI (mPa.s)	Ammonium Water inlet sec1 (kg/h)	Ammonium Water inlet sec2 (kg/h)	MEA sec1 (kg/h)	MEA sec2 (kg/h)
01.07.2017 00:00	2.817	0.374	2.370	88.800	6.180
01.07.2017 12:00	2.817	0.374	2.370	88.802	6.180
02.07.2017 00:00	2.817	0.374	2.370	88.798	6.180
02.07.2017 12:00	2.817	0.374	2.370	88.800	6.180
03.07.2017 00:00	2.555	0.109	1.092	113.996	1.692
03.07.2017 12:00	2.555	0.109	1.092	114.004	1.692
04.07.2017 00:00	3.607	0.109	1.092	114.001	1.692
04.07.2017 12:00	3.607	0.109	1.092	114.007	1.692
05.07.2017 00:00	3.607	0.355	3.402	102.004	2.394
05.07.2017 12:00	3.607	0.355	3.402	101.999	2.394
06.07.2017 00:00	3.200	0.355	3.402	102.002	2.394
06.07.2017 12:00	3.200	0.355	3.402	101.995	2.394
07.07.2017 00:00	1.844	4.272	7.200	1.128	2.178
07.07.2017 12:00	1.844	4.272	7.200	1.128	2.178
08.07.2017 00:00	1.844	4.272	7.200	1.128	2.178
08.07.2017 12:00	1.844	4.272	7.200	1.128	2.178
09.07.2017 00:00	1.844	4.272	7.200	1.128	2.178
09.07.2017 12:00	1.844	4.272	7.200	1.128	2.178
10.07.2017 00:00	3.602	1.170	5.328	84.001	1.044
10.07.2017 12:00	3.602	1.170	5.328	83.992	1.044
11.07.2017 00:00	2.758	1.170	5.328	83.995	1.044
11.07.2017 12:00	2.758	0.802	4.240	57.595	0.831
12.07.2017 00:00	2.758	1.560	7.104	111.994	1.392
12.07.2017 12:00	2.758	1.560	7.104	112.003	1.392
13.07.2017 00:00	3.215	1.560	7.104	112.002	1.392
13.07.2017 12:00	3.215	1.560	7.104	111.998	1.392
14.07.2017 00:00	3.879	7.112	19.359	1.432	1.272
14.07.2017 12:00	3.879	5.812	15.818	1.170	1.039
15.07.2017 00:00	3.879	4.501	12.259	0.906	0.805
15.07.2017 12:00	3.879	5.334	14.520	1.074	0.954
16.07.2017 00:00	3.879	5.334	14.520	1.074	0.954
16.07.2017 12:00	3.879	5.334	14.520	1.074	0.954
17.07.2017 00:00	2.398	1.086	4.176	119.997	1.626
17.07.2017 12:00	2.398	1.086	4.176	120.000	1.626
18.07.2017 00:00	2.534	1.086	4.176	119.995	1.626
18.07.2017 12:00	2.534	1.086	4.176	119.998	1.626
19.07.2017 00:00	2.411	0.870	4.560	135.001	6.960

Appendix D - Pre-processing code for lab data

```
import pandas as pd

data=pd.read_csv(' File Directory
                  , usecols=['Description','Sampled
date','Analsis','Component name','Result text','Units'],
                  index_col=0)

# Wash water inlet, sec.1 (WWIS1)

WWIS1=data.loc['Wash water inlet, sec.1']
Amine_WWIS1=WWIS1.loc[(WWIS1['Component name']=='Am1')|(WWIS1['Com-
ponent name']=='Am01')|(WWIS1['Component name']=='Am1_mg/kg')]
Ammonium_WWIS1=WWIS1.loc[(WWIS1['Component
name']=='NH4+')|(WWIS1['Component name']=='Ammonium')]

# Wash water inlet, sec.2 (WWIS2)

WWIS2=data.loc['Wash water inlet, sec.2']
Amine_WWIS2=WWIS2.loc[(WWIS2['Component name']=='Am1')|(WWIS2['Com-
ponent name']=='Am01')|(WWIS2['Component name']=='Am1_mg/kg')]
Ammonium_WWIS2=WWIS2.loc[(WWIS2['Component
name']=='NH4+')|(WWIS2['Component name']=='Ammonium')]

# Lean amine - absorber inlet (LAAI)

LAAI=data.loc['Lean amine - absorber inlet']
Amine_LAAI=LAAI.loc[LAAI['Component name']=='Am1_mg/kg']
TN_LAAI=LAAI.loc[LAAI['Component name']=='Total Nitrogen']
HEA_LAAI=LAAI.loc[LAAI['Component name']=='HEA_wt']
HEF_LAAI=LAAI.loc[LAAI['Component name']=='HEF_wt']
HEPO_LAAI=LAAI.loc[LAAI['Component name']=='HEPO_wt']
HEGly_LAAI=LAAI.loc[LAAI['Component name']=='HEGly_wt']
Nitrate_LAAI=LAAI.loc[LAAI['Component name']=='Nitrate']
Nitrite_LAAI=LAAI.loc[LAAI['Component name']=='Nitrite']

# Rich amine (RA)

RA=data.loc['Rich amine - downstream make-up & filter']
Amine_RA=RA.loc[RA['Component name']=='Am1_mg/kg']
```

Appendix E – Spearman's and Pearson's code

```
from pandas import DataFrame
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


data = pd.read_csv (' File Directory ')
data=data.iloc[ : , ]


# Pearson's Corrolation

cor_pearson= DataFrame.corr(data)

plt.figure(figsize=( , ))
sns.heatmap(cor_pearson, annot=True)
plt.savefig("CorrolationPearson.pdf")

# Spearman's Correlation

cor_spearman = data.corr(method='spearman')
plt.figure(figsize=( , ))
sns.heatmap(cor_spearman, annot=True)
plt.savefig("CorrolationSpearman.pdf")
```

Appendix F – Support Vector Regression (SVR) code

```
import pandas as pd
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score


data = pd.read_csv('File Directory')

# Features (Input)
x = data.iloc[:, ].values
x = (x-x.min())/(x.max()-x.min())

# Label (Output)
y = data.iloc[:, ].values
y = (y-y.mean())/(y.max()-y.min())

# Splitting data into train and test dataset
x_train , x_test , y_train , y_test = train_test_split(x,y, test_size=0.2,
random_state = 1)

# SVR model
classifier= SVR(kernel = 'rbf',epsilon= ,gamma='scale',tol= 0.00001 ,C =  )

# Fit the model
classifier= classifier.fit(x, y)

# Predict the result for train and test dataset
y_pred_train = classifier.predict(x_train)
y_pred_test = classifier.predict(x_test)

# R2 results for each model
R2_train_set= r2_score(y_train , y_pred_train)
R2_test_set= r2_score(y_test , y_pred_test)
```

Appendix G – Random Forest code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.model_selection import RandomizedSearchCV


data = pd.read_csv('file Directory')

# Features (Input)
x = data.iloc[:, ].values
x = (x-x.min())/(x.max()-x.min())

# Label (Output)
y = data.iloc[:, ].values
y = (y-y.mean())/(y.max()-y.min())

# Splitting data into train and test dataset
x_train , x_test , y_train , y_test = train_test_split(x,y, test_size=0.2,
random_state = 0)

# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 20, stop = 1000, num =
10)]

# Number of features in every split
max_features = ['auto', 'sqrt']

# Maximum number of tree depth
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)

# Minimum number of samples in splitting a node
min_samples_split = [2, 5, 8]

# Minimum number of samples needed in each leaf node
min_samples_leaf = [1, 2, 4]

# Selecting sample method for training each tree
bootstrap = [True, False]

# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}



# Random Forest model
rf = RandomForestRegressor()
```

```
# Random search of parameters, using 3-fold (k=3) cross validation
rf_random = RandomizedSearchCV(estimator = rf, param_distributions = random
grid, n_iter = 100, cv = 3, verbose=2, random_state=42, n_jobs = -1)

# Fit the model
rf_random.fit(x_train, y_train)

# Best parameter in each model
best_param = rf_random.best_params_

# Predict the result for train and test dataset
y_pred_train = rf_random.predict(x_train)
y_pred_test = rf_random.predict(x_test)

# R2 results for each model
R2_train_set= r2_score(y_train , y_pred_train)
R2_test_set= r2_score(y_test , y_pred_test)

# Print R2 results for each model
print( 'R2 for train set is : ', R2_train_set)
print( 'R2 for test set is : ', R2_test_set)

# plot Random Forest
fig = plt.figure(figsize=(15, 10))
plot_tree(rf_random.estimators_[ ])
plt.show()
```

Appendix H - Artificial Neural Network (ANN) code

```
import pandas as pd
import matplotlib.pyplot as plt
from tensorflow.keras.layers import Dense
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras import backend as K
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import r2_score

data = pd.read_csv ( File Directory)

data = pd.DataFrame(data)

# Features (Input)
x = data.iloc[:,  ]
x = (x-x.min())/(x.max()-x.min())

# Label (Output)
y = data.iloc[:, ]
y = (y-y.min())/(y.max()-y.min())

# Splitting data into train and test dataset
x_train , x_test , y_train , y_test = train_test_split(x,y, test_size=0.2,
random_state = 0)


# Defining R2 score
def det_coeff(y_true, y_pred):
    SS_res = K.sum(K.square(y_true - y_pred))
    SS_tot = K.sum(K.square(y_true - K.mean(y_true)))
    return K.ones_like(SS_tot) - (SS_res / SS_tot)

# Tuned learning rate
learning_rate=

classifier = Sequential()

classifier.add(Dense(units=  , kernel_initializer = 'glorot_uniform', acti-
vation = 'relu', input_dim =  ))

classifier.add(Dense(units=  , kernel_initializer = 'glorot_uniform', acti-
vation ='tanh'  ))

classifier.add(Dense(units= 1, kernel_initializer = 'glorot_uniform', acti-
vation = 'relu'))


opt=Adam(learning_rate)

classifier.compile(optimizer =opt , loss = 'mse', metrics = [det_coeff])

# Fit the model
classifier.fit(x_train, y_train, batch_size = 40, epochs =  )
```

```
# Predict the result for train and test dataset
y_pred_train = classifier.predict(x_train)

y_pred_test = classifier.predict(x_test)

# R2 results for each model
R2_train_set= r2_score(y_train , y_pred_train)
R2_test_set= r2_score(y_test , y_pred_test)

# Print R2 results for each model
print( 'R2  for train set is : ', R2_train_set )
print( 'R2 for test set is : ', R2_test_set )
```