

EVALUATING THE EFFECT OF BLENDING RATIO ON THE CO-GASIFICATION OF HIGH ASH COAL AND BIOMASS IN A FLUIDIZED BED GASIFIER USING MACHINE LEARNING

Furkan ELMAZ, Electrical and Electronics Engineering, İzmir Kâtip Çelebi University, Turkey, furkanelmaz@icloud.com

([ID](https://orcid.org/0000-0002-7030-0784) <https://orcid.org/0000-0002-7030-0784>)

Özgün YÜCEL, Environmental Engineering, İzmir Kâtip Çelebi University, Turkey, ozgun.yucel@ikc.edu.tr

([ID](https://orcid.org/0000-0001-8916-2628) <https://orcid.org/0000-0001-8916-2628>)

*Ali Yener MUTLU, Electrical and Electronics Engineering, İzmir Kâtip Çelebi University, Turkey, aliyener.mutlu@ikc.edu.tr

([ID](https://orcid.org/0000-0002-2221-8698) <https://orcid.org/0000-0002-2221-8698>)

Received: 17.10.2018, Accepted: 28.12.2018

Research Article

*Corresponding author

DOI: 10.22531/muglajsci.471538

Abstract

Co-gasification is a process that converts coal and biomass into useful products, such as syngas. Analytical and numerical approaches for modeling co-gasification process either require enormous amount of time or make a lot of assumptions which reduce consistency of the models in practical applications. Artificial Intelligence based modeling methods are used to simulate and to make predictions of outcomes of the co-gasification process. Even though previous studies result in successful modelling for specific cases, limited selection of methods and lack of implementation of cross-validation techniques causes insufficiency to explain unbiased performance evaluations and up-scale usability of the methods. In this paper, six different regression methods are employed to predict outputs of co-gasification process using a dataset containing 56 observations. Moreover, the original dataset is randomly resampled so that each model's generalization ability is further assessed. The prediction performance of the proposed techniques on both datasets is evaluated and practical usability is discussed.

Keywords: gasification, co-gasification, machine learning, regression modeling, high-ash coal, biomass

AKIŞKAN YATAK GAZLAŞTIRICIDA YÜKSEK KÜLLÜ KÖMÜR VE BİYOKÜTLENİN BİRLİKTE GAZLAŞTIRILMASINDA KARIŞIM ORANI ETKİLERİNİN MAKİNE ÖĞRENMESİ İLE DEĞERLENDİRİLMESİ

Özet

Birlikte gazlaştırma, kömür ve biyokütle sentez gazı gibi faydalı ürünlere dönüştürülmesini sağlayan bir süreçtir. Birlikte gazlaştırma sürecinin modellenmesi için analitik ve sayısal yaklaşımlar ya çok uzun zaman ya da pratik uygulamalarda modellerin tutarlılığını azaltan çok sayıda varsayım gerektirir. Yapay zeka tabanlı modelleme yöntemleri, birlikte gazlaştırma sürecini simüle etmek ve sonuçlarını tahmin etmek için kullanılmıştır. Her ne kadar önceki çalışmalar belirli durumlar için başarılı bir modelleme ile sonuçlanmışsa da, sınırlı yöntem seçimi ve çapraz doğrulama tekniklerinin uygulanmaması, yansız performans değerlendirmelerini ve yöntemlerin büyük ölçekli kullanılabilirliğini açıklamakta yetersizliğe neden olmuştur. Bu çalışmada, 56 gözlem içeren bir veri seti kullanılarak birlikte gazlaştırma işleminin çıktılarını tahmin etmek için altı farklı regresyon yöntemi kullanılmıştır. Ardından, orijinal veri kümesi rastgele yeniden örneklenilip, böylece her modelin genelleme yeteneği daha fazla değerlendirilmiştir. Önerilen tekniklerin her iki veri seti üzerinde tahmin performansı değerlendirilmiş ve pratik kullanılabilirlik tartışılmıştır.

Anahtar Kelimeler: gazlaştırma, birlikte gazlaştırma, makine öğrenmesi, regresyon modellemesi, yüksek küllü kömür, biyokütle

Cite

Elmaz, F., (2019). "Evaluating the Effect of Blending Ratio on the Co-Gasification of High Ash Coal and Biomass in a Fluidized Bed Gasifier Using Machine Learning", *Mugla Journal of Science and Technology*, 5(1), 1-12.

1. Introduction

Worldwide energy consumption has increased significantly in recent years due to rapid urbanization

and industrial development. Energy consumption around the globe has increased from 3,701 million tonne of oil equivalent (mtoe) in 1965 to 13,511 mtoe [1].

Currently, this enormously growing energy demand is mainly met with fossil-based resources like coal, oil and natural gas. Fossil based resources are not only limited but also pose danger to the environment when processed with conventional methods used in power plants [2]. Increasing awareness about climate change and depleting reserves of fossil-based resources, countries around the world started seeking alternative methods to use their limited energy resources efficiently with more environment-friendly processes. For this reason, gasification caught attention as an efficient and cleaner way to produce energy. Gasification is a process that converts organic or fossil fuels to carbon monoxide, hydrogen and carbon dioxide by reacting the fuels with oxygen and steam in high temperatures [3]. Therefore, synthetic gas is obtained as a product which can be used as gaseous fuel or further processed to be used in electric and heat generation [4]. Co-gasification (COG) is a type of gasification processes which uses coal and biomass as a fuel to produce synthetic gas. Besides being an environment-friendly process [5], COG uses; low calorie, high sulfur, moisture and ash content biomasses. Those type of biomasses can be obtained from forest waste, agricultural waste, poultry waste, and municipal solid waste, and that makes biomasses abundantly available [6,7] and renewable resource around the world. Turkey has been estimated to have 135-150 mtoe/year biomass waste potential [8]. It has been known that series of chemical reactions take place simultaneously during the COG process [9]. For this reason, control of the process is not an easy task and must be done precisely. However, if parameters such as temperature, coal feed rate, biomass feed rate and air/fuel ratio are selected carefully before the process, the maximum quantities of desired products can be obtained [10,11]. In order to choose those parameters, a mathematical model to simulate the COG process must be created. Yet, it is difficult to develop such a model due to complexity of process which involves heat and mass transfer, pyrolysis of biomass material, cracking and subsequent steam reforming of tar vapor arising from the pyrolysis, heterogeneous gas-solid reactions and homogeneous gas-phase reactions [12]. Machine learning (ML) algorithms have been widely used to develop estimation models in order to understand and solve various type of complex problems encountered in different disciplines of science. Regression is a branch of ML based modeling algorithms and it is useful to predict continuous type variables. One of the most useful attribute of regression modeling is that it does not require any mathematical definition of the phenomena involved in the process, and therefore regression modeling is proven to effectively create prediction models for complex and non-linear processes. Even though there are many regression techniques, Artificial Neural Networks (ANNs) are among the most commonly used methods for modeling of gasification processes. Guo et al. used ANN modeling to predict the product yield and gas composition of biomass gasification in an atmospheric pressure steam fluidized bed gasifier [13].

Bed temperature and the stock residence time variables are used as features of the model. Even though developed prediction model is satisfactory in the aforementioned study, because of using only two features and lack of performance evaluation of the model using cross validation techniques, the study is insufficient to explain and prove the suitability of the proposed method for gasification problems. Arnavat et al. [14] proposed more sophisticated ANN model for biomass gasification with fluidized bed gasifiers, which used eight features (Ash content, moisture content, carbon content, oxygen content, hydrogen content, equivalence ratio, gasification temperature and steam to biomass ratio) and five outputs (Gas yield, gas composition, H₂ content, CH₄ content, CO₂ content and CO content) while creating the model. They developed single ANN model to predict all of the outputs and came up with a model that can predict each output with high accuracy. However, authors have used all of the data to train ANN model, thus proposed model's possible overfitting problem is not investigated with splitting the data into training and testing sets and/or using cross validation techniques. Pandey et al. [15] have proposed a set of methods to select and apply most suitable ANN architecture using same features used by Arnavat et al. and predicted lower heating value of gas, lower heating value of gasification products and syngas yield during gasification of municipal solid waste in a fluidized bed reactor. Proposed method iteratively changes number of neurons on the hidden layer and activation functions on the network to choose best fitting model, authors also randomly split dataset into training, testing and cross-validation sets. Although, relatively small dataset (100 observations) is used in the paper and cross validation methods such as K-Fold which uses all of the dataset for both training and testing purposes haven't applied even though these methods are crucial for accurate evaluation of the models' performance [16], thus best ANN architecture selection according to proposed method might be misleading due to repeated random splitting on the small dataset. Model selection with using K-Fold cross-validation methods would be more consistent and accurate. To the best of our knowledge, there is only one publication by Tiwary et al. [17] that focuses on modelling COG in fluidized bed gasifier with different type of regression algorithms. Authors created regression models using Support Vector Machines (SVM), ANN and Genetic Programming. Dataset used to create models consists of thirteen features (Gasification temperature, coal feed rate, biomass feed rate, fixed carbon, volatile matter, ash, moisture, air/fuel ratio, steam/fuel ratio, carbon, hydrogen, oxygen and rate constant) and 4 outputs (total gas yield, carbon conversion efficiency, heating value of product gas and cold gas efficiency). Regression models are developed to predict each output individually, thus total of twelve models are created. Authors split dataset into train and test sets. However, it's been observed that they iterated over different randomization seeds while splitting the dataset in order to determine best performing models on

both training and testing phases and shared performance results are accorded to those models. Also, cross validation methods weren't applied to ensure results are consistent. This approach resulted in highly biased performance evaluation and misleading comparisons between techniques. Therefore, the study results in insufficiency to explain which method is most suitable to create prediction models for COG processes. In this paper, we use the same dataset used by Tiwary et al. [17] and we create six regression models, i.e., Linear, Polynomial, Support Vector Machine, Decision Tree, Gaussian Process and ANN regression techniques, for estimating outputs of COG process. All regression models have been developed and trained to predict each output separately. K-Fold cross-validation has been employed and prediction performances are evaluated to show accurate validity and usability of the developed models on the dataset. Moreover, original dataset is randomly resampled so that generalization ability of the proposed techniques further analyzed. Prediction performances and practical usability of the proposed techniques are discussed.

2. Methods

2.1. Data Preprocessing

Data preprocessing is the initial step to be able to apply supervised machine learning algorithms successfully and to have better understanding of the data used [18]. Completion of missing data, removal of unnecessary features, data normalization, merging or removing features that are correlated are the few examples which are commonly used for data preprocessing. The particular data set used in this study does not contain any missing data to be filled in or any unnecessary features to be removed or merged. However, for the acceleration of training phase of the machine learning models and for preventing larger valued features from dominating scale-sensitive computation of metrics such as Euclidean distance, feature normalization is performed on the dataset with linear scaling to unit range:

$$\tilde{x}_i[n] = \frac{x_i[n] - \min(x_i)}{\max(x_i) - \min(x_i)} \quad (1)$$

where $x_i[n]$ is the i^{th} feature of n^{th} observation in the dataset, $\min(x_i)$ and $\max(x_i)$ are the minimum and maximum values of i^{th} feature across all observations in the dataset, respectively. Therefore, we obtain $\tilde{x}_i[n]$, which is the normalized version of the i^{th} feature for the n^{th} observation in the dataset. Therefore, the new set of features ranges from 0 to 1.

2.2. Resampling of Dataset

Resampling is a series of methods used for arbitrarily increasing the number of observations in a data set, which is useful to reduce the variance between predictions made by a machine learning model for different iterations [19]. Furthermore, due to the increase in size of the data set, regression models are trained with more samples which gives intuition about

up-scale performance of the used method. In order to perform resampling, one can take random samples from the original data set and append the new samples to the data set to increase its size. However, the process of appending observations that are already in the original data set may falsify the performance of the regression methods since the training and testing data sets can include the same observations. Therefore, random resampling can cause a biased prediction model that does not reflect the true learning ability. In this study, we propose to take random samples from the data set and to perturb resampled observations before appending them to the original data set. For this purpose, we add Gaussian noise for perturbation, where the noise power is computed using the Signal to Noise Ratio (SNR):

$$SNR = 10 * \log_{10} \left(\frac{P_x}{P_n} \right) \quad (2)$$

where P_x is the power of the feature column which noise will be added while resampling and P_n is the power of the noise. The original dataset with 56 observations is resampled such that the new data set consists of 255 observations using a SNR value of 13dB.

2.3. Linear Regression

Linear regression is one of the regression methods for mathematically modeling the relationship between one or more independent variables (features) and a response variable (output) by fitting a linear equation to observed data. Linear Regression can be represented as Equation (3).

$$\hat{y} = \beta_0 + x_1 * \beta_1 + x_2 * \beta_2 + \dots x_n * \beta_n \quad (3)$$

where \hat{y} is the predicted output, k is the number of features, x_i ($i = 1, 2, 3, \dots, n$) are the features which are fed to the model, β_i ($i = 0, 1, 2, \dots, n$) are the constant parameters determined in the training phase of the model. Obviously, β_i coefficients are crucially important for model to predict output variable accurately and must be determined carefully. Even though there are analytical approaches such as Normal Equation [20] for finding optimal β_i coefficients efficiently in single step for small datasets with relatively small number of features. These methods can be computationally expensive when applied to larger datasets. Because one of the goals of this paper is to investigate performances of the regression algorithms for COG process with larger datasets, iterative based and widely used optimization algorithm Batch Gradient-Descent is selected for tuning β_i coefficients. Batch Gradient-Descent algorithm requires a cost function J to be minimized, commonly used cost function sum of the squared errors (SSE) is selected due to its convex structure. Algorithm starts tuning with assigning random values for each β_i coefficients. Then, J is calculated by subtracting predicted output value and the real output value (y) of each observation, taking squares of the results and summing them together (Equation (4)). A learning rate (α) is used as a scale factor and the coefficients are updated simultaneously in the direction towards minimizing the J by using Equation (5). The

process is repeated until a minimum J is achieved or no further improvement is possible.

$$SSE = J(\beta_0, \beta_1, \dots, \beta_n) = \sum_{i=1}^m (y_i - \hat{y}_i)^2 \quad (4)$$

$$\beta_{0,1,2\dots n} := \beta_{0,1,2\dots n} - \alpha * \frac{\partial J(\beta_0, \beta_1, \dots, \beta_n)}{\partial \beta_{0,1,2\dots n}} \quad (5)$$

where, m is the number of observations in the dataset

2.4. Polynomial Regression

Polynomial Regression is similar to Linear Regression in a way but rather than fitting data to a linear function, Polynomial Regression fits data to a polynomial function. There is no rule of thumb for determining the polynomial equation to be used. It must be selected specifically for the dataset which the method will applied to. According to authors' experience and preparation for this paper, quadratic function (Equation (6)) is selected as polynomial equation used in this method.

$$\hat{y} = \sum_{j=1}^n \sum_{k=j}^n \beta_{jk} * x_j * x_k \quad (6)$$

where, β_{jk} are the constant coefficients determined in the training phase of the model. While executing Polynomial Regression, overfitting is commonly encountered problem [21]. Overfitting occurs when a model excessively learns the details of the training data. This situation reduces models' ability to generalize [22]. Authors recognized overfitting problem for the dataset and equation used in this method when no precautions are taken. In order to prevent overfitting problem, 'regularization term' is added to the cost function to minimize the square of β_{jk} coefficients while minimizing the error (Equation (7)). Even though this method results in worse training fit of the data, it has been observed models' generalization to predict new data is improved. Hence, cost function is updated with regularization term and Polynomial Regression model is trained with Batch Gradient-Descent optimization algorithm.

$$J(\beta_{11}, \beta_{12}, \dots, \beta_{(n)(n)}) = \sum_{i=1}^m (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^n \sum_{k=1}^n \beta_{jk}^2 \quad (7)$$

where, λ is the regularization parameter.

2.5. Decision Tree Regression

Unlike the linear and nonlinear regression methods where the relationship between features and output is predetermined, Decision Tree doesn't assume such relationship exists. It is a method of constructing a set of decision rules on predictor variables [23]. The rules are created by repeatedly dividing the data into smaller groups in succession with binary sections based on a single output variable. The divisions for all predictors are examined by a search procedure and the best division is selected. Decision Tree Regression is a type of Decision

Tree classification algorithm that can be used to estimate continuous type variables like we're aimed to predict in this paper. The algorithm of a Decision Tree Regression is also based on repeated binary partitioning, which is a recursive process that separates it from the data partitions. Initially, all training samples are used to determine the structure of the tree. The algorithm then splits the data using every possible binary split and selects the split that partitions the data into two parts such that it minimizes the SSE from the mean in the separate parts. Then, same process is applied to each of the added new branches. The process is repeated until each node reaches a pre-specified minimum node and becomes a terminal node. Like many of the complex regression algorithms, overfitting problem may occur in Decision Tree Regression models. In order to prevent overfitting, a process called 'pruning' is used. In pruning process a validation dataset and cost complexity factor which is a parameter representing the complexity cost per node is used. Therefore, a pruning process' aim is to minimize the sum of the output variable variance in the validation data, the product of the number of terminal nodes and the cost complexity factor. This process starts with removing the last created node and keeps removing until the desired minimization is achieved.

2.6. Support Vector Regression

A Support Vector Machine is a machine learning algorithm which widely used classification and regression problems [24]. The goal of the Support Vector Machine is finding the optimal linear hyperplane such that the error on test samples is minimized which can be achieved by good generalization performance. Generalization performance in Support Vector Machines is determined by structural risk minimization inductive principle [25]. In order to create a Support Vector Machine model for a regression problem, features are first mapped into a high dimensional space (F) with a kernel function and they are correlated linearly with the outputs. General linear estimation function of Support Vector Machine Regression is given as Equation (8).

$$f(x) = \sum_{i=1}^n a_i * K(x_i, x) + b \quad (8)$$

where, a_i ($i = 1, 2, 3 \dots, n$) are constant parameters, b is the bias term and $K(x_i, x)$ is the kernel function. In this paper, polynomial kernel function is used to map feature into higher dimension space (Equation (9)). Degree of the polynomial is selected as three which commonly referred as cubic polynomial kernel function.

$$K(x_i, x_j) = (x_i'x_j + 1)^d \quad (9)$$

where; $d = 1, 2, 3 \dots$ which indicates the degree of the polynomial function, x is the input points in the training set and d is the constant value which influences coefficients of the higher order and lower order terms in the kernel. In Support Vector Machines, a nonlinear regression problem can be converted to a linear regression problem with high dimension feature space.

Even though, there are several cost functions that can be used [26], regularized cost function (J) with robust ε -insensitive loss function (L_ε , Equation (10)) is found appropriate by the authors and implemented in this paper (Equation (10)). With the minimization of selected cost function, model is trained to find optimal weight vector and constant term.

$$L_\varepsilon = \begin{cases} 0 & \text{if } |y - f(x)| \leq \varepsilon \\ |y - f(x)| - \varepsilon & \text{otherwise} \end{cases} \quad (10)$$

$$J = \sum_{i=1}^n L_\varepsilon(f(x_i), y) + \frac{\lambda}{2} |w|^2 \quad (11)$$

2.7. Gaussian Process Regression

Gaussian Process Regression is a method which extends the idea of a probability distribution of numbers to a probability distribution of functions. For a given training set, the aim is to learn a function which converts the feature vector (X) into the output value (y) with Equation (12). This equation can be described by a Gaussian Distribution (Equation (13)).

$$y = f(X) + \varepsilon \quad (12)$$

$$y \sim N(0, K(X, X) + \sigma^2 I) \quad (13)$$

where, ε is the Gaussian noise with zero mean and variance σ^2 and $K(X, X)$ is the kernel function. Exponential kernel function is selected for Gaussian Process Regression method in this paper (Equation (14))

$$K(x_i, x_j) = \sigma_f^2 e^{-\frac{r}{\sigma_l}} \quad (14)$$

where, σ_l is the characteristic length scale, σ_f is the signal standard deviation and r is Euclidean distance between x_i and x_j . In order to make satisfactory predictions with Gaussian Process Regression model, hyper-parameter matrix which is given as $\theta = [\sigma^2, \sigma_l, \sigma_f]$ must be set to the optimal values in the training phase. This optimization problem can be solved by maximizing the regularized log marginal likelihood function given in Equation (15).

$$\log p(y|X, \theta) = -\frac{1}{2} y^T K_y^{-1} y - \frac{1}{2} \log |K_y| - \log 2\pi \quad (15)$$

2.8. Artificial Neural Networks

ANNs modeling is widely used machine learning methodology originally designed to mimic the way how human brain analyzes and processes information. ANNs are composed of a number of interconnected simple processing elements called neurons connected with

coefficients (weights) to other neurons and are organized in layers [27]. First layer which includes features of the dataset is called 'input layer' and final layer of the model is called 'output layer'. Every layer stays between input and output layers called as 'hidden layers'. Even though there are plenty of ANNs architectures which determines how neurons and layers are connected to each other, most commonly used 'Feedforward' architecture is selected and implemented in this paper. In feedforward ANNs, neurons receive an input signal which is the sum of all 'information' from other nodes in the previous layer and external bias term, then applies an activation function to the summation and produces output signal for other neurons in the following layer. This process is repeated until the neuron(s) of the output layer produces output(s). Parameters like number of layers, number of the neurons on the layers and activation functions of the neurons are crucially important for an ANNs model to make accurate predictions. In this paper, an ANNs models are created with single hidden layer includes 5 neurons and an output layer with single neuron. Sigmoid activation function (Equation (16)) is used in the hidden layer and linear activation (Equation (17)) function used in the output layer.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (16)$$

$$f(x) = c * x \quad (17)$$

where, c is the constant value adjusts the proportion of the input information to the output. In the training phase of the model, weights are adjusted with Backpropagation and Batch Gradient-Descent algorithms. In Backpropagation, SSE is used as cost function and every weight's contribution on the cost (gradients) are calculated with propagating error back to the corresponding weights [28] and Batch-Gradient Descent algorithm is used to update the weights.

2.9. K-Fold Cross-Validation

Cross-validation (CV) methods are used to assess the predictive performance of the models and calculate how they perform outside of training data. There are different cross validation methods such as Hold-Out [16] requires an independent partition of the data and validates the model according the partition separated. This type of cross validation methods can have high variance especially for small datasets. When a dataset has small number of observations, the partition which will be separated for the cross-validation will also have small number of observations so there will be a lot of variation in the performance estimate for different samples of data, or for different partitions of the data to form training and test sets. In the K-fold cross validation method, the data set is divided into 'K' equal folds and a

model is trained over remaining K-1 folds. Then the created model is tested on the remaining fold. The same operation is repeated 'K' times with the selection of different folds, so that the entire data set is used for both training and testing. K-fold validation reduces variance by using K different partitions and the performance estimate is less sensitive to the partitioning of the data compared to cross validation methods which performs validation on a partition of the data. In this paper 'K' number is selected as 10 (also referred as 10-Fold CV) because it has been observed that 10-Fold CV reflects the most unbiased generalization performance of the models compared to other choices for 'K' [29].

3. Results and Discussion

The proposed six regression methods are used to develop prediction models using COG experiment dataset with 13 features, x_1, x_2, \dots, x_{13} referred to as gasification temperature, coal feed rate, biomass feed rate, fixed carbon, volatile matter, ash, moisture, air/fuel ratio, steam/fuel ratio, carbon, hydrogen, oxygen and rate constant, respectively. The developed regression models are used to estimate four outputs, y_1, y_2, y_3, y_4 i.e., total gas yield (kg/kg-fuel), carbon conversion efficiency (%), heating value of product gas (MJ/Nm³) and cold gas efficiency (%) using the data set consisting of 56 observations which is provided in [17]. The total gas yield and carbon conversion efficiency are important performance parameters, which measure the conversion rate of solid biomass to syngas. But additional parameters are also needed to determine whether the syngas is usable for combustion or not. The heating value of product gas and cold gas efficiency give the energy content of the syngas. For energy applications, they are equally significant. Therefore, one needs to estimate all of the outputs, i.e., y_1, y_2, y_3, y_4 , for reliably assessing the gasification performance. All prediction models are created for predicting single output and all methods are applied using 10-Fold cross-validation. The performance of the models is evaluated by employing the Correlation Coefficient (R^2), Adjusted R^2 (Adj. R^2), Root-Mean-Square-Error (RMSE) and Normalized Root-Mean-Square-Error (NRMSE) metrics:

$$R^2 = 1 - \frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2}{\sum_{i=1}^m (y_i - \bar{y})^2} \quad (18)$$

$$Adj. R^2 = 1 - \frac{[(1 - R^2) * (m - 1)]}{m - n - 1} \quad (19)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2}{m}} \quad (20)$$

$$NRMSE = \sqrt{\frac{1}{m} * \sum_{i=1}^m \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2} \quad (21)$$

where, \bar{y} is the mean of the output vector, $\min(y)$ and $\max(y)$ are the minimum and maximum values of the output vector, respectively. Table 1 illustrates the performance of the proposed techniques in predicting the four distinct output variables. One should note that the R^2 , Adj. R^2 , Root-Mean-Square-RMSE and NRMSE values should all be considered in order to evaluate a model performance. However, if the model is not linear, RMSE and NRMSE values would provide more reliable information which can lead us to interpret and compare model performance better. Since we use 10-Fold cross-validation, all of the observations in the dataset are used as both training and testing purposes. For a better understanding of the regression models' performances, the predicted outputs of all 56 observations versus actual observations are plotted where all values are scaled in the range of [0, 1]. As shown in Fig.1 and Fig.2, the straight reference lines indicate the ideal case where the predicted output levels and the true levels are equal. Hence, best scenario is achieved when a particular data point is on the reference line, which means the particular data is predicted correctly. For the sake of simplicity, only the two regression methods, Gaussian Regression and Support Vector Regression, with the best performances are presented in this paper. One can see from the Table 1 and the regression plots that the total gas yield, y_1 , has a strong linear relationship with the features since we achieve high prediction performance using Linear Regression. On the other hand, the performance of the linear regression method drops dramatically for the rest of the outputs which means there is no linear relationship between the features and the outputs. For the case of Polynomial Regression, prediction accuracy is just slightly improved compared to Linear Regression, but high variance between the prediction accuracy of the folds is observed. The reason for this high-variance attitude is mainly due to small size of the dataset. Decision Tree Regression has the worst overall prediction accuracy compared to other methods presented in this paper. As it will be mentioned later with the application of the same methods on a resampled dataset, Decision Tree Regression method is more successful at predicting outputs when applied on larger datasets. Even though ANN regression can be considered successful for predicting y_1 and heating value of product gas, y_3 , its' performance in predicting the carbon conversion efficiency, y_2 , and cold gas efficiency, y_4 , reduces. Also, same high variance between test folds observed as encountered in Polynomial Regression method. Support Vector Regression and Gaussian Regression perform better than the remaining approaches. When Fig.1 and Fig.2 are examined, both methods share similar failed predictions in data points. One of the reason for this behavior is that there are some

data points especially in y_2 and y_4 outputs that do not follow the same ‘pattern’ with the other data points in respective outputs as discussed in [17]. This situation makes some data points almost unpredictable because regression algorithms perform better in predicting values which follow the closer ‘patterns’. In order to have further understanding about the up-scale generalization abilities of the proposed methods, we use random resampling so that the size of the dataset is increased from 56 to 255 observations. Furthermore, the resampled observations are perturbed using additional Gaussian noise with a signal-to-noise ratio (SNR) of 13 dB to prevent overfitting issue discussed in Section 2.2. We selected 13 dB SNR value such that the noise power is approximately 5% of the signal power, which enables us to perturb the original data set without losing the information carried by the real observations (samples). We also used higher and lower SNR values to generate bootstrapped data. However, for very low SNR values, the addition of bootstrapped data is worsened the regression results significantly. On the contrary, the use of very high SNR values resulted in the addition of almost the same samples from the original data to the test and training samples. Hence, we obtained a very high prediction performance since the training data and the test data included very similar samples.

Table 2 illustrates the performance of the proposed approaches in predicting output levels using the

resampled data set. As one can see, the best performing methods are the Gaussian and Decision Tree regression, which are also indicated by the Figs. 3 and 4, respectively. When Table 1 and Table 2 are compared, the prediction performance for the output y_1 has dropped slightly. In spite of the fact that the proposed regression techniques have larger number of samples to learn from the data set, the performance decrease is caused by the added Gaussian noise, which also perturbs the linear relationship between the output y_1 and the given features. Moreover, it can be seen that overall prediction performances of the linear regression, polynomial regression, Support Vector regression and ANN regression have dropped using the larger resampled data set, indicating a worsened generalization ability. One should note that even the dataset is perturbed with noise, the performance decrease should be taken into account since one could encounter with noise factors such as experimental errors or differences between sensor sensitivities in real life scenarios. On the other hand, the performance of the Gaussian Regression model in predicting all outputs except for the y_1 has increased using the resampled data set. However, one should note that the added Gaussian noise is in accordance with the Gaussian regression model and this could have caused an increase in the model prediction performance since the noise is also modeled by the regression model. The performance of the decision tree regression has also increased significantly when applied on the resampled data set.

Table 1. Prediction performances of the proposed regression methods

Outputs	Linear Regression				Polynomial Regression				Gaussian Regression			
	R^2	$Adj.R^2$	RMSE	NRMSE	R^2	$Adj.R^2$	RMSE	NRMSE	R^2	$Adj.R^2$	RMSE	NRMSE
y_1	0.91	0.94	0.049	0.042	0.89	0.86	0.082	0.077	0.95	0.94	0.04	0.071
y_2	0.58	0.45	2.964	0.091	0.64	0.62	3.099	0.192	0.78	0.71	2.11	0.174
y_3	0.74	0.66	0.298	0.102	0.76	0.7	0.29	0.174	0.77	0.7	0.266	0.168
y_4	0.69	0.59	6.077	0.101	0.69	0.61	6.037	0.098	0.71	0.62	5.709	0.095
Outputs	Support Vector Regression				Decision Tree Regression				ANN Regression			
	R^2	$Adj.R^2$	RMSE	NRMSE	R^2	$Adj.R^2$	RMSE	NRMSE	R^2	$Adj.R^2$	RMSE	NRMSE
y_1	0.95	0.93	0.044	0.067	0.88	0.84	0.09	0.085	0.92	0.89	0.077	0.073
y_2	0.77	0.71	2.056	0.14	0.58	0.45	3.547	0.12	0.68	0.66	2.627	0.18
y_3	0.79	0.72	0.281	0.156	0.42	0.36	0.407	0.139	0.77	0.72	0.328	0.176
y_4	0.7	0.61	6.743	0.112	0.41	0.33	7.751	0.163	0.69	0.6	7.042	0.111

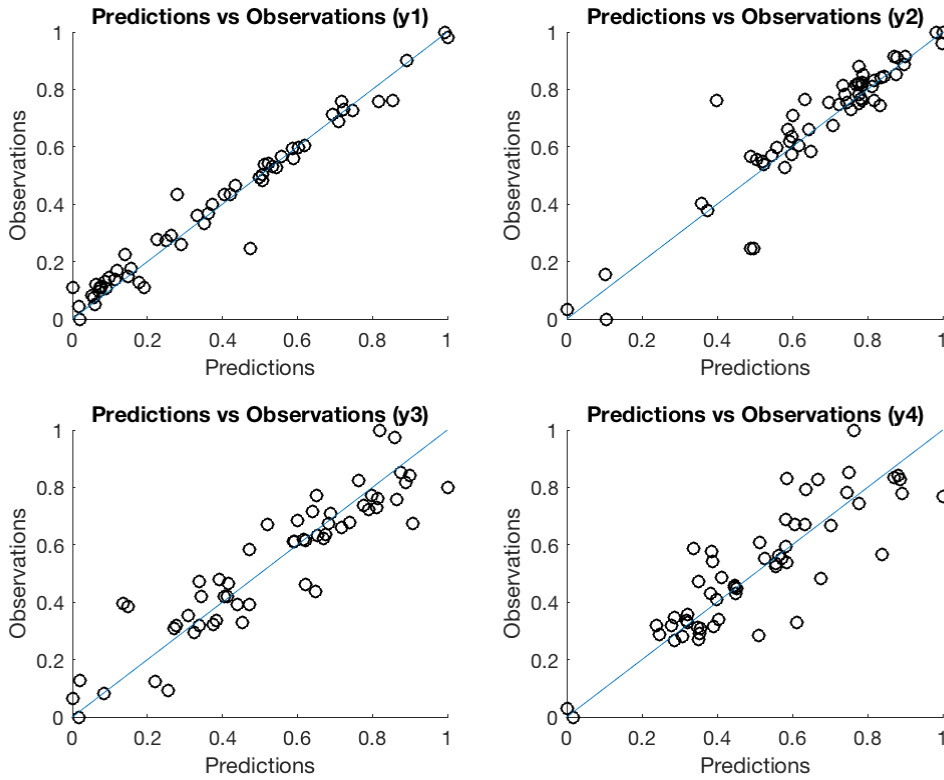


Figure 1. Actual values vs predictions made by Gaussian Regression models

The reason for this is that decision tree regression has also increased significantly when applied on the resampled data set. The reason for this is that the decision tree regression performs better as the size of the training data set increases [32]. However, Fig. 4 shows that the predictions made by the decision tree regression models are either almost totally on the reference line, i.e., predicted very accurately, or totally off the reference line. This is caused by the learning approach that the decision tree algorithm utilizes, which creates a strict set of rules to predict output levels. When the high preciseness required for COG process is concerned, Decision Tree Regression modelling may not provide good results as expected in practical applications of COG processes.

4. Conclusion

In this paper, we proposed machine learning regression techniques to predict four distinct outputs COG process using thirteen features out of the proposed six models, Gaussian Process regression was the best performing method for predicting outputs of the COG process for the both original and resampled datasets. On the other hand,

linear regression and polynomial regression fail to generalize and predict the carbon conversion efficiency and cold gas efficiency outputs. In addition, the prediction performance of the Support Vector Regression is significantly diminished for the resampled data set compared to the original data set. Hence, the Support Vector Regression may not be a good choice for modelling of the COG process. On the contrary, the performance of the Decision Tree Regression technique is increased when the resampled dataset is used since Decision Tree based methods learn better using larger datasets. The future work will focus on improving feature selection and identifying the importance of features such that overall prediction performance is enhanced. Moreover, dimensionality reduction techniques will be pursued since the application of the proposed artificial intelligence based regression approaches requires low computational cost for embedded designs. We believe that machine learning regression techniques could pave the way for researchers and professionals to analyze and predict the dynamics of co-gasification.

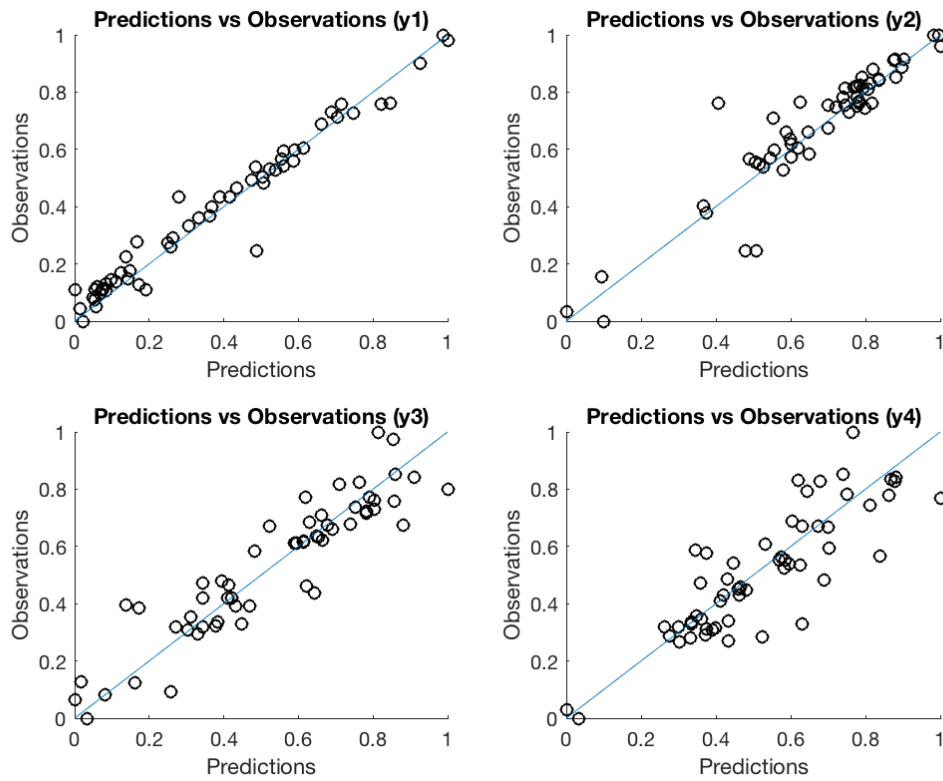


Figure 2. Actual values vs predictions made by Support Vector Regression models

Table 2. Prediction performances of the proposed methods on resampled dataset

Outputs	Linear Regression				Polynomial Regression				Gaussian Regression			
	R^2	$Adj.R^2$	$RMSE$	$NRMSE$	R^2	$Adj.R^2$	$RMSE$	$NRMSE$	R^2	$Adj.R^2$	$RMSE$	$NRMSE$
y_1	0.86	0.81	0.099	0.093	0.9	0.87	0.086	0.092	0.92	0.89	0.077	0.082
y_2	0.47	0.31	4.595	0.156	0.51	0.37	4.232	0.177	0.79	0.74	3.193	0.123
y_3	0.65	0.55	0.387	0.132	0.67	0.57	0.379	0.149	0.82	0.76	0.268	0.107
y_4	0.45	0.28	9.296	0.155	0.55	0.41	8.297	0.138	0.78	0.75	5.683	0.094
Outputs	Support Vector Regression				Decision Tree Regression				ANN Regression			
	R^2	$Adj.R^2$	$RMSE$	$NRMSE$	R^2	$Adj.R^2$	$RMSE$	$NRMSE$	R^2	$Adj.R^2$	$RMSE$	$NRMSE$
y_1	0.89	0.86	0.084	0.09	0.88	0.85	0.081	0.089	0.9	0.87	0.083	0.089
y_2	0.72	0.63	3.296	0.124	0.7	0.65	3.205	0.153	0.59	0.47	3.484	0.152
y_3	0.65	0.55	0.355	0.141	0.77	0.74	0.273	0.112	0.71	0.62	0.344	0.142
y_4	0.54	0.4	7.871	0.131	0.71	0.69	6.315	0.109	0.57	0.44	7.585	0.126

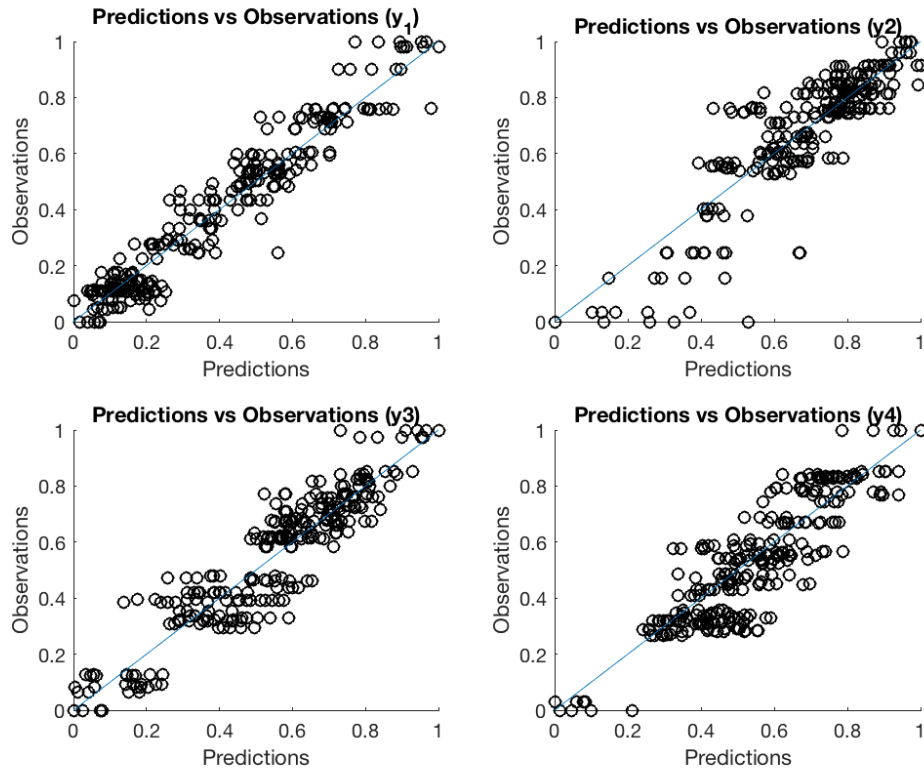


Figure 3. Actual values vs predictions made by Gaussian Regression models on resampled dataset

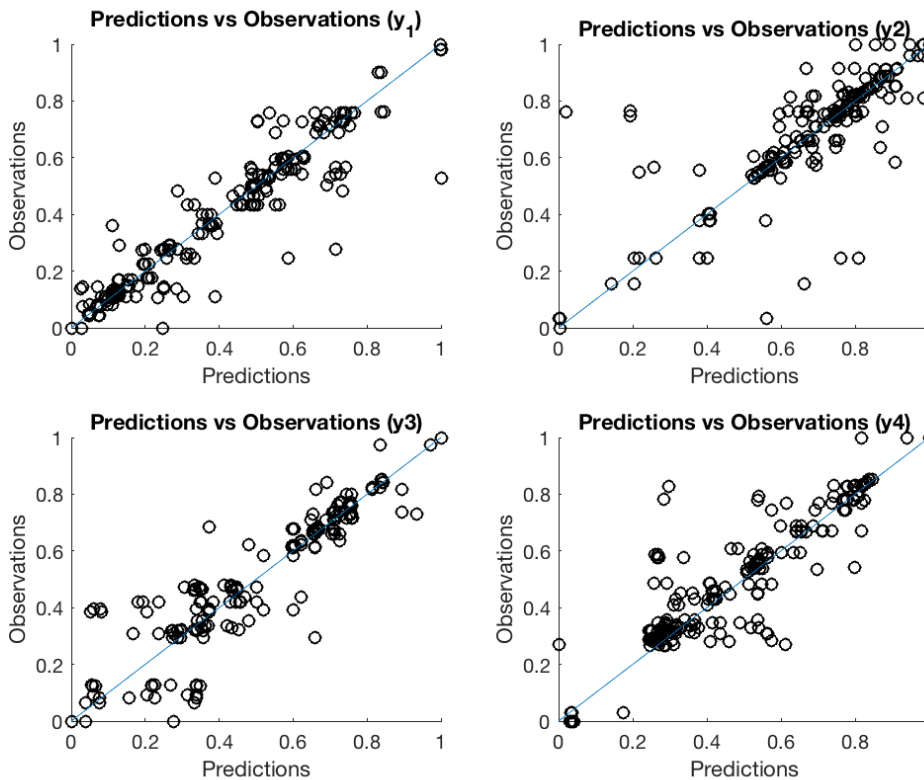


Figure 4. Actual values vs predictions made by Decision Tree Regression models on resampled dataset

5. References

- [1] BRITISH PETROLEUM COMPANY, "Statistical Review of World Energy," *Statistical Review of World Energy*, Available: <http://www.bp.com/productlanding.do?categoryId=6929&contentId=7044622>, 2018.
- [2] Panwar, N. L., Kaushik, S. C., and Kothari, S., "Role of renewable energy sources in environmental protection: A review," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 3, Pergamon, pp. 1513–1524, 2011.
- [3] Srivastava, T., "Renewable Energy (Gasification)," *Adv. Electron. Electr. Eng.*, vol. 3, no. 9, pp. 1243–1250, 2013.
- [4] Mutlu, A. Y. and Yucel, O., "An artificial intelligence based approach to predicting syngas composition for downdraft biomass gasification," *Energy*, vol. 165, pp. 895–901, 2018.
- [5] Kamińska-Pietrzak, N. and Smoliński, A., "Selected Environmental Aspects of Gasification and Co-Gasification of Various Types of Waste," *J. Sustain. Min.*, vol. 12, no. 4, pp. 6–13, 2013.
- [6] Akia, M., Arandiyani, H., Yazdani, F., Han, D., and Motaeae, E., "A review on conversion of biomass to biofuel by nanocatalysts," *Biofuel Res. J.*, vol. 01, no. 01, pp. 16–25, 2015.
- [7] Ud Din, Z. and Zainal, Z. A., "Biomass integrated gasification-SOFC systems: Technology overview," *Renewable and Sustainable Energy Reviews*, vol. 53, pp. 1356–1376, 2016.
- [8] Karayilmazlar, S. *et al.*, "Biyokütlenin Türkiye’de Enerji Üretiminde Değerlendirilmesi," *J. Bartın Fac. For.*, vol. 19, no. 13, pp. 63–75, 2011.
- [9] Arabloo, M., Bahadori, A., Ghiasi, M. M., Lee, M., Abbas, A., and Zendejboudi, S., "A novel modeling approach to optimize oxygen-steam ratios in coal gasification process," *Fuel*, vol. 153, pp. 1–5, 2015.
- [10] Seo, H. K. *et al.*, "Effects of operating factors in the coal gasification reaction," *Korean Journal of Chemical Engineering*, vol. 28, no. 9, pp. 1851–1858, 2011.
- [11] Kalina, J., "Retrofitting of municipal coal fired heating plant with integrated biomass gasification gas turbine based cogeneration block," *Energy Convers. Manag.*, vol. 51, no. 5, pp. 1085–1092, May 2010.
- [12] Sueyasu, T., Oike, T., Mori, A., Kudo, S., Norinaga, K., and Hayashi, J. I., "Simultaneous steam reforming of tar and steam gasification of char from the pyrolysis of potassium-loaded woody biomass," in *Energy and Fuels*, vol. 26, no. 1, pp. 199–208, 2012.
- [13] Guo, B., Li, D., Cheng, C., Lü, Z. A., and Shen, Y., "Simulation of biomass gasification with a hybrid neural network model," *Bioresour. Technol.*, vol. 76, no. 2, pp. 77–83, 2001.
- [14] Puig-Arnabat, M., Hernández, J. A., Bruno, J. C., and Coronas, A., "Artificial neural network models for biomass gasification in fluidized bed gasifiers," *Biomass and Bioenergy*, vol. 49, pp. 279–289, 2013.
- [15] Pandey, D. S., Das, S., Pan, I., Leahy, J. J., and Kwapinski, W., "Artificial neural network based modelling approach for municipal solid waste gasification in a fluidized bed reactor," *Waste Manag.*, vol. 58, pp. 202–213, 2016.
- [16] Kohavi, R., "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, vol. 2, no. 0, pp. 1137–1143, 1995.
- [17] Saha, S. *et al.*, "Co-gasification of High Ash Coal–Biomass Blends in a Fluidized Bed Gasifier: Experimental Study and Computational Intelligence-Based Modeling," *Waste and Biomass Valorization*, pp. 1–19, 2018.
- [18] Astorino, A., Gorgone, E., Gaudioso, M., and Pallaschke, D., "Data preprocessing in semi-supervised SVM classification," *Optimization*, vol. 60, no. 1–2, pp. 143–151, 2011.
- [19] Robert, C., "Machine Learning, a Probabilistic Perspective," *CHANCE*, vol. 27, no. 2, pp. 62–63, 2015.
- [20] Takajo, H. and Takahashi, T., "Noniterative method for obtaining the exact solution for the normal equation in least-squares phase estimation from the phase difference," *J. Opt. Soc. Am. A*, vol. 5, no. 11, p. 1818, 2008.
- [21] Ecology, S. and Apr, N., "Nonlinear Redundancy Analysis and Canonical Correspondence Analysis Based on Polynomial Regression," *Ecology*, vol. 83, no. 4, pp. 1146–1161, 2013.
- [22] Hawkins, D. M., "The Problem of Overfitting," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 1–12, 2004.
- [23] Prasad, A. M., Iverson, L. R., and Liaw, A., "Newer classification and regression tree techniques: Bagging and random forests for ecological prediction," *Ecosystems*, vol. 9, no. 2, pp. 181–199, 2006.
- [24] Bai, Y., Li, Y., Liu, Y., and Ma, Z., "Short-term prediction of distribution network faults based on support vector machine," in *Proceedings of the 2017 12th IEEE Conference on Industrial Electronics and Applications, ICIEA 2017*, vol. 2018-Febru, pp. 1421–1426, 2018.
- [25] Lu, C. J., Lee, T. S., and Chiu, C. C., "Financial time series forecasting using independent component analysis and support vector regression," *Decis. Support Syst.*, vol. 47, no. 2, pp. 115–125, 2009.
- [26] Williams, C. K. I., "Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond," *J. Am. Stat. Assoc.*, vol. 98, no. 462, pp. 489–489, 2009.
- [27] Zhang, G., Eddy Patuwo, B., and Y. Hu, M., "Forecasting with artificial neural networks:," *Int. J. Forecast.*, vol. 14, no. 1, pp. 35–62, 2002.
- [28] Gudise, V. G. and Venayagamoorthy, G. K., "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks," in *2003 IEEE Swarm Intelligence*

Symposium, SIS 2003 - Proceedings, pp. 110–117, 2013.

- [29] Ambroise, C. and McLachlan, G. J., “Selection bias in gene extraction on the basis of microarray gene-expression data,” *Proc. Natl. Acad. Sci.*, vol. 99, no. 10, pp. 6562–6566, 2002.