

## Article

# Agent-Based and Stochastic Optimization Incorporated with Machine Learning for Simulation of Postcombustion CO<sub>2</sub> Capture Process

Huilan Zheng<sup>1</sup>, Gaurav Mirlekar<sup>1,2,\*</sup>  and Lars O. Nord<sup>1</sup> 

<sup>1</sup> Department of Energy and Process Engineering, NTNU—Norwegian University of Science and Technology, Kolbjørn Hejes vei 1 B, 7491 Trondheim, Norway

<sup>2</sup> Department of Electrical Engineering, Information Technology and Cybernetics, USN—University of South-Eastern Norway, Campus Porsgrunn, 3199 Borre, Norway

\* Correspondence: gaurav.mirlekar@usn.no

**Abstract:** In this paper, a novel method is proposed for the incorporation of data-driven machine learning techniques into process optimization. Such integration improves the computational time required for calculations during optimization and benefits the online application of advanced control algorithms. The proposed method is illustrated via the chemical absorption-based postcombustion CO<sub>2</sub> capture process, which plays an important role in the reduction of CO<sub>2</sub> emissions to address climate challenges. These processes simulated in a software environment are typically based on first-principle models and calculate physical properties from basic physical quantities such as mass and temperature. Employing first-principle models usually requires a long computation time, making process optimization and control challenging. To overcome this challenge, in this study, machine learning algorithms are used to simulate the postcombustion CO<sub>2</sub> capture process. The extreme gradient boosting (XGBoost) and support vector regression (SVR) algorithms are employed to build models for prediction of carbon capture rate (CR) and specific reboiler duty (SRD). The R<sup>2</sup> (a statistical measure that represents the fitness) of these models is, on average, greater than 90% for all the cases. XGBoost and SVR take 0.022 and 0.317 s, respectively, to predict CR and SRD of 1318 cases, whereas the first-principal process simulation model needs 3.15 s to calculate one case. The models built by XGBoost are employed in the optimization methods, such as an agent-based approach represented by the particle swarm optimization and stochastic technique indicated by the simulated annealing, to find specific optimal operating conditions. The most economical case, in which the CR is 72.2% and SRD is 4.3 MJ/kg, is obtained during optimization. The results show that computations with the data-driven models incorporated in the optimization technique are faster than first-principle modeling approaches. Thus, the application of machine learning techniques in the optimization of carbon capture technologies is demonstrated successfully.

**Keywords:** postcombustion carbon capture; machine learning; process optimization; data-driven process modeling



**Citation:** Zheng, H.; Mirlekar, G.; Nord, L.O. Agent-Based and Stochastic Optimization Incorporated with Machine Learning for Simulation of Postcombustion CO<sub>2</sub> Capture Process. *Processes* **2022**, *10*, 2727. <https://doi.org/10.3390/pr10122727>

Academic Editors: Xiao Feng and Minbo Yang

Received: 24 October 2022

Accepted: 12 December 2022

Published: 16 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

According to the World Meteorological Organization (WMO), the world experienced the warmest weather on record because of greenhouse gas emissions (GHG) in the last decade (2011–2020). The temperature difference between the preindustrial baseline (1850–1900) and global mean surface temperature in 2020 has reached 1.2 °C [1]. The action is on call for addressing climate changes as reported in this report. To avoid the worst climate impact, carbon capture and storage (CCS) technologies have been developed. CCS is the process of capturing CO<sub>2</sub> before it enters the atmosphere, transporting and storing it for centuries or millennia [2]. Currently, the most mature technology is CO<sub>2</sub> capture, which can be realized through precombustion capture, oxy-fuel combustion capture, and

postcombustion capture (PCC) [3]. From an economic and practical perspective, PCC is preferred, as it can be retrofitted to an existing plant. Many research efforts now focus on topics within the PCC process, such as process simulation, solvent degradation, and equipment design. This study also contributes to the simulation of PCC process in order to better control and optimize the process efficiency. This technology is a process that consists of the chemical absorption and thermal-stripping route. The CO<sub>2</sub> in the flue gas stream is captured by an absorbent solvent, such as monoethanolamine (MEA) and diethanolamine (DEA) in the chemical absorption process. In the stripping process, the CO<sub>2</sub>-loaded solvent is sent to a stripper, the pure CO<sub>2</sub> is released, and the absorbent solvent is regenerated [4].

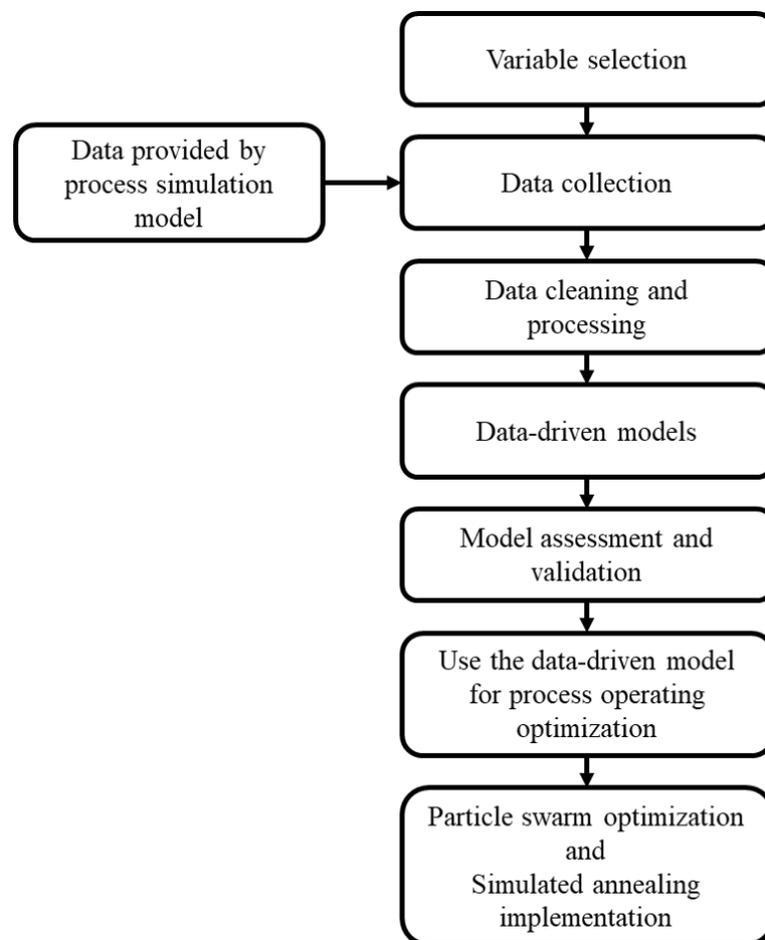
The PCC process studied in this paper is with 30 wt% MEA chemical solvent because it is regarded as the benchmark solvent in academic work and industrial research. In order to study the process behavior of the PCC process, the first-principal model simulation is widely used. The process simulation software, such as Aspen HYSYS<sup>®</sup>, Aspen Plus<sup>®</sup>, gPROMS<sup>®</sup>, and CO2SIM<sup>®</sup> are used to build models to calculate the mass flow rate of stream, temperature, pressure, and other physical quantities in PCC for analytical purposes [5–8]. Øi investigated the difference between equilibrium and rate-based models built by Aspen HYSYS<sup>®</sup> and Aspen Plus<sup>®</sup>. It is reported that the rate-based models in software are close to the equilibrium model when calculating CO<sub>2</sub>-removal efficiency as a function of circulation rate, number of column stages, and inlet temperature [9]. In general, PCC process simulations are developed by using first-principle models and compared for the accuracy of these models with real plants. In addition, some studies improve the accuracy of simulation models. To improve a simulation model's predictions for experimental and pilot results, Spek et al. upgraded the software package by using the regression of activated 2-amino-2-methyl-1-propanol(AMP)–piperazine (PZ) binary interaction parameters [10]. Luo et al. built a steady-state model, which is rate-based for the MEA-based PCC process. This model is validated against thermodynamic and physical properties over a wide range of pressures, temperatures, and CO<sub>2</sub> loadings [11]. Dutta et al. built an equilibrium-based absorber model by using the reduced stage efficiencies in order to reduce the computational time for simulation. This model is used to predict operating conditions within an accepted range [12]. Nonetheless, the implementation of further optimization and control based on first-principle simulation model is challenged as the time required for solving differential and algebraic equations in this simulation model is approximately 4 s for one case. Therefore, the alternative modeling method must be investigated and machine learning turned out to be a promising one. Fernandez et al. has proven that the machine learning techniques can be used to recognize high-performing metal-organic framework materials in the CO<sub>2</sub> capture process [13]. Venkatraman et al. used the machine learning technique to model structure–property relationships between molecular structures of cations and anions and their CO<sub>2</sub> solubilities in the ionic liquid (IL)-based CO<sub>2</sub> capture process [14]. Ahmadi et al. investigated the possibility of using an artificial neural network (ANN) and least squares support vector machine (LSSVM) approaches to predict the viscosity and thermal conductivity of CO<sub>2</sub>. Both methods are proven to be effective [15]. In order to change the CO<sub>2</sub> capture rate quickly and smoothly in a wide operating range, Wu et al. performed a nonlinearity analysis of the multimodel predictive control strategy for solvent-based postcombustion CO<sub>2</sub> capture plants [16]. Rahimi et al. discussed how machine learning implementations have improved the carbon capture process in absorption- and adsorption-based approaches, ranging from the molecular to the process level. In particular, the application of reinforcement learning to obtain the optimized operating parameters is highlighted [17]. Shalaby et al. applied Matérn Gaussian process regression (GPR), rational quadratic GPR, squared exponential GPR models, and a feed-forward ANN model in predicting the output parameters of the PCC process simulation in gPROMS<sup>®</sup>. The system's energy requirement, capture rate, and the purity of the condenser outlet stream are accurately predicted [18]. Shahsavand et al. applied the backpropagation multilayer perceptron (BPMLP) and the radial basis function (RBF) neural networks to understand packed absorption processes, especially the absorption of CO<sub>2</sub> from the air by various alkanolamine

solutions [19]. The result showed that the RBF networks perform more adequately than the MLP network for filtering noise and capturing the real trend. Chan et al. used the neural network rule extraction algorithm to reveal that the steam flow rate through reboiler, reboiler pressure, and the CO<sub>2</sub> concentration in the flue gas are the three most significant parameters of the CO<sub>2</sub> production rate [20]. Li et al. predicted the CO<sub>2</sub> capture rate by applying the bootstrap aggregated extreme learning machine (ELM) to build models based on the data collected from the gPROMS<sup>®</sup> simulation model. The ELM model performs well as it has a small mean square error (MSE) and reduces computational time [21]. The possibility of the application of machine learning technique has been demonstrated through research in the literature. However, the extreme gradient boosting (XGBoost) and support vector regression (SVR) have not been investigated yet. This study aims to fill this research gap of application of new algorithms. A mature machine learning-based modeling method is introduced in this paper. Additionally, these advanced machine learning techniques are applied in the context of process optimization. In this study, the process modeling approach is data driven with the application of new algorithms (XGBoost and SVR) that improve the prediction accuracy of PCC process parameters. In addition, the faster computational time of these data-driven techniques to model the PCC process is advantageous during optimization and control studies.

In the literature, the application of data-driven models for the optimization and control of the energy systems has been investigated [22,23]. Mirlekar et al. employed the autoregressive model with the exogenous inputs (ARX) method in the development of biologically inspired optimal control strategy (BIOCS) for implementation on a subsystem of a CO<sub>2</sub> capture process [23]. Saboori et al. studied the application of particle swarm optimization (PSO) in a multistage generation expansion planning (GEP) which includes nuclear units, renewable energy units, and different fossil fuel-fired units equipped with CCS [24]. Amar et al. used the genetic algorithm (GA), PSO, and artificial bee colony (ABC) to optimize the radial basis function neural network (RBFNN) models employed for modeling of CO<sub>2</sub> solubility in brine [25]. Simulated annealing (SA) has also been investigated in the study of the PCC process. SA is an optimization method based on the analogy between the simulation of the annealing of solids and solving large combinatorial optimization problems [26]. Mores et al. developed an optimized radial basis function based on grid search coupled with SA and tenfold cross-validation algorithms. This intelligent correlation model was used to predict the mass transfer coefficient for CO<sub>2</sub> capture with NaOH solution in different types of rotating packed beds [27]. Dashti et al. presented unique computational models to estimate CO<sub>2</sub> solubility in commonly used amines. A series of models, including the genetic algorithm-adaptive neurofuzzy inference system (GA-ANFIS), PSO-ANFIS, coupled simulated annealing-least squares support vector machine (CSA-LSSVM) and RBF neural networks were developed to estimate CO<sub>2</sub> equilibrium absorption capacity in 12 aqueous amine solutions [28]. In this paper, a systematic method is proposed to study the application of the machine learning technique for reducing computational time in the simulation of the PCC process. The PCC process simulation model is developed in Aspen HYSYS<sup>®</sup> in order to generate original data for the study. The characteristics of the datasets are summarized and analyzed before the application of machine learning. XGBoost and SVR algorithms are employed for building data-driven models with the objective of predicting the CO<sub>2</sub> CR and specific reboiler duty (SRD) in the PCC process. These models are further incorporated into optimization techniques represented by PSO and SA. Thus, the optimal process operating conditions are obtained. In general, the SRD increases with the CR, whereas the cost-effective situation deals with higher CR and lower SRD. Therefore, the optimized operating condition would be the one with relatively high CR and low SRD. Such an integration would help in improving online advanced control implementation and dynamic optimization of the process. The paper is organized as follows. The method is described in Section 2; the implementation results are discussed in Section 3; and in Section 4, the summary and conclusions are presented.

## 2. Method

In this section, the PCC process simulation, the machine learning algorithms incorporated into the optimization methods and the optimization techniques are described. The workflow diagram considered in this study is shown in Figure 1. The large amount of data required for machine learning from a pilot plant is unavailable. To overcome this challenge, a steady-state process simulation model of the PCC is set up in Aspen HYSYS<sup>®</sup> [12]. This model is used for generating the raw data for data-driven modeling purposes. These data are summarized and analyzed, and then the XGBoost and SVR algorithms are applied to develop data-driven models. The assessment indicators, such as coefficient of determination ( $R^2$ ) and MSE, are used to measure model performance. In the model validation step, a new dataset is generated specifically for validation purposes. The data-driven models are further used in optimization for searching the optimal operating conditions, such as the largest CR or lowest SRD. The PSO and SA are adopted to perform optimization.

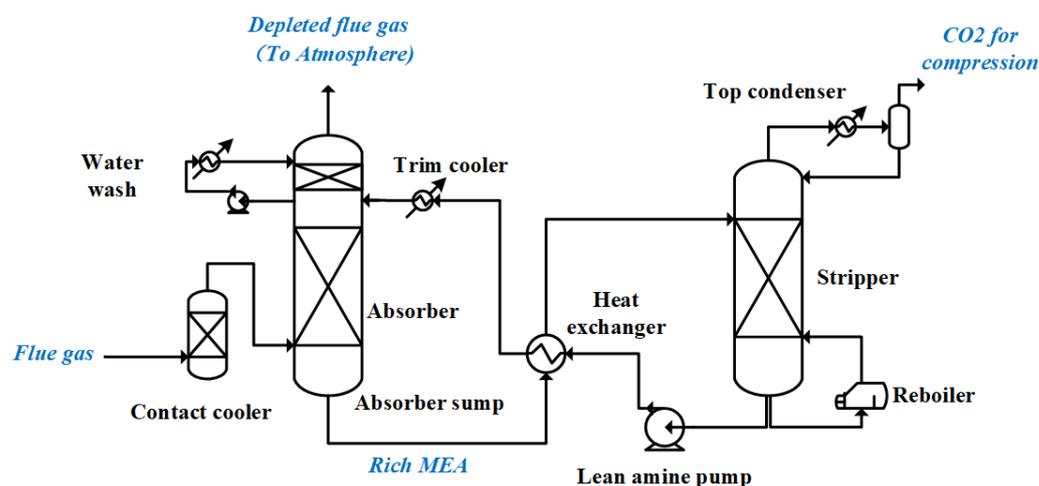


**Figure 1.** Workflow diagram of the proposed method for data-driven process modeling and optimization.

### 2.1. PCC Process Simulation Model

The steady-state simulation model of the PCC process used in this study is shown in Figure 2. There are two main process operations in the PCC. One is the absorption chemical process with the solvent, and the other one is the desorption of  $\text{CO}_2$  from the loaded solvent and regeneration of the lean solvent. The flue gas stream is absorbed by the lean amine stream in the absorber column. The depleted flue gas, which does not contain  $\text{CO}_2$ , is then released from the top of the column into the atmosphere and the  $\text{CO}_2$ -loaded MEA stream (shown as the rich MEA stream in Figure 2) leaves the absorber from the bottom, followed by a heat exchanger and a stripper column. In the stripper, the stripping vapor goes up

and condenses at the cooler and condenser on the top. The condensate containing the regenerated solvent is recycled back to the stripper. The CO<sub>2</sub> separated (shown as CO<sub>2</sub> for compression in Figure 2) is sent to the compressor and further processed for transportation or storage. Lean amine is recycled back to the absorber. This lean amine stream heats the CO<sub>2</sub>-loaded MEA stream from the absorber in the heat exchanger.



**Figure 2.** Schematic of chemical absorption process representing postcombustion CO<sub>2</sub> capture (PCC).

Flue gas mass flow rate, lean amine mass flow rate, CO<sub>2</sub> molar fraction in flue gas, lean amine loading, and rich amine loading are initially selected as independent variables. The control variable method is used for data generation. Each variable is selected as the control variable, and its value is changed in a stepwise manner while the other variables are held constant. The capture rate and specific reboiler duty of simulated cases are recorded and collected for further analysis and application. A base case, the main parameters of which are calibrated with industrial data, are listed in Tables 1 and 2 [29]. Here, lean amine loading is referred to the mole ratio of acid gas to the solvent from the bottom of the stripping column which is lean in acid gas. Rich amine loading is referred to the mole ratio of acid gas to the solvent from the bottom of the absorber column which is rich in acid gas. The reboiler duty is the amount of heat required to regenerate solvent.

**Table 1.** Parameters in base case.

Stream	Mass Flow Rate (kg/s)	Temperature (K)	Pressure (bar)
Lean amine	0.642	313.7	1.703
Flue gas	0.158	332.4	1.033

**Table 2.** Base case values.

Stream	Value
Lean amine loading (-)	0.281
Rich amine loading (-)	0.488
Capture rate (-)	0.763
Reboiler duty (MJ/kg)	5.958

The main components in the lean amine stream are H<sub>2</sub>O, CO<sub>2</sub> and MEA. In the flue gas stream, the main compositions are H<sub>2</sub>O, CO<sub>2</sub> and N<sub>2</sub>. Mass fractions of lean amine stream and molar fraction of compositions in the flue gas are listed in Table 3. A script in Python 3.8 connected to the process simulation model in the Aspen HYSYS<sup>®</sup> file is specifically written for the implementation of changing values of controlled variables. The step size, upper and

lower limits, and number of collected data samples for each variable are listed in Table 4. In total, 4393 cases are simulated to generate the raw data by controlling and changing values of different variables. In the next subsection, the machine learning techniques employed to develop data-driven PCC process models are discussed.

**Table 3.** Stream composition in base case.

Compositions	Lean Amine Mass Fraction	Flue Gas Molar Fraction
H <sub>2</sub> O	0.633	0.033
CO <sub>2</sub>	0.062	0.167
MEA	0.305	-
N <sub>2</sub>	-	0.801

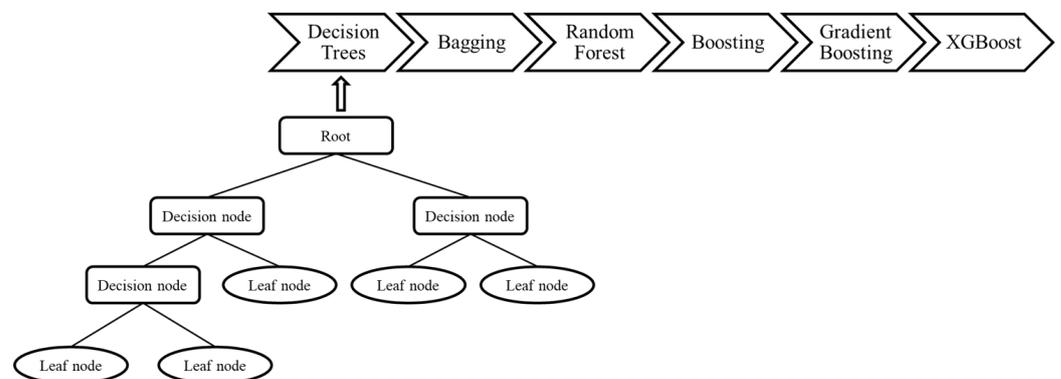
**Table 4.** Data collection information.

	Unit	Lower Limit	Upper Limit	Step Size	Total Samples
Flue gas mass flow rate	kg/s	0.108	0.262	0.0001	1543
Lean amine mass flow rate	kg/s	0.452	2.004	0.01	1553
CO <sub>2</sub> molar fraction in flue gas	-	0.167	0.226	0.0004	150
Lean loading	-	0.028	0.524	0.0005	1147

## 2.2. Machine Learning Techniques

### 2.2.1. Extreme Gradient Boosting

XGBoost, also known as extreme gradient boosting, was launched in 2016. It is a decision tree-based ensemble machine learning algorithm that uses a gradient boosting framework [30]. The evolution route of the XGBoost algorithm is shown in Figure 3, and the instruction starts from the decision tree.



**Figure 3.** Schematic of evolution from decision tree to XGBoost.

Decision tree is a nonparametric supervised learning method used for classification and regression. A decision tree usually consists of three main parts: the root node, decision nodes, and leaf nodes. The root node is a point at which to start the model. The decision node is a judgment node, a conditional judgement; for example, “lean amine loading is above 0.3 or not”, and “flue gas flow rate is larger than 0.38 kg/s or smaller than 0.25 kg/s” exist in a decision node. The different answer to the judgment node leads to different nodes in the next level. The calculation is performed to cross-decision nodes until it reaches the leaf node, the last level of the decision tree. The leaf node is the result of the calculation.

Bagging, short for “bootstrap aggregating”, is an ensemble learning method. In this method, a certain number of samples are randomly picked from the sample pool with replacement. After multiple selections, (for regression problems) the average of the multiple results, (for classification problems) or the result most voted, is the final result. A sample

can be chosen more than one time in the bagging method. Bagging can effectively reduce variance in noisy datasets.

Random forest is a bagging-based algorithm. As known from previous decision tree introduction, characteristics are chosen to be the judgement base in the decision node. In random forest, bagging is used to choose characteristics to form decision trees. All these trees come together to be the “forest”, and the final result comes from the average or the most voted of all tree results.

Boosting is a family of algorithms that convert weak learners into strong learners. At first, training data are used to build a basic learner, and then the wrong prediction samples are focused. The errors are corrected and used to build a stronger learner. This “correct” step is repeated until a model emerges that has an error within acceptable range.

Gradient boosting, which is also called gradient boosting or gradient tree boosting, was developed by Friedman [31]. Three basic elements, a loss function to be optimized, a weak learner to make predictions, and an additive model to add weak learners to minimize the loss function, are included in the gradient-boosting algorithm. Squared error is used as loss function for regression problem and logarithmic loss is used for classification problems. In gradient boosting, the decision tree is the weak learner. A decision tree could be constrained in different ways, including maximum number of layers, nodes, and leaf nodes. Each time, a new decision tree is added for minimizing the loss function and the existing trees remain unchanged.

XGBoost is a decision-tree-based ensemble machine learning algorithm. It takes the bootstrap sample 1 to build model 1, then takes the bootstrap sample 2 to build model 2, which leads a smaller regularized objective function than model 1. Then it iterates until the final model and result are produced. The regularized objective function is

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \quad (1)$$

where  $\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2$ .  $L(\phi)$  is the objective in minimization problem. It is the summation of convex loss function represented as  $\sum_i l(\hat{y}_i, y_i)$  and regularization item denoted as  $\sum_k \Omega(f_k)$ . Here,  $y_i$  and  $\hat{y}_i$  stand for targeted value and predicted value, respectively.  $\lambda$  and  $\gamma$  are hyperparameter constants.  $T$  is the number of leaf nodes.  $w$  is the predicted value of the leaf node. The data from the PCC process simulation is used in this machine learning method. The tasks of predicting capture rate and SRD are regression problems solvable by the XGBoost algorithm.

### 2.2.2. Support Vector Regression

Support vector machine (SVM) is commonly used for solving classification problems. In a few regression applications, SVR is used. In an SVR model, a line or a hyperplane (when the data has higher dimensions) is searched to fit the data. The users have the freedom to decide the acceptable error. In a two-dimensional space, assume the line to fit the data is

$$|Y_i - \alpha X_i| \leq \varepsilon. \quad (2)$$

For a classical linear regression, the objective is to minimize the squared error of original values and predicted values. In the SVR model, the objective (f) is to minimize the coefficients, the  $l_2$ -norm of the coefficient vector, which is shown as follows [32]:

$$f = \text{MIN} \left( \frac{1}{2} \|\alpha\|^2 + \sum_{i=1}^n C |\zeta_i| \right). \quad (3)$$

Here,  $Y_i$  stands for the targeted value,  $X_i$  represents the feature value, and  $\alpha$  denotes coefficients.  $\zeta$  is the differences between  $y$  value of outliers to the closest hyperplane. In SVR models, kernel functions are used to transform input data to the required form of processing data. There are different kinds of kernels: the Gaussian kernel, the radial basis

function, the sigmoid kernel, and the polynomial kernel. The advantages of SVR is that its computational complexity does not depend on the dimensionality of the input space. Moreover, it has excellent generalization capability with high prediction accuracy [33]. The SVR algorithm is an alternative way to build models for predicting CR and SRD in the PCC process.

### 2.2.3. Model Performance Assessment

Three indicators,  $R^2$ , MSE, and cross-validation score, are used to evaluate the performance of data-driven models:

#### A. Coefficient of determination ( $R^2$ )

R-squared ( $R^2$ ) is a statistical measure that represents how much the variance for a dependent variable can be explained by independent variables in a regression model. In general, a higher  $R^2$  indicates a better regression effect. The formula of  $R^2$  is

$$R^2 = 1 - \frac{SSR}{SST} = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}. \quad (4)$$

$SSR$  represents the sum squared regression,  $SST$  represents the total sum of squares. In Equations (4) and (5),  $y_i$  and  $\hat{y}_i$  have the same meaning as in Equation (1). The  $y_i$  is real value,  $\hat{y}_i$  is the predicted value,  $\bar{y}$  is the average of the actual  $y$ , and  $q$  is the number of observations.

#### B. Mean Square Error

MSE is the average squared distance between the actual and predicted values. The formula is

$$MSE = \frac{\sum (y_t - \hat{y}_t)^2}{q}. \quad (5)$$

#### C. Cross-validation score

Cross-validation is a technique for detecting overfitting or assessing the generalized ability of the model. Overfitting means a model fits well and has high accuracy with the training data, while it has low accuracy against the test or unseen data. Generalization refers to a model's ability to adapt properly to new or unseen data that are drawn from the same distribution as the one used to create the model. A model that is overfitting has a low generalized ability. When cross-validation is used, datasets are separated into several folds evenly. At each turn, one fold is selected as a test fold whereas the others are training folds. This method is useful as it gives the algorithm different subsets of data to train on. In this study, the cross-validation score is the average value of  $R^2$  of all testing folds. In the next subsection, optimization techniques associated with machine learning incorporation are explained. Specifically, the agent-based and stochastic optimization methods used in this study are illustrated.

## 2.3. Optimization Algorithms

### 2.3.1. Particle Swarm Optimization

PSO is an agent-based search optimization technique inspired by the migration behavior of birds proposed by Kennedy et al. [34]. Assume there are  $N$  particles subject to random initialization in a swarm. These particles have random positions and velocities in a  $D$ -dimensional search space, and they move at a certain speed toward the objective location inside the whole space. For each particle, the new velocity is updated based on its own historical experience and the group general experience. Assume the  $D$ -dimensional position vector of the  $i$ -th particle is

$$S_i = (S_{i1}, S_{i2}, S_{i3}, \dots, S_{iN}), i = 1, 2, \dots, N. \quad (6)$$

The velocity vector of  $i$ -th particle is

$$V_i = (V_{i1}, V_{i2}, V_{i3}, \dots, V_{iN}), i = 1, 2, \dots, N. \quad (7)$$

The equations used to update the position and velocity of each particle are

$$v_{id}^{k+1} = mv_{id}^k + c_1 r_1 (p_{id}^k - s_{id}^k) + c_2 r_2 (p_{gd}^k - s_{id}^k) \quad (8)$$

$$s_{id}^{k+1} = s_{id}^k + rv_{id}^{k+1}. \quad (9)$$

The equation to update the speed of each particle consists of three parts. The  $v_{id}$  and  $s_{id}$  are the velocity and position of the  $i$ -th particle in the  $d$ -th dimension,  $k$  and  $k + 1$  represent the current and next iterations. The first part,  $mv_{id}^k$ , is the exploration, and  $m$  is an inertia factor of fixed value. The second part  $c_1 r_1 (p_{id}^k - s_{id}^k)$  represents self-learning, and the third part,  $c_2 r_2 (p_{gd}^k - s_{id}^k)$ , represents group learning.  $c_1$  and  $c_2$  are learning factors,  $r_1$  and  $r_2$  are random numbers within the range  $[0, 1]$ .  $p_{id}$  and  $p_{gd}$  are the most satisfied positions searched by the  $i$ -th particle and the whole group. The  $p_{id}$  and  $p_{gd}$  are recorded as

$$P_{best} = (P_{i1}, P_{i2}, P_{i3}, \dots, P_{iN}), i = 1, 2, \dots, N \quad (10)$$

$$G_{best} = (G_{i1}, G_{i2}, G_{i3}, \dots, G_{iN}), i = 1, 2, \dots, N. \quad (11)$$

The PSO is a heuristic algorithm and available for this case to search for the optimized operating conditions in PCC within the XGBoost models.

### 2.3.2. Simulated Annealing

SA is a stochastic optimization algorithm based on Monte Carlo iterative solution strategy introduced by S. Kirkpatrick et al. in 1983 [35]. The main idea of this method is to introduce the annealing in solids into traditional optimization problems. A temperature,  $\tau$ , is set for the control of the whole optimal solution searching process. Assume there is a initial solution  $\psi$  for the goal function  $E(\psi)$ , and random perturbations are added to this solution to generate the new solution. For these two solutions, if the new solution is better than old one, it is accepted. If it is not, it is not to be given up. Instead, Metropolis criteria is used to decide whether the new solution will be accepted or not. In Metropolis criteria, the probability of accepting the new solution is

$$P = \begin{cases} 1 & E(\psi + 1) < E(\psi) \\ \exp(-\frac{E(\psi+1)-E(\psi)}{\tau}) & E(\psi + 1) \geq E(\psi). \end{cases} \quad (12)$$

After deciding to accept the new solution or not, the number of iterations are confirmed. If the end condition is not met, the temperature is adjusted slowly, and a new solution is regenerated. The SA optimization method is versatile as it does not rely on restrictive properties of the models. It is also used for optimization in this study to search for the optimal operating conditions. Note that all the data-driven model development and optimization studies in this paper are carried out by using Python script and subroutines on an Intel Core i7 (Sandy bridge) 2.30 GHz processor. In the next section, the data analysis result, the prediction performance of built models, and the optimization results are shown and discussed.

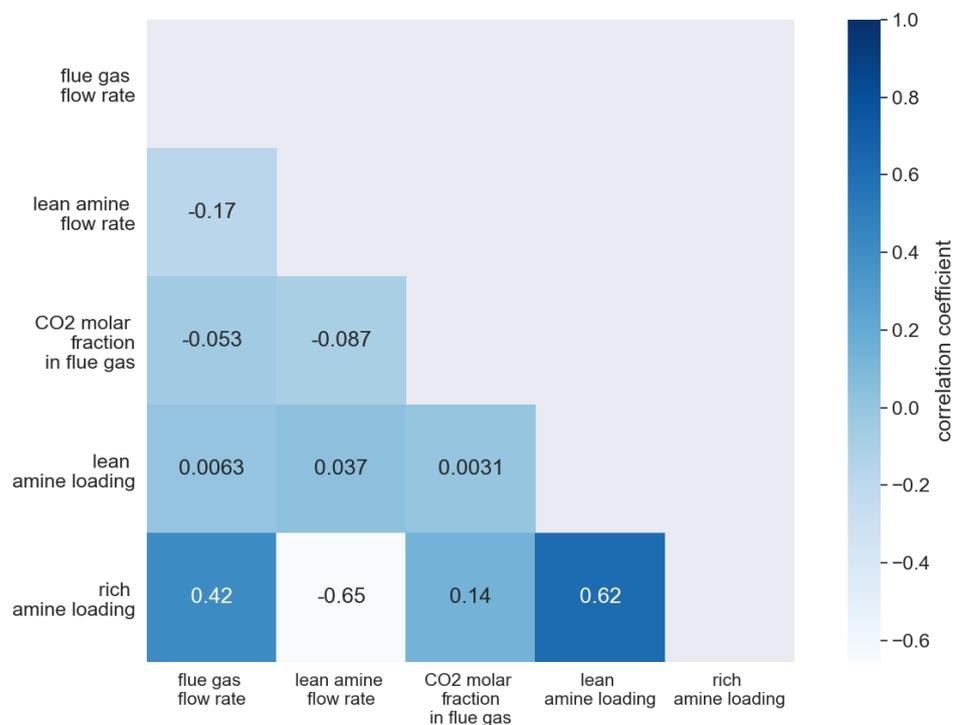
## 3. Results

### 3.1. Data Analysis and Model Development

Data analysis aims at having an overview of the datasets by investigating the relations between selected variables. The Pearson correlation coefficient which measures the linear correlation between variables is used. Given the paired dataset  $(Z_{11}, Z_{21}), (Z_{12}, Z_{22}), \dots, (Z_{1n}, Z_{2n})$ , the formula of Pearson correlation coefficient  $R_{Z_1, Z_2}$  is

$$R_{Z_1, Z_2} = \frac{\sum_{i=1}^n (Z_{1i} - \bar{Z}_1)(Z_{2i} - \bar{Z}_2)}{\sqrt{\sum_{i=1}^n (Z_{1i} - \bar{Z}_1)^2} \sqrt{\sum_{i=1}^n (Z_{2i} - \bar{Z}_2)^2}}, \quad (13)$$

where  $\bar{Z}_1$  and  $\bar{Z}_2$  are the average values of variables  $Z_1$  and  $Z_2$ . Figure 4 shows Pearson correlation coefficients between distinct variables in this study.

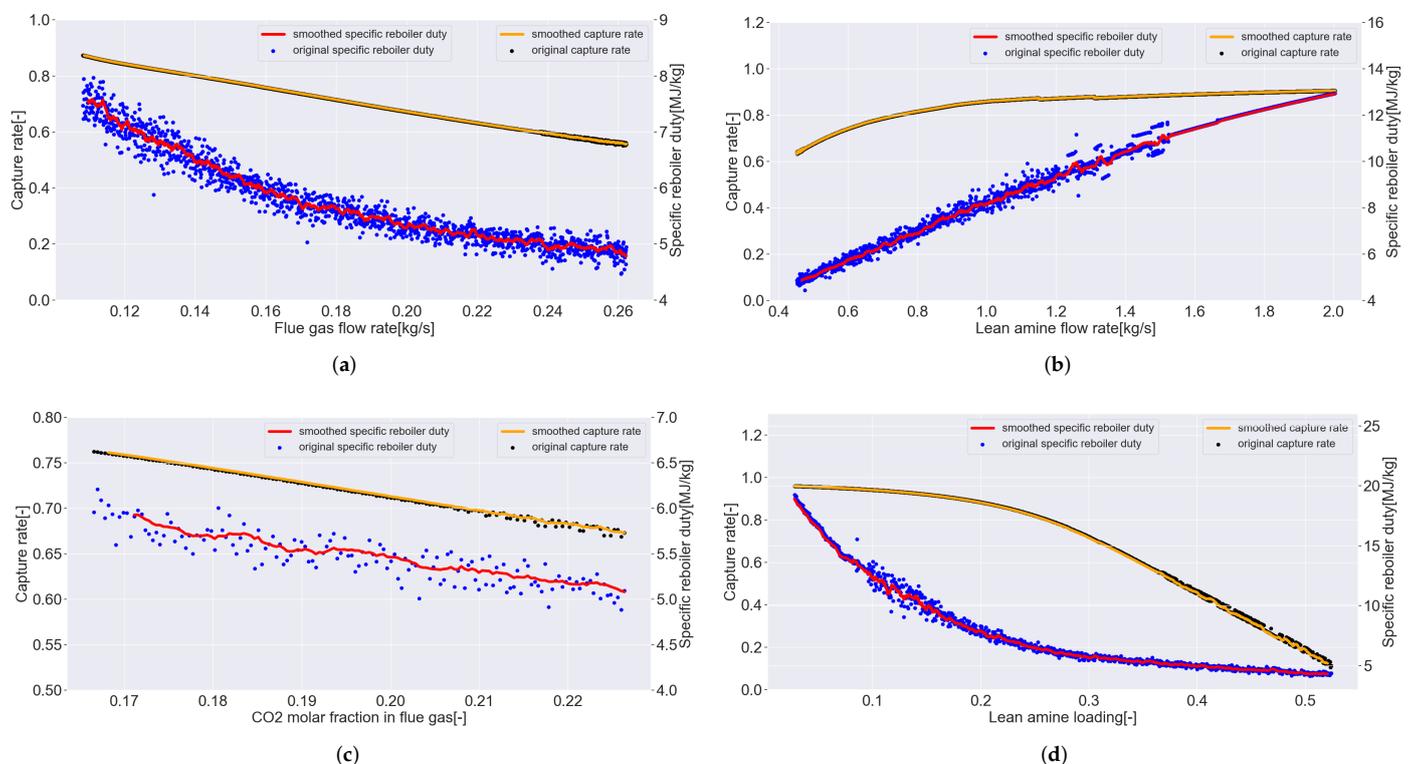


**Figure 4.** Plot of data correlation coefficients.

In Figure 4, the correlation coefficients of different variables considered in this study are shown in the grid format. The grid color is proportional to the correlation coefficient. The color tends to be greener with smaller values, whereas the blue color corresponds to a larger value. For example, the grids intersected by lean amine flow rate and the flue gas flow rate show the correlation coefficient is  $-0.17$ , and the color is light green. This means that these two variables are slightly inverse related. The correlation coefficient between rich amine loading and lean amine loading is  $0.62$ , which shows they have positive linear relationships and that the rich amine loading is influenced by the lean amine loading. Therefore, rich amine loading is not included as an input variable. A general understanding of the datasets is obtained by this Pearson correlation coefficient method. Note that nonlinear relations are not taken into account here.

A detailed analysis showing the relationship between dependent and independent variables is shown in Figure 5. In this section, *Depleted flue gas (To atmosphere)*, *CO<sub>2</sub> for compression*, and *Rich MEA* refer to the gas stream from the absorber, the captured CO<sub>2</sub> stream out from the stripper, and the loaded amine stream out from absorber, as shown in Figure 2, which depicts the simulation model. In Figure 5, the solid line denotes smoothed data and solid dots representing original data obtained from process simulation models. Figure 5a includes the capture rate and the flue gas flow rate of 1543 process simulation cases with a downward trend. This shows that the capture rate and the flue gas flow rate are inversely proportional. The highest point represents a case in which the flue gas flow rate is  $0.108$  kg/s and the capture rate is  $0.874$ ; the lowest one indicates a case in which the flue gas flow rate is  $0.262$  kg/s and the capture rate is  $0.556$ . The CO<sub>2</sub> entering the absorber in a time unit increases with the flue gas mass flow rate. Due to the solvent quantity limitation and increased inflow of CO<sub>2</sub>, the capture rate is lower, and more CO<sub>2</sub> is released into the

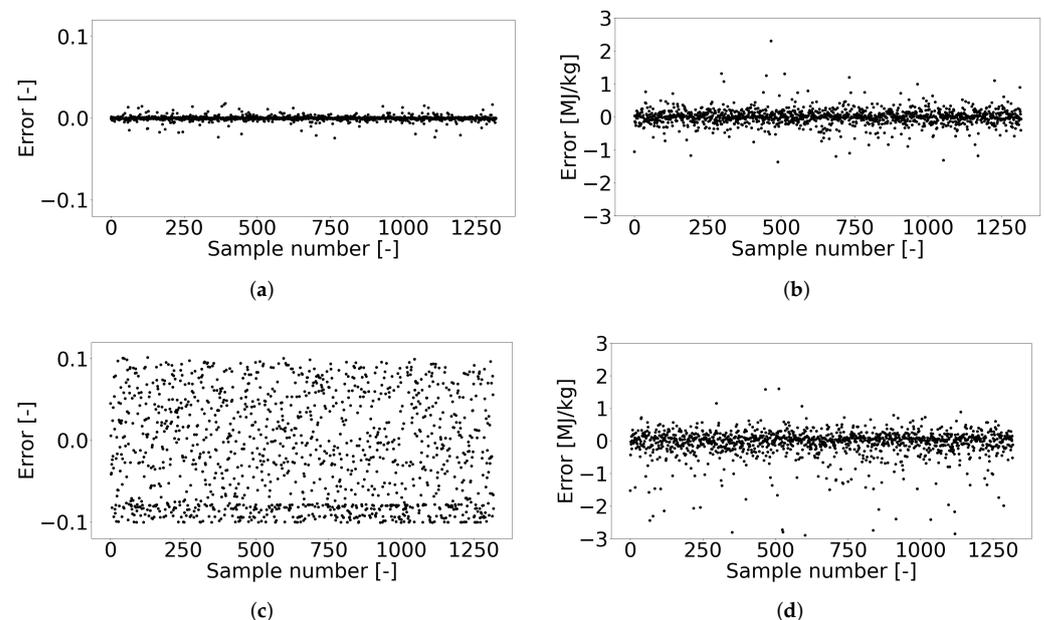
atmosphere through *Depleted flue gas (To atmosphere)* stream. Therefore, the capture rate is smaller. Toward the left side of this plot, where the flue gas flow rate is close to 0.262, the capture rate fluctuates. This could be because of the model instability. In Figure 5a, the relation between the SRD and the flue gas flow rate is inversely proportional. Because SRD is the reboiler duty divided by the mass of CO<sub>2</sub> in CO<sub>2</sub> for compression stream, the SRD decreases as the CO<sub>2</sub> in CO<sub>2</sub> for compression stream goes up and the reboiler duty goes down. In Figure 5b, the capture rate grows with the lean amine flow rate. This is because the CO<sub>2</sub> entering the rich stream increases with the lean amine flow rate. In Figure 5b, the trend of CO<sub>2</sub> in the rich MEA stream is upward due to increase in the lean MEA flow rate and CR. Consequently, the reboiler duty is larger as the stream enters the stripper that contains more MEA. The SRD increases in a similar manner. The rise in CO<sub>2</sub> molar fraction in flue gas results in the increase in CO<sub>2</sub> in the stream that is released into the atmosphere and the *Rich MEA* stream, but the increment in *Depleted flue gas (To atmosphere)* is larger while the increment in *Rich MEA* stream is smaller. This provides an explanation for the lower CR values in this case. The reboiler duty of the stripper is lower, resulting in a lower SRD. Lean amine loading is changed with the mass flow rate of CO<sub>2</sub> in the lean amine. Because the lean amine loading is larger, CO<sub>2</sub> in the *Depleted flue gas (To atmosphere)* stream increases, and the capture rate decreases. Reboiler duty and SRD are lower accordingly. From Figure 5, it is observed that the SRD fluctuates due to the instability in the reboiler duty. In general, data smoothing is a way to remove the fluctuations or disturbances. The data smoothing is implemented via methods such as simple exponential, moving average, exponential moving average, and Holt–Winters smoothing methods. However, the expected improvement is not observed while using the smoothed data to develop the model. For the consideration of keeping original information, the following models are developed by using the raw data without smoothing.



**Figure 5.** Plots of original data collected and smoothed curve for each independent variable and dependent variable. (a) Flue gas flow rate vs. capture rate/specific reboiler duty. (b) Lean amine flow rate vs. capture rate/specific reboiler duty. (c) CO<sub>2</sub> molar fraction in flue gas vs. capture rate/specific reboiler duty. (d) Lean amine loading vs. capture rate/specific reboiler duty.

### 3.2. Validation Results

The 4393 data samples are randomly divided into training datasets and test datasets with a division ratio of 7:3. The data-driven models are built based on training datasets. The predicted values based on independent variables in test datasets are compared with the original values to measure the model performance. The four indicators, cross-validation score, MSE, and  $R^2$ , are used for assessing model performances. The model which is built by XGBoost for predicting capture rate had a cross-validation average score of 0.9995,  $R^2$  of 0.9996, and MSE of 0.0000. Figure 6a depicts the plot of the model results obtained by employing the XGBoost machine learning algorithm. In particular, the difference between predicted CR and tested CR in terms of the error represented by black solid dots is shown. Most points are distributed around the horizontal line  $y = 0$ . This means that the predicted values are close to the original values, which indicates the model has an accurate prediction. This aligns well with the high  $R^2$  of 0.9996. One possibility to lead to the high  $R^2$  is the overfitting. To validate this accuracy, cross-validation is performed to check the existence of the overfitting. In addition, the SVR algorithm is implemented to develop the other model for comparison purposes. The SVR models show accurate/similar prediction performance when compared with XGBoost model. If the SVR models show different prediction performances, then it can be concluded that the XGBoost models show better prediction abilities. Consequently, four models are built to predict capture rate and SRD individually by employing XGBoost and SVR algorithms, respectively. The closer the dots are to the  $y = 0$ , the better performance the model has. The qualities of both XGBoost and SVR models are observed in Figure 6. In particular, the XGBoost model shows better performance in predicting CR and SRD when compared with the SVR model as the error plot is less scattered. When using the XGBoost to predict the capture rate, the model has the highest prediction accuracy among the four models. The model using the SVR algorithm to predict the capture rate has the most scattered pattern, which illustrates the lowest accuracy. In Table 5, the performance of distinct models are summarized.



**Figure 6.** Plots of errors for model prediction. (a) Error of capture rate prediction (XGBoost). (b) Error of SRD prediction (XGBoost). (c) Error of capture rate prediction (SVR). (d) Error of SRD prediction (SVR). All errors for model prediction are based on original data.

**Table 5.** Model assessment in terms of model validation and verification. The cross-validation score is the average value of  $R^2$  of all testing folds. Validation  $R^2$  and Validation MSE are the values during the model validation. After validation, 78 cases different from training and test datasets are used for verification. The verification  $R^2$  and verification MSE are the results of model performance assessment during verification.

Assessment	XGBoost Predict CR	XGBoost Predict SRD	SVR Predict CR	SVR Predict SRD
cross-validation score	0.9995	0.9934	0.8573	0.9620
Validation $R^2$	0.9996	0.9930	0.8418	0.9622
Validation MSE	0.0000	0.0595	0.0041	0.3231
Verification $R^2$	0.9170	0.8077	0.8793	0.8716
Verification MSE	0.0029	0.4043	0.0042	0.2700

### 3.3. Verification Results

The verification of the developed models is performed by simulating 78 cases with the consideration of variables within the same range, as shown in Table 4 during process simulation. These cases are different from the cases in training datasets and test datasets. This verification dataset can include more data samples; however, the number of data samples is restricted due to the operational range where the process model is developed. The verification results of models are shown in Table 5.

It is assumed that the models are available when the independent variables are within certain ranges shown in Table 4. The availability of developed models in all cases or outside the range is a subject investigation. From Table 5, it is seen that the XGBoost models show  $R^2$  of 91.70% in predicting capture rate and 80.77% in predicting SRD, whereas the  $R^2$  of SVR models are 87.93% in predicting capture rate and 87.16% in predicting SRD. The model results are discussed as follows.

(1) In general, data-driven modeling is implemented via data collection, data analysis, variables selection, modeling and verification. It is noted that data collection and data analysis are steps that precede the variable selection. In this order, the related variables are kept, and unrelated variables are discarded. However, in this study, the order is slightly different due to data availability. The data are not preprovided and are generated by using process simulation. Hence, it is unrealistic to produce datasets that include all variables in process simulation models. Thus, the variable selection is conducted first for ensuring the working efficiency. The variables are chosen mainly based on previous research and industrial experience.

(2) Figure 5b–d have fluctuations, and Figure 5c has breakpoints. This is due to the instability of the process simulation models. The variables in these first-principal models include multiple differential and algebraic equations. Therefore, the same inputs lead to outputs with slight differences due to steady-state multiplicity.

(3) The amount of the original data samples is more than 4000, only 70% of the data samples are used to train the model and the step size between different samples are small, which may not differentiate much. Therefore, the model still has room for improvement if more data samples are included in training datasets.

(4) In many applications, the XGBoost algorithm has shown high accuracy of predictions. In this study, training and test datasets are synthetic data generated by the steady-state simulation. Therefore, these data have certain mathematic expressions because they come from the same simulation process model. Consequently, it is possible that the  $R^2$  of the model can reach 99%. The other reasons for this high accuracy are the complexity of the prediction task, the number of training datasets, and the possibility of overfitting, which has mentioned before.

### 3.4. Optimization Results

In this section, the optimization results of the PCC process simulation are explained. In general, the SRD increases with the capture rate. However, the lower SRD and higher capture rate are favorable. Therefore, three cases are investigated corresponding to the maximum capture rate, the lowest SRD, and the one with a tradeoff between capture rate and SRD respectively. The objective functions are shown in Equations (14)–(16). We have

$$f(x) = CR(x) \quad (14)$$

$$f(x) = SRD(x) \quad (15)$$

$$f(x) = \frac{SRD(x)}{SRD_{upperlimit}} - \frac{CR(x)}{CR_{upperlimit}}, \quad (16)$$

where the  $SRD(x)$  and  $CR(x)$  are the SRD and CR predicted by the XGBoost model based on the vector  $x$ , which represents the operating conditions. In the tradeoff case, the capture rate and the SRD are scaled to have a similar unit as they are not within the same ranges. The implementation of the agent-based and stochastic optimization involve many iterations. At each iteration, the values of different variables obtained by running process simulation are desired. It takes the process simulation model 3.15 s to predict the CR and SVR for 1 case. On the other hand, the data-driven models developed by using the XGBoost and SVR algorithms require 0.022 s and 0.217 s for predicting the CR and SVD for 1318 cases. The reduction of the prediction time has a positive effect on the optimization. During the application of the PSO algorithm, a swarm of 10 particles and 1500 number of iterations are considered. The operating conditions for the three specific objectives mentioned above are searched. The maximum capture rate reaches 95.98%, and the lowest SRD is 3.720 MJ/kg. The tradeoff case shows the minimal value of fitness in Equation (16) as  $-0.530$ . The detailed results are shown in Table 6. For the tradeoff case, the CR is lower than the CR of 76.3% in the base case and the SRD is 1.7 MJ/kg less than the 5.9 MJ/kg in the base case with the decrease of 28.2%.

**Table 6.** Optimized operating conditions for PSO algorithm implementation. Case 1: Find the operating conditions that resulted in the largest CR without leveraging the value of SRD. Case 2: Find the operating conditions corresponding to the smallest SRD without leveraging the value of CR. Case 3: A trade-off case, it aims to find a relatively high CR and low SRD.

Operating Conditions	Case 1: Largest CR	Case 2: Smallest SRD	Case 3: Trade-Off Case
flue gas flow rate	0.158	0.26	0.18
lean amine flow rate	0.606	0.46	0.46
CO <sub>2</sub> molar fraction	0.167	0.21	0.200
lean amine loading	0.028	0.45	0.309
CR	0.960	0.550	0.722
SRD	19.029	3.720	4.3

The results obtained from the SA algorithm implementation are different from the PSO algorithm. For the SA algorithm, the initial temperature  $\tau$  of 1000 °C, 300 iterations are considered. The searched operating points are depicted in Table 7. The largest capture rate is 94.87%, smaller than the 95.98% in the PSO algorithm. The SA method is better in searching the smallest SRD operating point. The operating point found by SA is 3.624 MJ/kg, is smaller than the 3.720 MJ/kg in the PSO algorithm. In the tradeoff case, a CR of 70.08% and SRD of 4.079 MJ/kg are obtained by employing SA. Compared with the base case, which has a CR of 76.3% and an SRD of 5.9 MJ/kg, although the CR was 6.22% lower, the SRD decreased by 1.83 MJ/kg, which accounts for 31.02% of 5.9 MJ/kg. The fitness  $f(x)$  in Equation (16) with this operating point is  $-0.518$ . As this value is larger than  $-0.530$  in the PSO algorithm, the tradeoff operating points in the PSO performs better.

**Table 7.** Optimized operating conditions for SA algorithm implementation. Case 1: Find the operating conditions that resulted in the largest CR without leveraging the value of SRD. Case 2: Find the operating conditions corresponding to the smallest SRD without leveraging the value of CR. Case 3: A trade-off case, aims to find a relatively high CR and low SRD.

Operating Conditions	Case 1: Largest CR	Case 2: Smallest SRD	Case 3: Trade-Off Case
flue gas flow rate	0.143	0.261	0.163
lean amine flow rate	1.914	0.460	0.474
CO <sub>2</sub> molar fraction	0.166	0.218	0.201
lean amine loading	0.079	0.409	0.339
CR	0.949	0.550	0.701
SRD	13.540	3.624	4.079

The operating points with the largest CR found by PSO and SA are quite different, whereas those with the smallest SRD and tradeoff case are similar. For the operating conditions corresponding to the lowest SRD, despite the other conditions being similar, lean amine loading reduces the value from 0.45 to 0.409, and the SRD reduces from 3.720 MJ/kg to 3.624 MJ/kg. This may not align with the trend shown in Figure 5c. These two operating points happen to be in the fluctuation part, where the decrease of lean amine loading leads to a lower SRD. This is possible in the case that the change of lean amine loading is small. The reason for this trend reversal is due to the three other parameters (flue gas flow rate, lean amine flow rate, CO<sub>2</sub> molar fraction in flue gas) in these operating cases being similar rather than exactly the same. The change of the other three parameters may also influence the SRD. The tradeoff cases searched by PSO and SA are similar, the differences between values are less than 0.05, for example, and the flue gas flow rate is 0.18 kg/s and 0.162 kg/s, less than 0.02 kg/s. By combining these two algorithms, it is seen that the largest capture rate is 95.98%, and the lowest SRD is 3.624 MJ/kg. The tradeoff point, which has a balance between the larger capture rate and the lower SRD, is with flue gas flow rate 0.18 kg/s, lean amine flow rate 0.46 kg/s, CO<sub>2</sub> molar fraction 0.201, and lean amine loading 0.309. The corresponding capture rate is 72.2% and specific reboiler duty is 4.3 MJ/kg. In the next section, the conclusions of this paper are presented.

#### 4. Conclusions

In this paper, the application of machine learning techniques such as XGBoost and SVR for the PCC process model simulation was demonstrated successfully. First, the energy efficiency indicators and parameters associated with the PCC process model were identified. Machine learning algorithms were then applied to build models to predict CR and SRD. The models were used in optimization routines, and adequate operating conditions are characterized. The data-driven models showed high accuracy in predicting the capture rate and energy requirement in the reboiler of the PCC model. The XGBoost model had the accuracy of 99.9% and 99.3% for predicting CR and SRD based on the validation datasets. The SVR model showed 84.1% and 96.2% in CR and SRD prediction, respectively. The computation time for XGBoost and SVR models was approximately 90% lower when compared with the first-principle-based process model. Thus, the data-driven models showed improved performance in the time-efficient aspect. The goal of developing time-efficient models by machine learning techniques was achieved. The data-driven model were incorporated in optimization methods such as PSO and SA. The largest capture rate reached was found to be 96.0%, and the corresponding specific reboiler duty was 19.0 MJ/kg. The lowest specific reboiler duty was 3.6 MJ/kg; however, it had a low capture rate of 55.0%. The ideal operating condition was flue gas flow rate as 0.18 kg/s, lean amine flow rate as 0.46 kg/s, CO<sub>2</sub> molar fraction in flue gas as 0.20, and lean amine loading as 0.31. The corresponding CR and SRD are 72.2% and 4.3 MJ/kg, respectively. In this case, the CR decreases 4.1% lower and SRD dwindles 1.7 MJ/kg compared with the base case. Thus, the application of machine learning techniques was demonstrated as useful in process

optimization considering PCC process simulation. The advantage of fast model predictions by using data-driven techniques can be extended in the application of advanced online control and optimization methods. Processes with fast dynamics, in which the control makes calculations within a second, are required to benefit from the proposed method.

**Author Contributions:** H.Z. performed all simulations and wrote the first draft of the paper. G.M. helped with the writing, reviewing, editing, analysis, conceptualization, supervision and verification of the technical aspects of the research work. L.O.N. helped with the paper writing, funding acquisition and oversaw all the aspects of the research work. All authors have read and agreed to the published version of the manuscript.

**Funding:** This publication has been funded by the CleanExport project—Planning Clean Energy Export from Norway to Europe, Project no. 308811. The authors gratefully acknowledge the financial support from the Research Council of Norway and the user partners Agder Energi, Air Liquide, Equinor Energy, Gassco, and Total E&P Norge.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All datasets used in this work are generated by simulation software and available from the corresponding author on request.

**Acknowledgments:** The authors acknowledge the Department of Energy and Process Engineering at NTNU for facilitating with computer resources and software package. The authors would also like to thank NTNU Ph.D. candidate Valentin Formont for help during the work.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Nomenclature and Abbreviations

The following nomenclature and abbreviations are used in this manuscript:

XGBoost	eXtreme Gradient Boosting
SVR	Support Vector Regression
CR	Capture Rate
SRD	Specific Reboiler Duty
R <sup>2</sup>	Coefficient of determination
WMO	World Meteorological Organization
GHG	Greenhouse Gas Emissions
CCS	Carbon capture and storage
PCC	Post-combustion capture
MEA	Monoethanolamine
DEA	Di-EthanolAmine
AMP	2-amino-2-methyl-1-propanol
PZ	Piperazine
IL	Ionic Liquid
ANN	Artificial Neural Network
LSSVM	Least Squares Support Vector Machine
GPR	Gaussian Process Regression
BPMLP	Back Propagation Multi-Layer Perceptron
RBF	Radial Basis Function
ELM	Extreme Learning Machine
MSE	Mean squared error
ARX	Auto-Regressive model with eXogenous inputs
BIOCS	Biologically Inspired Optimal Control Strategy

PSO	Particle Swarm Optimization
GEP	Generation Expansion Planning
GA	Genetic Algorithm
ABC	Artificial Bee Colony
RBFNN	Radial Basis Function Neural Network
SA	Simulated Annealing
GA-ANFIS	Genetic Algorithm-Adaptive Neuro Fuzzy Inference System
PSO-ANFIS	Particle Swarm Optimization ANFIS
CSA-LSSVM	Coupled Simulated Annealing-Least Squares Support Vector Machine
SVM	Support Vector Machine
SST	The total sum of squares
SSR	The sum squared regress
MSE	Mean squared error
SA	Simulated Annealing

## References

- World Meteorological Organization. 2020 Was One of Three Warmest Years on Record. 2021. Available online: <https://public.wmo.int/en/media/press-release/2020-was-one-of-three-warmest-years-record> (accessed on 13 May 2022).
- Boot-Handford, M.E.; Abanades, J.C.; Anthony, E.J.; Blunt, M.J.; Brandani, S.; Dowell, N.M.; Fernández, J.R.; Ferrari, M.-C.; Gross, R.; Hallett, J.P.; et al. Carbon capture and storage update. *Energy Environ. Sci.* **2014**, *7*, 130–189. [[CrossRef](#)]
- Rubin, E.S.; Mantripragada, H.; Marks, A.; Versteeg, P.; Kitchin, J. The outlook for improved carbon capture technology. *Prog. Energy Combust. Sci.* **2012**, *38*, 630–671. [[CrossRef](#)]
- Svendsen, H.F.; Hessen, E.T.; Mejdell, T. Carbon dioxide capture by absorption, challenges and possibilities. *Chem. Eng. J.* **2011**, *171*, 718–724. [[CrossRef](#)]
- Aspen HYSYS® (Version 10) [Software]. 2022. Available online: <https://www.aspentech.com/en> (accessed on 22 August 2022).
- Aspen Plus® (Version 12) [Software]. 2022. Available online: <https://www.aspentech.com/en> (accessed on 22 August 2022).
- gPROMS® (Version 2) [Software]. 2022. Available online: <https://www.psenterprise.com/products/gproms> (accessed on 22 August 2022).
- Co2sim® [Software]. 2022. Available online: <https://www.sintef.no/en/software/co2sim-flowsheet-simulator-for-co2-absorption-proc/> (accessed on 22 August 2022).
- Øi, L.E. Comparison of aspen hysys and aspen plus simulation of CO<sub>2</sub> absorption into mea from atmospheric gas. *Energy Procedia* **2012**, *23*, 360–369. [[CrossRef](#)]
- van der Spek, M.; Arendsen, R.; Ramirez, A.; Faaij, A. Model development and process simulation of postcombustion carbon capture technology with aqueous amp/pz solvent. *Int. J. Greenh. Gas Control* **2016**, *47*, 176–199. [[CrossRef](#)]
- Luo, X.; Wang, M. Improving prediction accuracy of a rate-based model of an MEA-based carbon capture process for large-scale commercial deployment. *Engineering* **2017**, *3*, 232–243. [[CrossRef](#)]
- Dutta, R.; Nord, L.O.; Bolland, O. Applicability and validation of use of equilibrium-based absorber models with reduced stage efficiency for dynamic simulation of post-combustion CO<sub>2</sub> capture processes. *Energy Procedia* **2017**, *114*, 1424–1433. [[CrossRef](#)]
- Fernandez, M.; Boyd, P.G.; Daff, T.D.; Aghaji, M.Z.; Woo, T.K. Rapid and accurate machine learning recognition of high performing metal organic frameworks for CO<sub>2</sub> capture. *J. Phys. Chem. Lett.* **2014**, *5*, 3056–3060. [[CrossRef](#)]
- Venkatraman, V.; Alsberg, B.K. Predicting CO<sub>2</sub> capture of ionic liquids using machine learning. *J. Co<sub>2</sub> Util.* **2017**, *21*, 162–168. [[CrossRef](#)]
- Ahmadi, M.; Kashiwao, T.; Rozyn, J.; Bahadori, A. Accurate prediction of properties of carbon dioxide for carbon capture and sequestration operations. *Pet. Sci. Technol.* **2016**, *34*, 97–103. [[CrossRef](#)]
- Wu, X.; Shen, J.; Li, Y.; Wang, M.; Lawal, A. Flexible operation of post-combustion solvent-based carbon capture for coal-fired power plants using multi-model predictive control: A simulation study. *Fuel* **2018**, *220*, 931–941. [[CrossRef](#)]
- Rahimi, M.; Moosavi, S.M.; Smit, B.; Hatton, T.A. Toward smart carbon capture with machine learning. *Cell Rep. Phys. Sci.* **2021**, *2*, 100396. [[CrossRef](#)]
- Shalaby, A.; Elkamel, A.; Douglas, P.L.; Zhu, Q.; Zheng, Q.P. A machine learning approach for modeling and optimization of a CO<sub>2</sub> post-combustion capture unit. *Energy* **2021**, *215*, 119113. [[CrossRef](#)]
- Shahsavand, A.; Fard, F.D.; Sotoudeh, F. Application of artificial neural networks for simulation of experimental CO<sub>2</sub> absorption data in a packed column. *J. Nat. Gas Sci. Eng.* **2011**, *3*, 518–529. [[CrossRef](#)]
- Chan, V.; Chan, C. Learning from a carbon dioxide capture system dataset: Application of the piecewise neural network algorithm. *Petroleum* **2017**, *3*, 56–67. [[CrossRef](#)]
- Li, F.; Zhang, J.; Oko, E.; Wang, M. Modelling of a post-combustion CO<sub>2</sub> capture process using extreme learning machine. *Int. J. Coal Sci. Technol.* **2017**, *4*, 33–40. [[CrossRef](#)]
- Mirlekar, G.; Gebreslassieb, B.; Diwekar, U.; Lima, F.V. Biomimetic model-based advanced control strategy integrated with multi-agent optimization for nonlinear chemical processes. *Chem. Eng. Res. Des.* **2018**, *140*, 229–240. [[CrossRef](#)]

23. Mirlekar, G.; Al-Sinbol, G.; Perhinschi, M.; Lima, F.V. A biologically-inspired approach for adaptive control of advanced energy systems. *Comput. Chem. Eng.* **2018**, *117*, 378–390. [[CrossRef](#)]
24. Saboori, H.; Hemmati, R. Considering carbon capture and storage in electricity generation expansion planning. *IEEE Trans. Sustain. Energy* **2016**, *7*, 1371–1378. [[CrossRef](#)]
25. Menad, N.A.; Hemmati-Sarapardeh, A.; Varamesh, A.; Shamshirband, S. Predicting solubility of CO<sub>2</sub> in brine by advanced machine learning systems: Application to carbon capture and sequestration. *J. CO<sub>2</sub> Util.* **2019**, *33*, 83–95. [[CrossRef](#)]
26. Van Laarhoven, P.J.; Aarts, E.H. Simulated annealing. In *Simulated Annealing: Theory and Applications*; Springer: Dordrecht, The Netherlands, 1987; pp. 7–15. [[CrossRef](#)]
27. Zhao, B.; Su, Y.; Tao, W. Mass transfer performance of CO<sub>2</sub> capture in rotating packed bed: Dimensionless modeling and intelligent prediction. *Appl. Energy* **2014**, *136*, 132–142. [[CrossRef](#)]
28. Dashti, A.; Raji, M.; Alivand, M.S.; Mohammadi, A.H. Estimation of CO<sub>2</sub> equilibrium absorption in aqueous solutions of commonly used amines using different computational schemes. *Fuel* **2020**, *264*, 116616. [[CrossRef](#)]
29. Dutta, R.; Nord, L.O.; Bolland, O. Prospects of using equilibrium-based column models in dynamic process simulation of post-combustion CO<sub>2</sub> capture for coal-fired power plant. *Fuel* **2017**, *202*, 85–97. [[CrossRef](#)]
30. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [[CrossRef](#)]
31. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]
32. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [[CrossRef](#)]
33. Awad, M.; Khanna, R. Support vector regression. In *Efficient Learning Machines*; Apress: Berkeley, CA, USA, 2015; pp. 67–80. [[CrossRef](#)]
34. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [[CrossRef](#)]
35. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)]