F. ELMAZ

IZMIR KATIP CELEBI UNIVERSITY

2020

IZMIR KATIP CELEBI UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A MACHINE LEARNING APPROACH TO BIOMASS GASIFICATION PROCESS

M.Sc. THESIS

Furkan ELMAZ

Department of Electrical and Electronics Engineering

Thesis Advisor: Prof. Dr. Adnan KAYA
Thesis Co-Advisor: Asst. Prof. Dr. Özgün YÜCEL

FEBRUARY 2020

IZMIR KATIP CELEBI UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A MACHINE LEARNING APPROACH TO BIOMASS GASIFICATION PROCESS

M.Sc. THESIS

Furkan ELMAZ

Department of Electrical and Electronics Engineering

Thesis Advisor: Prof. Dr. Adnan KAYA
Thesis Co-Advisor: Asst. Prof. Dr. Özgün YÜCEL

FEBRUARY 2020

İZMİR KATİP CELEBİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİYOKÜTLE GAZLAŞTIRMASI SÜRECİNE MAKİNE ÖĞRENMESİ YAKLAŞIMI

YÜKSEK LİSANS TEZİ

Furkan ELMAZ
Y180207003

Elektrik-Elektronik Mühendisliği Ana Bilim Dalı

Tez Danışmanı: Prof. Dr. Adnan KAYA
İkinci Tez Danışmanı: Dr. Öğr. Üyesi Özgün YÜCEL

ŞUBAT 2020

Furkan ELMAZ, a **M.Sc.** student of **IKCU Graduate School Of Natural And Applied Sciences**, successfully defended the thesis entitled "**A MACHINE LEARNING APPROACH TO BIOMASS GASIFICATION**", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisors:

Prof. Dr. Adnan KAYA
İzmir Kâtip Çelebi University .....

Jury Members:

Assoc. Prof. Dr. Savaş ŞAHİN
İzmir Kâtip Çelebi University .....

Asst. Prof. Dr. Özgün YÜCEL
Gebze Technical University .....

Asst. Prof. Dr. Volkan KILIÇ
İzmir Kâtip Çelebi University .....

Assoc. Prof. Dr. Devrim ÜNAY
İzmir Democracy University .....

Date of Defense : 13.03.2020

To my family

FOREWORD

First and foremost, I would like to thank Dr. Ali Yener Mutlu for his unmatched guidance and patience throughout the majority of my M.Sc. education as my supervisor. Your vision, knowledge and professionalism not only guided but expanded my excessive passion for machine learning. You taught me to challenge myself to be persistent and critical toward my work, as well as developing myself to be a better researcher and a better person.

Secondly, I would like to thank my co-advisor Asst. Prof. Dr. Özgün Yücel, your magical ability for finding the shortcomings of the literature with extreme precision enabled me to develop unique solutions to your findings with machine learning and to create high-quality work. Your expertise, hard work and guidance improved the quality of this thesis to the point that I could not imagine. This thesis would be impossible to complete in your absence.

I also want to thank Prof. Dr. Adnan Kaya for supervising in the last months of my M.Sc. I was able to continue on my thesis and projects with a clear mind in difficult times thanks to your never-ending support and guidance.

Also, I want to thank my lab partner Barkın Büyükçakır for his companionship in many sleepless hours we spent together. The laughs and pleasant memories we created had undoubtedly helped us to stay sane during our extremely hard working. I will surely miss the times we had, and our ability to find something to laugh even in the worst of the days.

Last but not least, I would like to thank TUBITAK for supporting my thesis under "2210-C Yurt İçi Lisansüstü Burs Programı" by grant 1649B021907037.

February 2020

Furkan ELMAZ

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

ABBREVIATIONS

KNN : K-nearest Neighbors

SVM : Support Vector Machine

DT : Decision Tree

RF : Random Forest

HC : Hierarchical Classification

KCV : K-fold Cross-validation

FC : Flat Classification

PCA : Principal Component Analysis

PR : Polynomial Regression

SVR : Support Vector Regression

DTR : Decision Tree Regression

ANN : Artificial Neural Network

NARX : Nonlinear Auto Regressive with Exogenous Inputs

NARXNN : Nonlinear Auto Regressive with Exogenous Inputs Neural Network

MPC : Model Predictive Controller

RWA : Rolling-windows Analysis

A MACHINE LEARNING APPROACH TO BIOMASS GASIFICATION PROCESS

ABSTRACT

Machine learning (ML) has been paving the way for researchers to create unique data-driven solutions in many areas of science by offering a strong set of computational tools. Especially the "collecting related input and output data and letting ML algorithms try to discover the underlying phenomenon" approach enabled researchers to overcome severe limitations of conventional analytical and/or numeric approaches in their respective fields. One such phenomenon is the biomass gasification. Biomass gasification is a promising power generation process due to its ability to utilize waste materials and similar renewable energy sources that is highly open to ML augmentation due to its complex and unpredictable nature. In this thesis, challenges of the biomass gasification are undertaken from ML perspective. Firstly, a hierarchical classification framework is developed by employing ML classifiers to distinguish solid fuels for their optimal use. With this framework, over 92% classification accuracy is obtained. Secondly, ML regression techniques are applied on an experimentally collected data set. Performances of the proposed regression models are evaluated with k-fold cross validation. artificial neural networks and decision tree regression outperformed other modeling approaches in the literature by achieving $R^2 > 0.9$ for the majority of outputs. Lastly, a time series modeling approach is implemented on biomass gasification process and $R^2 > 0.98$ for all outputs is achieved. Furthermore, a model predictive controller (MPC) is designed to control output concentrations of the biomass gasification process. Designed MPC is challenged in practical scenarios. MPC showed satisfactory performance for all scenarios and also showed high compliance with the experimental data which further strengthened its practical usability potential.

BİYOKÜTLE GAZLAŞTIRMASI SÜRECİNE MAKİNE ÖĞRENMESİ YAKLAŞIMI

ÖZET

Makine öğrenmesi (ML), araştırmacılara güçlü hesaplama araçları sunarak bilimin birçok alanında benzersiz veri odaklı çözümler yaratmalarının yolunu açmaktadır. Özellikle "ilgili girdi ve çıktı verilerinin toplanarak ve ML algoritmalarının sayesinde altında yatan sürecin keşfedilmesi" yaklaşımı, araştırmacıların kendi alanlarında geleneksel analitik ve/veya numerik yaklaşımların sahip olduğu sınırlamaların üstesinden gelmelerini sağlamıştır. Biyokütle gazlaştırmasıdır da bu süreçlerden biridir. Biyokütle gazlaştırması, biyolojik atık ve benzeri yenilenebilir enerji kaynaklarını kullanma kabiliyeti nedeniyle umut verici bir enerji üretim sürecidir. Bu tezde, biyokütle gazlaşmasının sahip olduğu zorluklar ML perspektifinden ele alınmıştır. İlk olarak, katı yakıtları optimum kullanımlarına ayırmak için ML sınıflandırıcıları kullanılarak hiyerarşik bir sınıflandırma yapısı geliştirilmiştir. Bu yapıda, 92% ve üzeri sınıflandırma doğruluğu elde edilmiştir. İkinci olarak, ML regresyon teknikleri deneysel olarak toplanan bir veri setine uygulanmış ve performansları k-katlı çapraz geçerlik yöntemiyle değerlendirilmiştir. Yapay sinir ağları ve karar ağacı regresyonu, çıktıların çoğunluğu için R^2>0.9'a ulaşarak literatürdeki diğer modelleme yaklaşımlarından daha iyi performans göstermiştir. Son olarak, biyokütle gazlaştırma sürecinde bir zaman serisi modelleme yaklaşımı uygulanmış ve tüm çıktılar için $R^2 > 0.98$ elde edilmiştir. Ayrıca, bir model tahmini kontrolör, biyokütle gazlaştırma işleminin çıkış konsantrasyonlarını kontrol etmek amacıyla tasarlanmıştır. Tasarlanmış MPC pratik senaryolarda test edilmiş ve tüm senaryolar için tatmin edici bir performans sergilemiştir. Aynı zamanda pratik deneysel verilere yüksek uyum göstererek pratik kullanılabilirlik potansiyelini daha da güçlendirmiştir.

1. INTRODUCTION

1.1 Motivation

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 tonnes of oil equivalent (mtoe) in 1965 to 13,511 mtoe in 2019 [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, biomass gasification caught attention as an efficient and cleaner way to produce energy. Biomass is the general name of all non-fossilized biological material obtained from living or recently living creatures. It refers to animal and vegetable-based products which make biomass abundantly available and renewable resource of energy all around the globe [3,4]. Gasification is a thermochemical process that converts organic or fossil fuels to combustible gases, i.e., a mixture of carbon monoxide, carbon dioxide, hydrogen, methane, light hydrocarbons and char by reacting the fuels with oxygen and steam in high temperatures [5]. This gaseous mixture referred to as “syngas” which can be further processed to generate heat and electricity [6]. Thus, biomass gasification is one of the subtypes of gasification processes that use biomass as a fuel. It can utilize widely available biomass such as forest waste, municipal solid waste and agricultural waste which makes it a highly environment-friendly chemical conversion process [7]. It has been known that a series of chemical reactions take place simultaneously during the biomass gasification process [8]. Thus, control and optimal syngas production via biomass gasification has been accepted as a challenging task due to its sensitivity to small changes in many parameters of the

process such as temperature, equivalence ratio and characteristics of the fed fuel as well as due to the highly nonlinear dynamics of the chemical reactions that occur during the process [9–11]. However, it has been show that if the parameters such as temperature, equivalence ratio, biomass feed rate and air/fuel ratio are selected before the process successfully with respect to the type of biomass that will be used, the maximum quantities of desired products can be obtained [3]. Therefore, a model that can describe and/or predict the dynamic behavior of production relative to the input parameters (operation conditions) of the process is well needed for optimal syngas production. The effect of operating conditions on syngas composition has been evaluated by a few numbers of kinetic mathematical models [12,13]. However, these models that evaluate homogeneous and heterogeneous reactions require extensive efforts to formulate kinetic equations and transport (heat, mass and momentum), where finding the solution is a time consuming and computationally expensive [12]. Another approach to model biomass gasification process is the use of thermodynamic equilibrium models. thermodynamic equilibrium approach consists of two equilibrium methods referred to as stoichiometric and non-stoichiometric approaches. The non-stoichiometric method is based on Gibbs free energy minimization [14–16]. In stoichiometric models, the equilibrium is determined using equilibrium constants for particular reactions such as water gas shift and methanation for the stoichiometric approach [17–20]. However, the equilibrium condition in the gasifiers is never achieved for both techniques. Thus, the high number of assumptions made in equilibrium models and extreme computational requirement in kinetic models significantly reduced these modeling approaches' practical usability and reliability [21]. Naturally, control of the output syngas production for optimal energy production is not studied due to the practical/computational limitations of these conventional methods. In order to overcome the drawbacks of the mentioned modeling approaches, a new approach was needed and machine learning-based methods has shown to be outstanding candidate for such purpose. Machine learning (ML) algorithms have been widely used to develop estimation models to understand and solve a various type of complex problems encountered in different disciplines of science [22]. One of the most useful attributes of ML techniques is that they do not require any mathematical definition of the phenomena involved in the process, and

therefore limitations caused by the need for high number of assumptions in the equilibrium models are fundamentally eliminated [23]. As such, ML models usually have low computational expense once they are trained which eliminates the extensive computational and time requirement of the kinetic modeling approach [24]. Thus, mentioned advantages shows a great potential of ML techniques for their use in biomass gasification.

Aside from the mentioned challenges, there is another obstacle for optimal and clean energy production via biomass gasification which is the classification of the fuel used. Type of the biomass, and fuel in general, to be used is a critically important information in biomass gasification as well as in almost all energy applications [25]. It is also necessary to consider the effect of energy conversion process on air pollution as well as impacts on water and soil [26]. For example, manufactured biomass generally contains higher amounts of heavy metals (e.g., Cu, Cr, Ni and Zn), while coal-type fuels contain more sulfur [27]. However, the recovered fuels can be heterogeneous mixtures generated from different types of solid fuels [28]. As a result, a specific classification is needed for the research of thermal conversion of solid fuels and it is essential to plan to preprocess and improve the production of power [26]. Therefore, a highly accurate classification framework to classify biomass from other fuel-based resources as well as to distinguish different types of biomass from one another is a need for optimal syngas production via biomass gasification and to minimize the environmental damage caused by it. Furthermore, ML methods are also shown to have capability at classification tasks in various problems [29]. Thus, the great potential of ML on modeling of biomass gasification as well as on the classification of solid fuels establishes the motivation of this thesis.

1.2 Related Work

1.2.1 Machine learning-based approaches to biomass gasification

With the mentioned advantages of the ML algorithms and widely available computational tools for implementing these algorithms, researchers started using ML methods to develop prediction models for the biomass gasification process. The most suitable approach for this modeling task is referred to as 'regression'. Regression is a branch of supervised ML algorithms that are used to predict continuous type output

variable(s) (dependent variables) in a data set by using one or more features (independent variables) that include observations (samples) which ideally contains enough information about the output variable(s) [30]. Even though the term 'regression' has been used for many years in statistics, it has been interchangeably mentioned as a set of ML algorithms in the computer science literature [31]. Because the syngas compositions of the biomass gasification are continuous type variables (mostly volume percentage), regression algorithms, mainly artificial neural networks (ANN), were usually go-to methods used during the development of prediction models. Guo et al. [32] employed ANN modeling to predict the product yield and gas composition of biomass gasification in an atmospheric pressure steam fluidized bed gasifier. 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 was insufficient to explain and prove the suitability of the proposed method for biomass gasification problem. Arnavat et al. [33] 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, H_2 content, CH_4 content, CO_2 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. They predicted each output variable with $R^2 > 0.95$ and showed the impact of each feature for corresponding outputs by utilizing weight and bias variables in the trained ANN model. 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. [34] have proposed a set of methods to select and apply most suitable ANN architecture 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 splat 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 [35]. Thus, the best ANN architecture selection according to proposed method might be misleading due to repeated random splitting on the small dataset. Brown et.al. [36] used nonlinear regression and ANN methods to predict product compositions of biomass gasification using feedstock compositions, ER and reaction temperature variables. They have tried several hidden layer sizes in their feedforward ANN architecture and conclude that ANN modeling can outperform equilibrium and kinetic modeling approaches when applied properly. In Mutlu et.al. [6], authors predicted the syngas composition released in the biomass gasification process by employing support vector machines and random forest algorithms. They used gasification temperature, ultimate analysis results, ER and fuel ratio variables and obtained lower root-mean-square error (RMSE) values compared to equilibrium and kinetic modeling approaches.

1.2.2 Classification of solid fuels

In the literature, studies based on the classification of fuels usually conducted to explain certain characteristics of the fuels. For example, Van Krevelen diagram which is a graphic method to characterize the source and maturity of organic matter by plotting molar H/C ratios against molar O/C ratios. It is often used in the literature to separate different classes of biomass to analyze their heating values [37–39]. In Zhou et al. [40], authors classified MSW components using a cluster analysis method according to the proximate and ultimate analyses and heating value results, as well as thermogravimetric (TG) characteristics. The classification groups include vegetables including banana peel, starch food, orange peel, wood waste, printing paper, cellulose, PVC, PET, PE/PP, PS, and rubber. Furthermore, the classification of liquid fuels, i.e. algae, rocket, diesel, and jet fuels are also introduced to the literature. In Ross et al. [41] they classified five macroalgae from the British Isles; Fucus vesiculosus, Chorda filum, Laminaria digitata, Fucus serratus, Laminaria Hyperborea, and Microcystis pyrifera from South America. The macroalgae have been characterized for proximate and ultimate analysis, inorganic content, and

calorific value. Their result has also been presented in terms of a Van Krevelen diagram. In Rearden et al. [42], a fuzzy rule-building expert system (FuRES) was used as a multiclass classifier for the two-way gas chromatograms of fuels, including rocket, diesel, and jet fuels. Wang et al. [43] they demonstrated the feasibility of using two-dimensional correlation coefficient mapping to classify gas chromatograms of aviation fuels under environmental hazards.

For the case solid fuel classification explicitly, there are no studies that focus on developing a such classification framework. Expert opinion is a usually used and a practical approach to classify the type of the fuel, but it is prone to be misleading due to the human-error [44]. More scientific approach is to conduct ultimate analysis and decompose the material into elementary contents, i.e., C, H and O which are unique to the material (if the tools used can measure with enough precision). Although the results obtained from ultimate analysis can be used to classify the type of corresponding fuel, one has to search for a database and/or a look-up table to find the class which the fuel belongs to. When the expensive equipment requirement of ultimate analysis is also taken into account, fuel classification with using ultimate analysis may become a frustrating and inefficient process. On the other hand, proximate analysis expresses the material in terms of its fixed carbon, volatile matter and ash contents. It is widely used by researchers due to its lower equipment cost compared to ultimate analysis [45]. Although the results obtained from proximate analysis may not as deterministic as they are in ultimate analysis when the fuel characteristics are concerned, proximate analysis have been used in various modeling and prediction studies [46–49]. Thus, proximate analysis has shown to be an efficient and useful method when characteristics of the fuels are concerned.

1.3 Research Objective

When the studies in the literature about both ML-based approaches to biomass gasification and classification of solid fuels are concerned, one can recognize the shortcomings as follows:

1. ML-based studies for modeling of biomass gasification almost always focus on ANN, other methods are rarely employed.

2. Performance evaluations and generalization capabilities of the proposed methods are not deeply discussed. Possible overfitting problem and randomness effect is not investigated by cross-validation techniques.
3. Biomass gasification is not treated as a "process" during the applications of ML on it. All of the developed models are time-independent. When the time dependency, effect of the previous states on the current one, is concerned.
4. Even though it is the one of the important factors for preparation for biomass gasification process, classification framework for the solid fuels is not proposed nor studied in the literature.

Thus, the current state of the literature leaves a space for improvement which this thesis aims to fill. The goals of this thesis can be expressed in three parts. First is the development of a ML-based classification framework to accurately classify various type of fuels that includes both fossil-based and biomass types. This framework benefits not only the biomass gasification related applications but many more energy applications that utilizes fuels. Second goal is to create predictive models for biomass gasification process by employing several ML-based regression methods and the use of appropriate metrics for unbiased performance evaluation. Last goal is the development of a time dependent model of biomass gasification with ML-based time series modeling techniques and a model predictive controller design that can control the certain outputs of the process in practical manner. And it has been ultimately aimed that to use power of ML to help researchers and engineers around the globe to help produce energy via biomass gasification more cleanly and more effectively.

1.4 Thesis Outline

The remainder of the thesis is structured as follows:

Section 2 proposes a ML-based framework for solid fuel classification, for this purpose, a data set is collected from the literature and was grouped into four classes, i.e., coals, woods, agricultural residue and manufactured biomass with their respective proximate analysis results. Then, K-nearest neighbor, support vector machine and random forest machine learning classifiers are employed to develop classification models. Furthermore, hierarchical classification approach is taken to combine each classifier's advantages with integration of expert opinion to create a complete and highly accurate classifier framework which can classify an unknown fuel in to one of the four categories by just utilizing proximate analysis results.

Section 3 explains the development of regression models by employing i.e., polynomial regression, support vector regression, decision tree regression and multilayer perceptron to predict CO, CO_2, CH_4, H_2 and HHV outputs of the biomass gasification process. Experimentally collected data set is used in this section. Feature extraction and application of Principal Component Analysis (PCA) technique to the extracted features is demonstrated to prevent multicollinearity and to increase computational efficiency. Performances of the proposed regression methods are evaluated with k-fold cross validation accompanied by the deep discussion about the results. Results of the previous studies on the subject are also compared to the ones obtained in this section.

Section 4 proposes a time dependent regression model for biomass gasification that can describe and predict outcomes of biomass gasification using non-linear autoregressive with exogenous neural networks (NARXNN) with the use of the knowledge extracted from **Section 3**. Secondly, a model predictive controller (MPC) design is proposed in order to control a certain output variable at a desired state. Moreover, the designed controller is challenged in practical scenarios such as maximum hydrogen production to test its usability in practical applications.

Section 5 summarizes the results of previous sections and offers recommendations for future research.

2. CLASSIFICATION OF SOLID FUELS WITH MACHINE LEARNING

In this section of the thesis, a set of classification models is proposed to classify four different types of fuels, i.e., coals, woods, agricultural residues and manufactured biomass by employing three machine learning based methods, i.e., random forest, K-nearest neighbor and support vector machines. These methods are selected due to their strictly different way of "learning" which led to the exploration different sides of the problem. K-nearest neighbor algorithm which is used to classify unlabeled data according to the closest data points (neighbors) is one of the most extensively used classification methods due to its easy-to-implement algorithmic structure and effectiveness for various data-related problems [50]. As a more sophisticated and complex algorithm, support vector machines is a classification method that separates classes in the data set as wide as possible with the use of hyper-planes. As a result of its unique approach for classification, it is extensively used especially for image and text recognition problems [51]. Random forest, on the other hand, is a method that built upon decision trees which is another classification algorithm. In decision trees classification is performed with recursively answering 'yes' or 'no' questions [52]. A random forest contains multiple decision trees that are trained with different sections of the data. It makes the final classification by the selecting most commonly occurred class in the predictions made by decision trees [53].
This feature of random forest enables it to look at different perspectives of the problem and to decide the final prediction with more unbiased fashion [54]. Furthermore, the use of classification algorithms to predict all classes in the output vector is referred to as flat classification [55]. Even though this approach is viable for many problems, it ignores any kind of hierarchical structure between classes that are aimed to be predicted [56]. On the contrary, hierarchical classification is a subset of classification approaches where the output classes are grouped depending on their similarity or differences, and classification is performed with several stages starting from the arbitrarily created groups and continues until original classes are predicted [57]. Even though this approach is well-defined and have been used in many

computer science applications in the literature, it is not well studied especially on the interdisciplinary application of machine learning [58].

The data set used in this section contains the proximity analyses and the name of corresponding fuels and is collected from multiple studies in the literature. After collection and merging of the data set, the study is conducted in two phases. In the first phase, proposed methods are separately trained to classify fuels. In the second phase, hierarchical classification is employed to combine each individual method's strengths. Moreover, this method allowed to integrate expert opinion during the training of the models to obtain improved stability and accuracy. 10-fold cross validation technique is used to evaluate each models' accuracy and generalization performance in an unbiased manner. Results obtained from each phase are discussed and source code for proposed hierarchical classifiers is provided for researchers to use (https://github.com/furkanelmaz/SolidFuelClassification).

2.1 Dataset Acquisition and Analysis

2.1.1 Collection of the data set

Dataset used in this section has 585 samples and is collected from various reliable studies in the literature as well as from Phyllis. All samples contain fixed carbon, volatile matter, ash contents, i.e., proximate analysis results, and the corresponding name of the fuel. Because trying to develop a classifier to classify each fuel individually is not a realistic approach, the fuels are categorized into coals, woods, agricultural residues (AR) and manufactured biomass (MB) classes as shown in Table 2.1.

Fortunately, dataset contains three features and single output column with four distinct classes, thus, it can be plotted it in 3D surface (Figure 2.1). As one can see in Figure 2.1, there are considerable overlaps and no strict distinction between classes at least from plain eye observation which indicates the requirement of further analyses. Also, one must not confuse categorization of raw fuels and categorization in hierarchical classification mentioned earlier, categorization here is to obtain classes that is aimed to be predicted at the end and have no connection to hierarchical classification.

Table 2.1 Categorization of raw fuels.

Class Name	# of Samples	Elements
Coals	94	coals, charcoals, chars
Woods	251	wood, shell, pruning
AR	167	seed, husk, leaves, grass, bark, straw, stalk
MB	73	municipal solid waste, RDF, sludge, briquettes

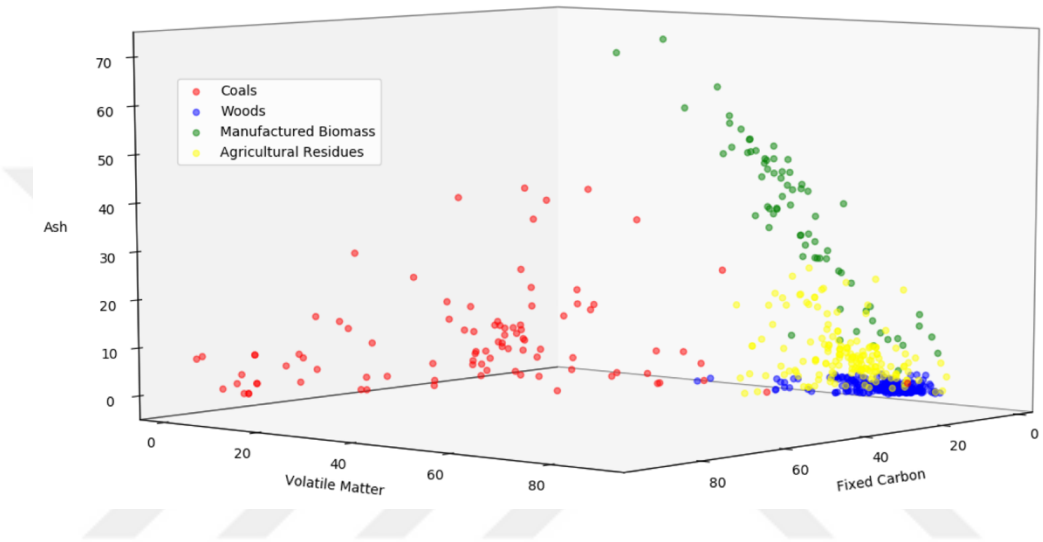


Figure 2.1 3D plot of the data set.

2.1.2 Statistical analyses

In order to have better understanding of the dataset and to find relationships between classes, several statistical tests are conducted. Means and standard deviations of each feature for each class are calculated to explore characteristics of the classes (Table 2.2).

Table 2.2 Means and standard deviations of classes for each feature.

	Coals		MB		AR		Woods	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Fixed Carbon	58.62	17.12	9.19	3.46	18.38	4.73	17.86	3.79
Volatile Matter	30.48	15.27	43.17	13.48	73.84	5.89	80.71	4.18
Ash	10.89	9.14	33.63	15.27	7.76	5.24	1.44	0.94

According to Table 2.2, one can deduce following statements:

1. Coals tends to have higher fixed carbon content compared to all others
2. MB tends to have higher ash content compared to all others
3. AR and woods tend to have higher volatile matter content compared to coals and manufactured biomass
4. AR and woods tend to have similar fixed carbon content

Although deductions made were based on means and standard deviations, they are not enough to prove a certain characteristic of a class is different from another. For this reason, Mann–Whitney U non-parametric statistical test is employed to justify these deductions [59]. Like many other statistical tests, Mann-Whitney U assumes two given lists came from similar distribution and there is no statistical difference between them (null hypothesis). Then a p-value ($0 \leq p - value \leq 1$) is outputted after the test is conducted, if p−value is less than 0.05, hypothesis gets rejected which concludes that there is a statistical difference between two given lists. Otherwise, hypothesis is accepted. Mann-Whitney U test is conducted for all classes and all binary combination of the classes. In order to keep things simple, only p−values for the deductions made are included in Table 2.3. One must note that, different deductions can be made with using Table 2.2 and results of the Mann-Whitney U test. Given deductions are based on prior experience and what is mentioned as expert opinion throughout this section.

Table 2.3 Mann-Whitney U test on selected classes and group of classes

Compared Groups	**Coals - MB**	**Coals - AR**	**Coals - Woods**	**AR - Woods**	**AR&Woods - Coals & MB**	**MB - Coals**	**MB - AR**	**MB - Woods**
Compared Feature	*Fixed Carbon*				*Volatile Matter*	*Ash*		
p-values	9.97E-29	7.56E-40	2.64E-45	0.122	3.8E-66	1.23E-18	3.73E-29	2.96E-39

As one can see from Table 2.3, distribution of fixed carbon content of Coals is statistically different all other classes. And with its higher mean value in mind, one can assume first statement is justified. In the same way, ash content of MB is different and tend to be higher than the other classes which justifies second statement. For third statement, AR class is combined with Woods and Coals is combined with MB, and with resulting p-value, these two combined categories are statistically different from each other and AR & Woods category tend to have higher volatile matter content due to their higher mean values. For final statement, AR and

Woods tend to have similar fixed carbon content with p-value greater than 0.05, thus, it is not a distinctive feature between them.

2.2 Classification Methods & Performance Evaluation Techniques

2.2.1 K-nearest neighborhood

K-nearest neighborhood (KNN) is a primitive yet effective machine learning algorithm which is easy to implement and has low computational complexity compared to other complex machine learning algorithms [60]. KNN is based on a simple idea; unlabeled data can be classified as the most occurred class in k number of nearest labeled data points. This idea raises two fundamental questions; how can we measure the 'distance' between data points to determine whether a data point is near or not? What is the optimal number k? In KNN, distance is usually calculated with geometrical distance metrics such as Euclidean and minkowski metrics [61]. In this section, minkowski metric, which is defined in Equation 2.1, is used.

$$D(x, y) = \sqrt[p]{|x - y|^p} \tag{2.1}$$

Euclidean distance is a special case of minkowski when the p is equal to 2. The distance is calculated between a labeled sample and unlabeled sample for each feature column individually, then the results are summed to obtain final distance value. This process is repeated between each labeled sample and the unlabeled sample to calculate all distance values, thus, one can predict the class of unlabeled sample with using the class information of k number of data points which have the least distance to the unlabeled data. Selection of number k is highly dependent on the data set used, and usually determined by trying different number of integers and by evaluating their performances [62].

2.2.2 Support vector machine

Support Vector Machine (SVM) is a widely used machine learning algorithm created by Vapnik and Cortes [63]. Although it was originally proposed as a binary classifier, due to its enormous success especially in text and image recognition, it is twisted and have been also used in multi-class classification and regression problems [64]. SVM aims to find not only feasible but an optimal hyperplane to separate given

classes which is also known as large margin classification. This feature distinguishes SVM from other machine learning algorithms because it makes SVM fundamentally resistant to overfitting where others prone to overfit more easily [65]. Decision condition for two arbitrary classes 1 and -1 and objective function J which will be minimized during the training phase are given in Equation 2.2 and Equation 2.3, respectively [63].

$$Prediction = \begin{cases} 1, & \text{if} \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0 \\ -1, & \text{otherwise} \end{cases} \tag{2.2}$$

$$J = \frac{1}{2} * \|w\|^2 - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{x}_\mathrm{j} \cdot \mathbf{w} + b) - 1] \tag{2.3}$$

where, m is number of samples in the data set, $\boldsymbol{x_i}$ is the unlabeled feature column vector, $\boldsymbol{x_j}$ is the labeled feature column vector, y is the label (output) vector. $\boldsymbol{w}$, b and $\boldsymbol{a}$ are weight vector, constant term and Lagrange multipliers which will be determined in training process, respectively. Equation 2.3 is called primal formulation of SVM. Even though it is possible to train an SVM model for linearly separable simple problems, true power of SVM yields on kernel method. With kernels, input data can be mapped into higher dimensional space with the help of a kernel function. Thus, linearly inseparable input data can become linearly separable in higher dimension. In order to use kernels, primal form must be converted to dual form as given in Equation 2.4 [66].

$$\begin{gathered} J = \sum_{i=1}^{m} a_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ s.t. \\ \sum_{i=1}^{m} \alpha_i y_i = 0\,, \\ \alpha_i \geq 0, \alpha_i \leq C, \\ K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j \end{gathered} \tag{2.4}$$

where, C is the regularization parameter which controls the violation of large margin classification, $K(x_i, x_j)$ the kernel function and in the original form, it is equal to dot product of x_i and x_j which is known as linear kernel. In order to map the inputs to higher dimension space, there are several kernel functions that can be used, radial basis (Equation 2.5) and polynomial (Equation 2.6) kernels are the most commonly used and what have been used in this section.

$$K(x_i, x_j) = e^{\frac{\|x_i - x_j\|^2}{2\sigma^2}} \tag{2.5}$$

$$K(x_i, x_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + c)^d \tag{2.6}$$

Where σ is a free parameter which configures sensitivity to differences in feature vectors, d is the order of the polynomial. After selection of kernel and convex optimization of Equation 2.4, one can obtain the non-zero α values (support vectors), calculate w and b variables by using Equation 2.7 and Equation 2.8, respectively. Then unlabeled data can be classified with conditions given in Equation 2.2.

$$w = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i \tag{2.7}$$

$$b = \frac{1}{N_s} \sum_{i \in S} y_i - \sum_{j \in S} \alpha_j y_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j) \tag{2.8}$$

where, S corresponds to the indices of the support vectors in the α, N_s is the number of support vectors.

2.2.3 Random forest

Random Forest (RF) is an ensemble type machine learning method built upon Decision Tree (DT) algorithm [67]. It uses multiple DT models and merges them to make more accurate and stable predictions [68]. In DT, the tree is grown by recursively determining the most decisive values in input space with using a measure of randomness such as entropy or gini index [69]. Thus, a tree of binary split nodes is grown to classify unlabeled data. Although DT is useful in various machine learning problems, overfitting and unstability are commonly encountered problems [70]. There are several techniques like pruning to overcome overfitting but they are less reliable compared to RF algorithm [71]. Although, there are several ways of implementing RF, the most common "bagging" method is focused in this section. Bagging is an abbreviation for "bootstrap aggregating". For demonstration, let us assume we have a data set which has m number of samples and n feature columns, bootstrapped data set is created by inserting randomly selected sample from the original data set (with replacement) until the number of samples in bootstrapped set is equal to m [72]. Number of bootstrapped sets will be created is equal to the number of DT that will used in RF. After that, each DT will be trained with different bootstrapped set, but each best split during the training phase will be determined by using k number of randomly selected features where $k \leq n$. Moreover, an unlabeled sample will be classified by "aggregating" DTs, in other words, prediction will be made with majority vote of the trained DTs. With using a bagging technique, RF algorithm is able to look "different perspective" of the data set, thus, becomes more resistant to overfitting and creates more stable prediction model in return for higher computational cost.

2.2.4 Hierarchical classification

Large amount of classification studies in literature is conducted by using flat classification. Flat classification is conventional approach where the classifier is trained to classify all classes in a data set [73]. Although, this approach has helped researchers to solve various types of problems successfully, it ignores any kind of relationship between classes. On the other hand, hierarchical classification creates prediction models in multiple hierarchically structured stages with using one or more

classifiers [58]. Idea of hierarchical classification ranges broadly from simple classifiers to the deep learning applications. The approach that is used in this section local classification which also known as top-down approach in the literature [74]. In top-down approach, c number of classes in the dataset are categorized into t number of categories with respect to "similarities" between classes, where $t \leq c$. Created categories become new classes for the classifier used in upper level of the hierarchy which can be considered as parent–child relationship. In the training phase, each parent classifier learns to distinguish its child categories individually with no connection to other classifiers. In the testing phase, prediction of an unlabeled data starts with the classifier at highest level of the hierarchy (root node), it classifies the data as one of its child categories and corresponding classifier continues this procedure until a one of the original class in the data set (leaf node) is reached. There are several advantages of hierarchical classification to the flat classification, one of them is to ability to merge multiple classifiers for the same problem. In the same data set, one classifier may outperform other one for predicting a certain child category, this situation may reversed for another category, thus, increase in overall accuracy can be observed with combination of classifiers compared to selecting one algorithm to classify all classes [75]. Secondly, hierarchical classification allows researcher to explore and analyze the problem more deeply due to simplification of the problem at each parent–child stage. Also, one can use algorithms such as logistic regression, which is not suitable for multiclass problems, in different levels of the hierarchy. Moreover, a different subset of features can be used rather than using all features during the training of the classifiers, thus, expert opinion can be integrated at each level of hierarchy which also can increase overall accuracy and stability. With the given advantages, hierarchical classification approach is also employed to develop a prediction model in this section. In order to visually demonstrate the way how the hierarchical classifier is developed in this section, the flowchart given in Figure 2.2 is created for an arbitrary number of classes grouped into 2 groups.

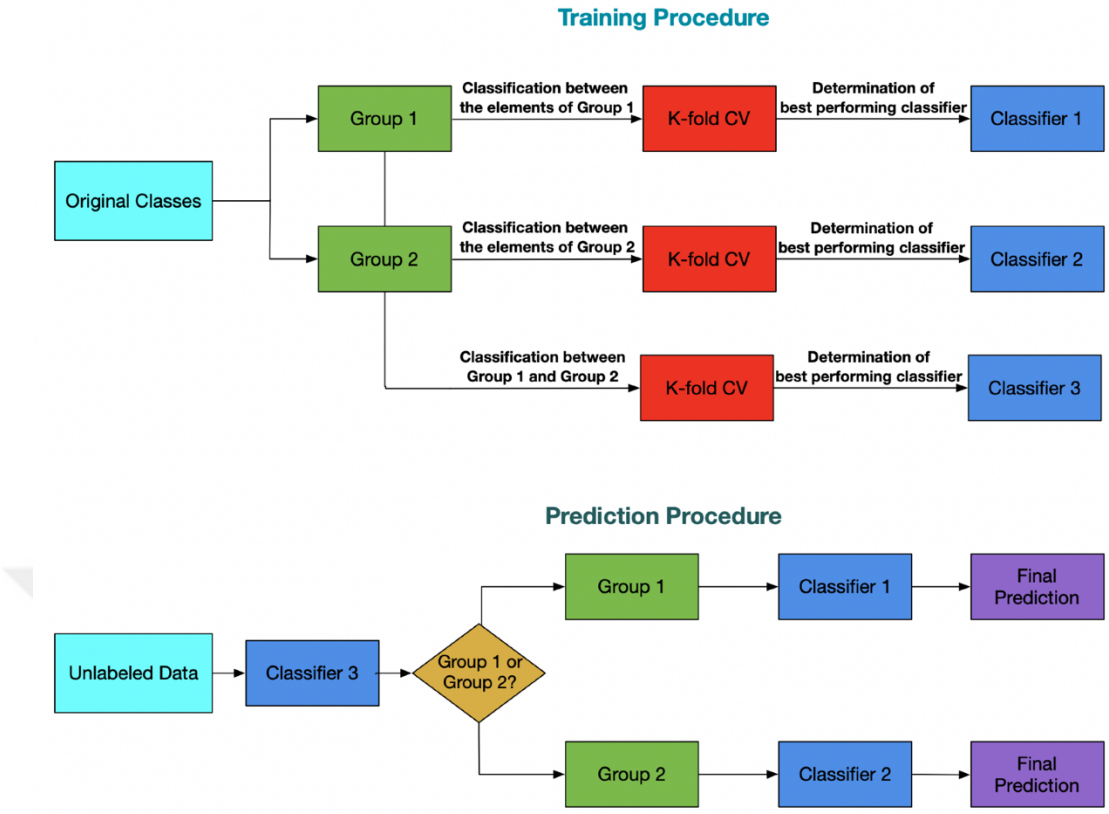


Figure 2.2 Flowchart of a hierarchical classifier design

2.2.5 K-fold cross-validation

Cross validation is extremely important procedure to evaluate generalization performance as well as the stability of a prediction model [35]. K-fold cross-validation (KCV) is popular technique due to its use of all the samples in the data set and its viability whether the data set is relatively small or not [76]. KCV splits the data set into k number of equally sized folds. Then, $k-1$ number of folds is used to train the model and the trained model is tested on the remaining one. This process is repeated k times to use all folds for testing once and all predictions made on the test folds are merged. Cross-validation accuracy is calculated between merged predictions and original data set by using appropriate error metrics depending on type of the problem. Because training and testing are performed on different folds in each iteration and accuracy is calculated with using whole data, KCV shows the true prediction ability of the candidate model in unbiased manner [77]. Moreover, each fold is used $k-1$ times for training in KCV, thus, one can also look at training accuracies by averaging $k-1$ performances of each fold to check for stability

problems [78]. In this section, KCV is used to evaluate training and testing performances of the models proposed and k number is chosen as 10 due to its reliability shown in previous studies in the literature [79]. Evaluation scheme of 10-fold cross validation is illustrated in Figure 2.3.

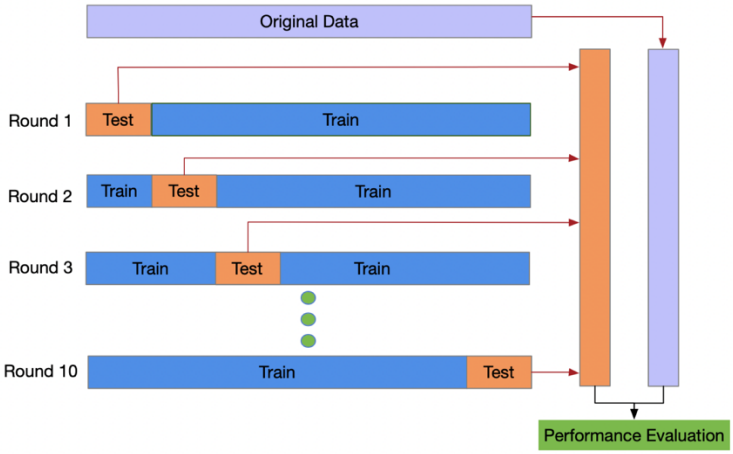


Figure 2.3 Visual demonstration of 10-fold cross-validation

As one can see in Figure 2.3, predictions of each fold are stored until all ten rounds are completed, then performance evaluation is conducted between all of the stored predictions and original data [80]. Moreover, in this section, procedure illustrated in Fig. 1 is repeated ten times with different random number generator seeds and the means and relative standard deviations for resulting ten iterations are calculated during the performance evaluations of the employed methods. With the repetition of 10-fold cross validation for ten times, it is easier to check the dependency to the randomness of each proposed method [81]. Because these elements in ten equal-sized folds are selected randomly, random number generator seed is changed in Python at each iteration to check whether the candidate model can predict target classes with the same success regardless of the different random splits. The mean and relative standard deviation of these ten repetitions of 10-fold cross-validation is calculated to show if the model is successful at predicting target classes as well as how resistant is the model to randomness factor which is referred to as stability in remainder of this section. Moreover, because 10-fold cross validation is performed 10 times with different random number generator seeds, mean and relative standard

deviation of iterations is also provided in tables provided in the further subsections are in the form of $mean \pm relative\ standard\ deviation$. Also, confusion matrices are plotted for more in-depth analyses. A confusion matrix is a 2D plot where one of the axes is the actual labels and other is the predicted labels. This plot shows how many samples are labeled correctly and incorrectly for each class. Percentage values are also added in these plots for better clarity.

2.2.6 Performance metrics

During the performance evaluations of the proposed models, classification accuracy (Accuracy) (Equation 2.9), Precision (Equation 2.10), Recall (Equation 2.11) and F1-Score (Equation 2.12) metrics are calculated.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\#ofTruePredictions}{\#ofTotalPredictions} \tag{2.9}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.10}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.11}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.12}$$

where, a TP (true positive) corresponds to a correct identification of a test sample from a particular class. FN (false negative) means that the prediction is not the class that the test sample belongs to, which is the target class. TN (true negative) happens when predictions based on samples from classes which are not the target class, and they are classified as non-target class. FP (false positive) happens when the sample is predicted wrongly as the target class [82]. For instance, if a medical doctor wants to

predict if a patient has cancer or not by examining tomography of the patient, then a TP happens if the patient has cancer (cancer=positive in this case) and the doctor also diagnoses the patient as cancer. On the contrary, if the patient does not have cancer but the doctor diagnoses him/her with cancer, then this would be a FP. Also, if the patient has cancer but the doctor diagnoses him/her without cancer, then this would be a FN. One should note that in both cases, FP and FN, the prediction is wrong. And finally, the true negative (TN) happens when the patient does not have cancer and the doctor diagnoses him/her as non-cancer or healthy. Classification accuracy shows overall prediction capability of the model and is calculated as the proportion of number of correctly predicted labels to the number of the prediction made. Precision and recall metrics judge the prediction model from different perspective by only utilizing TP, FP and FN scores. F1-Score uses both precision and recall metrics to obtain a final score. All metrics range between 0 and 1, where 1 corresponds to the perfect score [83].

2.3 Results and Discussion

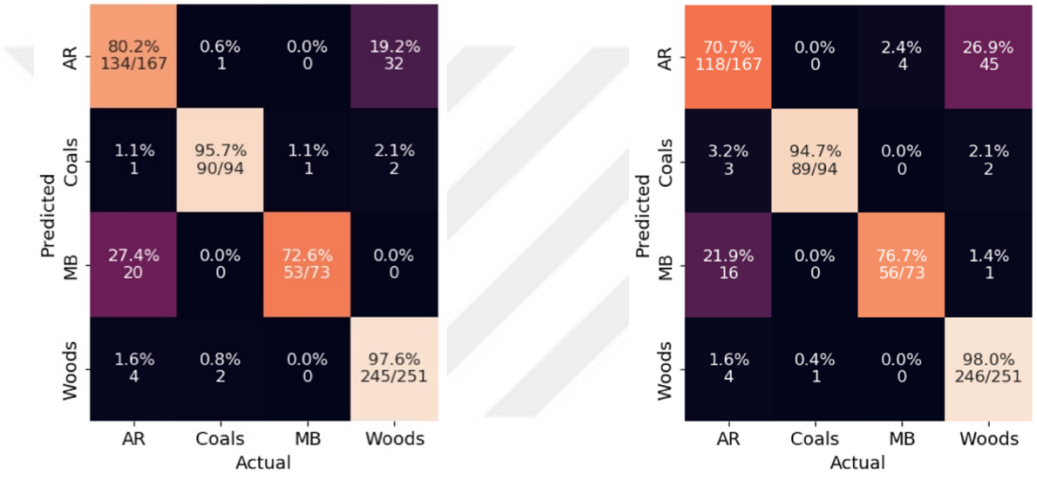
2.3.1 Flat classification

Classifiers proposed in this paper have hyper-parameters such as number of trees in RF, number of neighbors in KNN or regularization parameter in SVM which can dramatically increase or decrease the performance of the predictions. In order to show true potential of each method, different hyper-parameters for each method have been tried. Results with most successful hyper-parameters are given in Table 2.4 and their corresponding confusion matrices are given in Figure 2.4. Hyper-parameter with highest test score is selected as best, if one or more of them had equal test scores, one that had least relative standard deviation is selected. One must note that percentage accuracies, number of correctly or incorrectly labeled classes given in confusion matrices are also mean values of 10 iterations of 10-fold cross validation

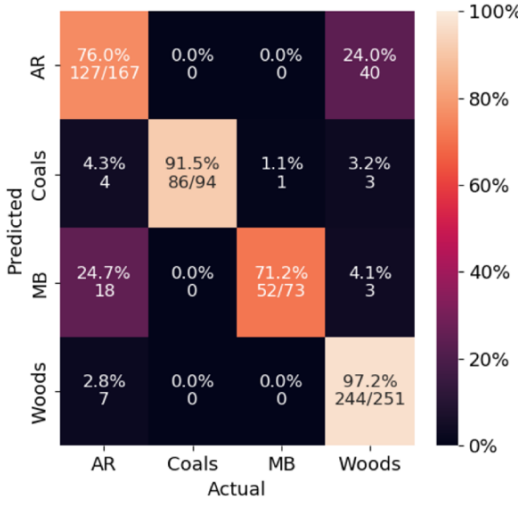
evaluations.

Table 2.4 Performance evaluation results of best performing flat classifiers

	Classification Accuracy		Precision		Recall		F1-Score	
	Training	*Testing*	*Training*	*Testing*	*Training*	*Testing*	*Training*	*Testing*
KNN (# of Neighbors=10 Minkowski Order=1)	0.842 ± 0.08%	0.8228 ± 0.5%	0.842 ± 0.08%	0.823 ± 0.5%	0.842 ± 0.08%	0.8228 ± 0.49%	0.84 ± 0.1%	0.8213 ± 0.5%
RF (# of Trees=100 Maximum Depth=10)	0.988 ± 0.83%	0.901 ± 6.3%	0.989 ± 1.02%	0.9 ± 6.5%	0.98 ± 0.83%	0.9 ± 6.38%	0.98 ± 0.83%	0.9 ± 6.54%
SVM (Kernel=Quadratic (d=2) C=1)	0.841 ± 0.13%	0.838 ± 0.6%	0.842 ± 0.14%	0.839 ± 0.64%	0.841 ± 0.13%	0.838 ± 0.6%	0.84 ± 0.2%	0.836 ± 0.61%



(a) (b)



(c)

Figure 2.4 Confusion matrices resulting from best performing RF (a), SVM (b) and KNN (c) flat classifiers

Results shown in Table 2.4 can be considered somewhat successful. SVM and KNN methods were able to "understand" the dynamics of the problems to some degree and more promisingly developed models were stable because the relative standard deviation between cross-validation iterations were quite low. RF performed best when the overall accuracy mean is considered. On the other hand, RF also had the greatest relative standard deviation between iterations of 10-fold cross validation evaluation, this situation refers least stability and generalization performance. The main reason for this high accuracy and high variance behavior is the random feature selection during the growing of trees in the RF. When the distinctive features are selected, RF was able to predict high accuracy, otherwise accuracy drops accordingly and creates variance between iterations. This situation leads to another conclusion, usage of all features is not necessary and may even drop the accuracy of the model for some classes. And with flat classification it is not possible to determine which features are determinative to distinguish which classes. Even though flat classification results were not catastrophic, there is a space for improvement in overall accuracy and stability with hierarchical classification which is discussed next.

2.3.2 Hierarchical classification

There are several steps to follow while developing a hierarchical classification model. Firstly, a hierarchical structure between classes is created by using their "similarities" and "differences". Woods and AR tend to have larger amount of volatile matter compared to other classes as mentioned in Section 2.1.2. These two classes make up the first group "Group 1". Naturally, other two classes, i.e., Coals and MB make up the "Group 2". As a second step, distinctive feature(s) between the elements of each group are determined. For elements of Group 2, Coals have greater fixed carbon content while MB having greater ash content as inferences made in Section 2.1.2. Therefore fixed carbon and ash are optimal features to distinguish these two elements. For Group 1, fixed carbon is not an optimal choice as mentioned. Therefore, only volatile matter and ash features are used during the classification of these two elements. As a final step, three classifiers are determined to classify Group 1 and Group 2, as well as their respective elements. For each classifier, RF, KNN and SVM with different hyper-parameters are implemented, best results are given in

Table 2.5, Table 2.6, Table 2.7 for classification of Group 1 and Group 2, Coals and MB, Woods and AR, respectively.

Table 2.5 Performance evaluation for the classification of Group 1 and Group 2

	Classification Accuracy		**Precision**		**Recall**		**F1-Score**	
	Training	*Testing*	*Training*	*Testing*	*Training*	*Testing*	*Training*	*Testing*
KNN (# of Neighbors=5 Minkowski Order=3)	0.919 ± 0.02%	0.91 ± 0.31%	0.919 ± 0.03%	0.91 ± 0.31%	0.919 ± 0.02%	0.91 ± 0.31%	0.919 ± 0.03%	0.909 ± 0.31%
RF (# of Trees=100 Maximum Depth=8)	0.998 ± 0.58%	0.933 ± 2.37%	0.998 ± 0.58%	0.933 ± 2.46%	0.998 ± 0.83%	0.933 ± 2.37%	0.99 ± 0.58%	0.933 ± 2.46%
SVM (Kernel=Quadratic(d=2) C=0.01)	0.983 ± 0.07%	0.979 ± 0.31%	0.982 ± 0.07%	0.979 ± 0.32%	0.982 ± 0.07%	0.979 ± 0.32%	0.982 ± 0.05%	0.979 ± 0.33%

Table 2.6 Performance evaluation for classification of Coals and MB

	Classification Accuracy		**Precision**		**Recall**		**F1-Score**	
	Training	*Testing*	*Training*	*Testing*	*Training*	*Testing*	*Training*	*Testing*
KNN (# of Neighbors=3 Minkowski Order=2)	0.993 ± 0.03%	0.987 ± 0.06%	0.993 ± 0.02%	0.988 ± %0.0	0.993 ± 0.03%	0.987 ± 0.07%	0.993 ± 0.03%	0.988 ± 0.02%
RF (# of Trees=50 Maximum Depth=2)	0.995 ± 0.19%	0.984 ± 0.97%	0.995 ± 0.19%	0.984 ± 0.97%	0.995 ± 0.19%	0.984 ± 0.97%	0.995 ± 0.19%	0.984 ± 0.97%
SVM (Kernel=Quadratic(d=2) C=0.01)	0.989 ± 0.09%	0.98 ± 0.73%	0.989 ± 0.09%	0.981 ± 0.72%	0.989 ± 0.09%	0.98 ± 0.73%	0.989 ± 0.09%	0.98 ± 0.73%

Table 2.7 Performance evaluation for classification of Woods and AR

	Classification Accuracy		**Precision**		**Recall**		**F1-Score**	
	Training	*Testing*	*Training*	*Testing*	*Training*	*Testing*	*Training*	*Testing*
KNN (# of Neighbors=10 Minkowski Order=1)	0.925 ± 0.121%	0.907 ± 0.5%	0.927 ± 0.11%	0.909 ± 0.51%	0.925 ± 0.12%	0.907 ± 0.5%	0.925 ± 0.11%	0.907 ± 0.51%
RF (# of Trees=100 Maximum Depth=8)	0.985 ± 0.14%	0.948 ± 1.07%	0.986 ± 0.14%	0.949 ± 1.08%	0.985 ± 0.14%	0.948 ± 1.07%	0.985 ± 0.14%	0.948 ± 1.08%
SVM (Kernel=Quadratic(d=2) C=0.01)	0.931 ± 0.2%	0.91 ± 0.68%	0.932 ± 0.24%	0.911 ± 0.69%	0.931 ± 0.2%	0.911 ± 0.68%	0.93 ± 0.2%	0.911 ± 0.67%

According to Table 2.5, SVM had the greatest success to classify Group 1 and Group 2 due to its superior testing performance compared to other algorithms. In same fashion, as shown in Table 2.7, RF was the most successful algorithm for classifying elements of Group 2. One must note that, unlike what is used in flat classification, RF was trained with only boot-strapping rather than bagging algorithm in this part. The reason is the selection of features to be used beforehand, thus, random feature selection in RF is not required and even it might lead to underfitting [96]. For

classification of the elements Group 1, it was a quite simple task for all classifiers and all of them managed to achieve almost perfect prediction accuracy for both training and testing phases as shown in Table 2.6. With little to no accuracy differences between classifiers, KNN algorithm is selected for this stage due to its lower computational complexity compared to others when the sample size is quite low.

With the construction of hierarchical structure between classes, selection of classifiers and features to be used at each stage, proposed hierarchical classification model is developed. Training and testing procedures of the model are illustrated in Figure 2.5 and Figure 2.6, respectively. Performance evaluation and confusion matrix is given in TABLE and Figure 2.7, respectively.

Figure 2.5 Training procedure of the proposed hierarchical classifier

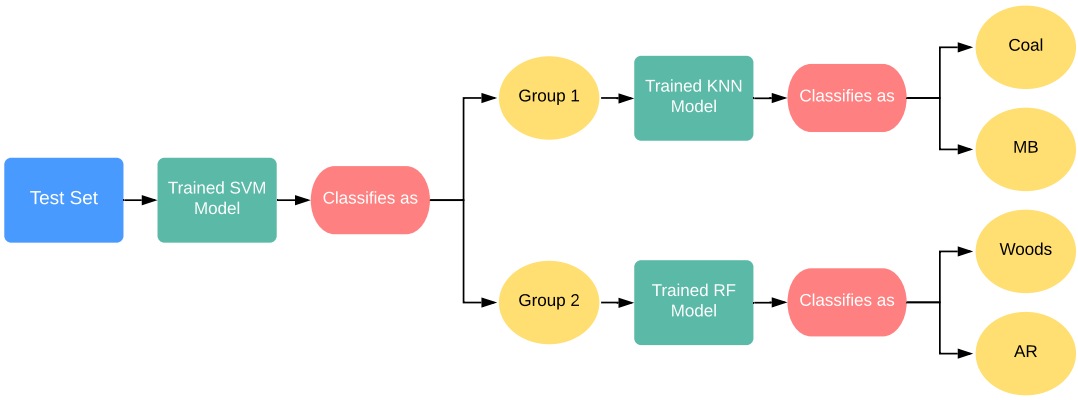


Figure 2.6 Testing procedure of the proposed hierarchical classifier

Table 2.8 Performance evaluation of the hierarchical classifier.

	Classification Accuracy		Precision		Recall		F1-Score	
	Training	*Testing*	*Training*	*Testing*	*Training*	*Testing*	*Training*	*Testing*
Hierarchical Classifier	0.963 ± 0.06%	0.923 ± 0.17%	0.963 ± 0.06%	0.923 ± 0.18%	0.963 ± 0.06%	0.923 ± 0.5%	0.925 ± 0.17%	0.962 ± 0.18%

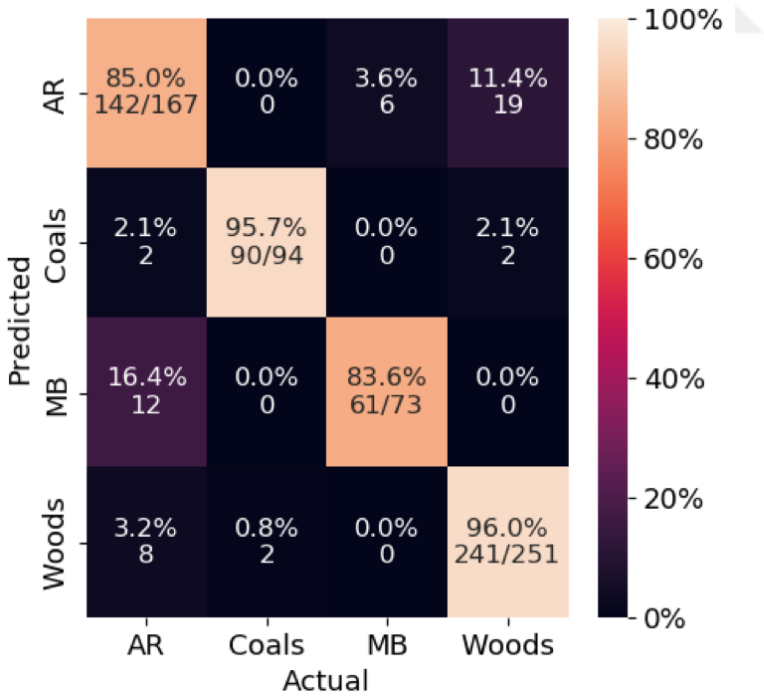


Figure 2.7 Confusion matrix resulting from hierarchical classifier

When Table 2.8 is examined and compared to Table 2.4, performance of hierarchical classifier is improved compared to other flat classification models especially in testing phase. Furthermore, training and testing scores of the model is quite close to each other which concludes that, model was not over-fitting and has good generalization performance. But there are few aspects one has to cover; firstly, are the results significantly better when the additional effort required to develop hierarchical model in mind? When flat classification models with KNN and SVM and hierarchical model are compared, there are reasonable performance difference. Both KNN and SVM was not able to distinguish each class as good as hierarchical model did when given accuracy metrics are taken into the account. For RF model, mean accuracies is quite similar to hierarchical model, but relative standard deviation between iterations are the main difference between those models. As mentioned in Section 2.3.1, random feature selection process in RF is a "two-edged sword" which can increase or decrease prediction accuracy depending on the randomness factor [84]. In hierarchical model, this problem was solved by selecting the features by using expert opinion and statistical analyses for each stage of the hierarchy. Therefore, hierarchical model grows into both stable and accurate prediction model as opposed to flat classification models.

In order to justify the results of the section statistically, Wilcoxon test which is an another non-parametric statistical to check is there a significant difference between performance of hierarchical model and flat classification models is employed [85]. In order to perform Wilcoxon test, test performances of best performing flat classifiers and hierarchical classifier are collected by changing random seed of 10-fold cross-validation splitting 100 times iteratively. Then, Wilcoxon test is used to obtain p-values with using these results (Table 2.9).

Table 2.9 Performance evaluation of the hierarchical classifier.

	SVM	**KNN**	**RF**
Hierarchical Classifier	$3.816E-18$	$3.789E-18$	$3.818E-18$

According to the p-values strictly less than 0.05, there is a statistically significant difference between hierarchical classifier and flat classifiers. With superior performance of hierarchical model in mind, it is safe to say that, for fuel classification problem, hierarchical prediction modeling outperforms flat classification approaches by both accuracy and stability perspectives when the feature and classifier selections are done correctly. This result also indicates that hierarchical structure between fuels exists and can be utilized in many different applications.

Even though there are no studies that focus on the classification of solid fuels, one can compare the performance of hierarchical classification framework to ones obtained in Rearden et al. and Wang et.al. where the focus is on the classification of liquid fuels [42,43]. In Rearden et al., authors reported 95 ± 0.3% accuracy for the classification performed between jet fuels. Moreover, in Wang et al., 90% accuracy is reported for the classification between Diesel/JP-4 binary mixture. These studies also mentioned that their respective results are satisfactory and proposed techniques can be utilized in practical applications. In this section, 92 ± 0.17% accuracy for the classification of solid fuels with a hierarchical classifier framework is obtained. When the accuracies of the previously mentioned studies are concerned, the proposed hierarchical classifier can be considered a successful and strong framework that can be utilized in practical applications.

3. PREDICTIVE MODELING OF BIOMASS GASIFICATION WITH MACHINE LEARNING

After the completion of the fuel classification which is an important procedure before the energy production via biomass gasification, next step is the development of a prediction model for the critical outputs of the biomass gasification process. As previously discussed in Section 1.2.1, there are several studies that focused on ML-based modeling of the biomass gasification. Nevertheless, these studies mainly deficient about the unbiased performance evaluations and excessively focused on the use of ANN. For this reason, in this section, four regression techniques, i.e., polynomial regression, support vector regression and decision tree regression are employed alongside ANN to predict five critical outputs, i.e., CO, CO_2, CH_4, H_2 and HHV of the biomass gasification process. Moreover, the experimentally collected data set is utilized during this section. The data set was collected by Gebze Technical University and the collection was independent from this thesis and only explained for clarification in Section 3.1. Furthermore, feature extraction is performed on the data set and PCA technique is applied to the extracted features to prevent multicollinearity and to increase computational efficiency. Performances of the proposed regression methods are again evaluated with 10-fold cross validation.

3.1 Dataset Collection

10 kW down draft fixed-bed gasifier provided by All Power Labs Inc is used during the data collection. Downdraft gasifiers enable solid fuels to be moved together with air in the downward direction, whereas the venturi ejector provides air suction that causes air entering the reactor via an airflow meter. The biomass is exposed to multiple processes in four different stages of the fixed-bed gasifiers: drying, pyrolysis, combustion, and reduction stages, in which some certain combination of complex heterogeneous and homogeneous chemical reactions take place. In order to filter the particles, cyclone separator is used to clean the produced syngas. Also, the packed bed filter is used to clean the remaining condensable materials. Lastly, the flue gas is burnt out by a swirl burner. The concentrations of CH_4, H_2, CO_2, CO and O_2 in the syngas measured by Wuhan Cubic Syngas Analyzer Gas board 3100P, a portable infrared syngas analyzer, simultaneously as heating value (calorific value) is

computed using Thermal Conductivity Detector and NDIR. The amount of heat that is released when a unit mass of fuel is fully combusted is defined as the higher heating value (HHV). It also includes the latent heat of vaporization of water. Temperatures measured using a group of 6 in-line K-type thermocouples with 8 mm diameter at 6 different heights inside the gasifier as T_0, T_1, T_2, T_3, T_4, T_5 during the experiments. Then were recorded real-time values of temperatures, pressures and flow rates using an Arduino based system illustrated in Fig.1. Equivalence Ratio (ER) expresses the amount of external air supplied to the gasifier, is computed using the recorded airflow rate values and it is a crucial operating variable in biomass gasification. Pinecone particles and wood pellets are used as the feedstock for gasification. Fallen mature pinecones collected from the Gebze Technical University campus in Kocaeli, Turkey. The wood pellets are purchased from a local manufacturer of wood pellets. During the data collection; ASTM E1755-01, ASTM E871-72 and ASTM D3175 methods are used to determine ash, moisture and volatile matter contents by using ash furnace, respectively. FC was determined by balance. Ultimate analysis results are obtained by using LECO Truspec CHNS Elemental Analyzer. Oxygen content is calculated with Equation 3.1, HHV is calculated with Equation 3.2 which is provided by Sheth and Babu [86]. The ultimate and proximate analysis of the woody biomass are given in Table 3.1. Thus, total number of 4826 data samples including temperatures, equivalence ratio values, concentration values and fuel flow rate were collected from five different experiments.

$$\mathrm{O\%} = 100 - \mathrm{C\%} - \mathrm{H\%} - \mathrm{N\%} - \mathrm{S\%} - \mathrm{Ash\%} \tag{3.1}$$

$$\mathrm{HHV} = 0.3536\mathrm{FC} + 0.1559\mathrm{VM} - 0.0078\mathrm{Ash} \tag{3.2}$$

Table 3.1 Physical and chemical characterization of the pine cone and wood pellet.

Physical analysis (mm)	Pine cone	Wood pellet
Width	8	7
Length	12	20
Proximate analysis (wt%)		
Moisture	9.6	12
Volatile Matter	77.8	79.12
Fixed Carbon	11.7	7.83
Ash	0.9	1
Ultimate analysis (wt% db)		
Carbon	42.62	50.67
Hydrogen	5.56	6.18
Oxygen	51.01	40.97
Nitrogen	0.76	2.0
Sulphur	0.05	0.18
Calorimetric analysis (MJ/kg db)		
Lower heating value	16.25	18.69

The gas composition for the same temperature values and the same air-fuel ratios should be fixed, even though there are instantaneous differences due to channeling and bridging phenomena, it is expected that the average value and trend of the gas composition will be the same when the experiment is repeated. Moreover, C, H and O contents of biomass are calculated with elemental analysis of biomass. Among the features that are used in this experiment, the measurement range for the temperature values, T_0, T_1, T_2, T_3, T_4, T_5 is up to 1250°C, where the measurement uncertainty is ±2°C. In addition, the measurement uncertainty for the C is ±0.45% and for the H is ±0.1%. For the O, the measurement uncertainty is ±0.6%, whereas, for the ER and FR, it is ±1%.

3.2 Methods

3.2.1 Feature extraction and preprocessing

In order to use machine learning-based methods successfully, one must select a set of features that are informative and appropriate for the problem. For gasification process, equivalence ratio (ER), fuel flow rate (FR) and distribution of temperature (T_0, T_1, T_2, T_3, T_4, T_5) are important dynamics, thus, they are selected as features. Furthermore, the feature set was extended by adding the ultimate and proximate values of the biomass that are Carbon (C), Hydrogen (H), Oxygen (O), Nitrogen (N),

Moisture (M), VM, FC and Ash. However, S (Sulphur) that comes from elemental analysis from biomass involves redundant information since the changes in these variables are either constant or too low to discriminate. Therefore, the remaining sixteen features are used to predict CO, CO_2, CH_4, H_2 and HHV values and constructed a $16x1$ dimensional feature column vector, $\boldsymbol{x}_n$, for the n^{th} sample in the data set.

Once the features are extracted, distance measures will be exploited for the prediction of the dependent variables of biomass gasification, i.e., CO, CO_2, CH_4, H_2 and HHV levels, based on similarities among the features. To prevent larger valued features from dominating scale sensitive computation of metrics such as Euclidean or Manhattan distances, feature normalization is performed on the dataset with linear scaling to the unit range (Equation 3.3).

$$\tilde{x}_i[n] = \frac{x_i[n] - min(x_i)}{max(x_i) - min(x_i)} \tag{3.3}$$

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

3.2.2 Principal component analysis (PCA)

In order to address possible multicollinearity among the sixteen features and to represent independent variables with minimal redundancy, PCA is employed, which can find feature components that have low and high variance in a given data set [87]. For this purpose, one compute a sample covariance matrix as:

$$\boldsymbol{C} = \frac{1}{m-1}\sum_{n=1}^{m}(\boldsymbol{x}_n - \overline{\boldsymbol{x}_n})(\boldsymbol{x}_n - \overline{\boldsymbol{x}_n})^T \tag{3.4}$$

where, $\overline{\boldsymbol{x}_n} = \frac{1}{m}\sum_{n=1}^{m} \boldsymbol{x}_n$ and m is the total number of samples in the data set. Then, eigen decomposition of the 16x16 sample covariance matrix, C, is then performed to compute eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{16}$, which are sorted in decreasing order such that, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{16}$ whereas the corresponding eigenvectors are denoted as $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{16}$.

In order to achieve minimum redundancy, one needs to project the original feature vectors onto the first K principal components, i.e., eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k$, with the largest K eigenvalues such that:

$$\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{16} \lambda_i} * 100 \geq \zeta \tag{3.5}$$

the cumulative energy captured by these principal components account for at least ζ percent of the total energy in the dataset. In this section, ζ is selected as 90% and computed a projected set of vectors:

$$\boldsymbol{p}_n = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_K]^T \boldsymbol{x}_n \tag{3.6}$$

which are then used as features with reduced dimensionality. With the implementation of PCA, a total of sixteen features are reduced to three. One must note that three features created by the PCA algorithm have no physical correspondence for the biomass gasification process. They are algebraically transformed and dimensionally reduced vectors which consist of at least %90 of the information in the original feature set. It is also worth mentioning that, C, H, N, Moisture, Ash and FC are not included in the correlation equations by the PCA algorithm. This means that these features are highly correlated with other feature(s), thus, they do not contain enough unique information to be included in these equations [88].

3.2.3 Regression

3.2.3.1 Polynomial regression

Polynomial regression is a statistics-based technique which aims to represent the relationship between the independent variables (features) and the dependent variable (output) with an n^{th} $(n = 2, 3\ldots)$ degree polynomial equation [89]. The main goal of polynomial regression is to find optimal coefficients that can make satisfactory predictions of the output variable in terms of features used in a data set. The choice of the degree and type of the polynomial equation vary according to the problem for which one would like to develop a solution. In this section, the quadratic and cubic polynomial functions with interception terms are used as shown in Equation 3.7 and Equation 3.8, respectively:

$$h(x) = \sum_{i=1}^{n} \left(\alpha_i x_i + \sum_{j=i}^{n} \beta_{ij} x_i x_j) \right) + \gamma \tag{3.7}$$

$$h(x) = \sum_{i=1}^{n} \left[\alpha_i x_i + \sum_{j=i}^{n} (\beta_{ij} x_i x_j + \sum_{k=j}^{n} \delta_{ijk} x_i x_j x_k) \right] + \gamma \tag{3.8}$$

where, $\alpha, \beta \; and \; \delta$ are the coefficients of the given polynomial function and γ is the constant bias term.

In order to determine these variables, one must select a proper cost function (J) and minimize it to obtain a hypothesis function, $h(x)$ that can mimic pheromone one aims to model which corresponds to the great generalization performance. In this section, least-squares cost function with L_2 regularization is used to find optimal coefficients while preventing overfitting to maximize generalization performance (Equation 3.9) [90].

$$J = \sum_{i=1}^{m} (y_i - h(x_i))^2 + \lambda \left(\sum_{i=1}^{n} \alpha_i^2 \sum_{j=i}^{n} \beta_{ij}^2 \sum_{k=j}^{n} \delta_{ijk}^2 \right) \quad (3.9)$$

where, $h(x_i)$ presents the prediction of the hypothesis function for i^{th} sample of the feature vector, m is the number of samples in the data set. λ is the regularization parameter. After the minimization of J w.r.t. parameters of $h(x)$, one can obtain a polynomial representation of the phenomenon.

3.2.3.2 Support vector regression

Support vector machines (SVM) is a large-margin classification algorithm that has been broadly used in various applications as explained in Section 2.2.2. Support vector regression (SVR) is a special case of SVM where the aim is to predict continuous variables rather than categorical ones [91]. General estimation function of SVR is given in Equation (3.10)

$$f(x) = \sum_{i=1}^{n} \alpha_i * K(x_i, x_j) + b \quad (3.10)$$

where, $a_i (i = 1,2,3, \dots n)$, b and $K(x_i, x_j)$ are the support vectors and bias term and kernel function as same as in SVM, respectively. Radial basis function with $\sigma = 0.15$ is used as the kernel function in this section. After the selection of kernel, a cost function must be selected to find support vectors and bias term. Obviously, same cost function used in SVM cannot be utilized for SVR due to the dependent variable being continuous type. For this reason, empirical risk minimization approach (Equation 3.11) with robust ε-insensitive loss function (Equation 3.12) is used to train SVR model. With the minimization of Equation 3.11, one can use resulting $\boldsymbol{a}$ vector and constant b to make predictions using Equation 3.10.

$$J = \sum_{i=1}^{m} L_\epsilon(f(x_i), \boldsymbol{y}) \quad (3.11)$$

$$L_\epsilon = \begin{cases} 0, & \text{if } |\boldsymbol{y} - f(x)| \leq \epsilon \\ |\boldsymbol{y} - f(x)| - \epsilon, & \text{otherwise} \end{cases} \quad (3.12)$$

where $\boldsymbol{y}$ is the output vector and ε is the very small epsilon value.

3.2.3.3 Decision tree regression

Decision Trees (DT) is a widely used and intuitive machine learning method for creating prediction models based on basic logical statements [92]. Unlike SVM or polynomial regression, the relationship between features and output is not predetermined which means it does not use a function to fit data. DT creates a set of questions such as "is equal" or "is greater" using feature set, with the given "yes" or "no" answers, another question will be encountered to answer. This process is repeated until there are no more questions to answer, thus, the output is obtained. DT is grown by repeatedly dividing the data into binary sections. The divisions for all features are examined with a randomness measure such as entropy [93]. The division that makes output distribution most homogeneous is selected as the best division and added to the tree. This selection recursively repeated until perfect homogeneity (if possible, otherwise best homogeneity available in the set) is obtained or prespecified maximum number of nodes is reached. Even though this algorithm is suitable for the classification task, it can be twisted for regression problems also known as DTR. In DTR, rather than using a randomness metric, DTR utilizes some of square errors metric to create subsets which has the lowest possible difference between the elements rather than homogeneity of the classes [94]. After that, one can again follow the questions in the tree to obtain a continuous variable which is the prediction for the given feature set.

3.2.3.4 Artificial neural network

ANN is a machine learning methodology inspired by the human brain and its information processing structure [95]. It has found widespread usage across different types of problems in various disciplines due to its proven success for predicting both continuous and discrete variables [96]. ANN consist of neurons which are the primary processing elements, and neuron clusters which are called layers. Neurons receive inputs from neurons in the previous layer, process the information with its

activation function and transmit output for the neurons in the next layer [97] (Figure 3.1).

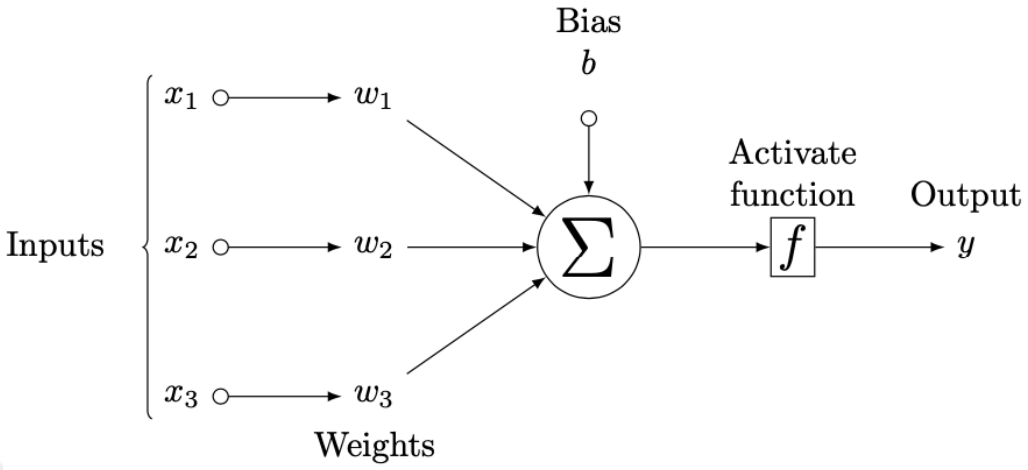


Figure 3.1 Simple neuron representation of ANN

This transmitting process starts with the input layer and continues until the neurons in the output layer produce output(s). Every layer between input and output layers are called hidden layer, the number of hidden layers and number of neurons in each hidden layer and activation functions are hyper-parameters which must be determined beforehand. Output (prediction) formulation for a single hidden layer and single output feedforward neural network can be defined as:

$$\tilde{\mathbf{y}} = \delta_2\left(\sum_{i=1}^{m}\left(\boldsymbol{w}_i^{(2)}\delta_1(\mathbf{X})\right) + \mathbf{b}^{(2)}\right)$$
$$\mathbf{X} = \sum_{j=1}^{n}\left(\mathbf{x}_j\mathbf{w}_{xj}^{(1)}\right) + \mathbf{b}^{(1)} \tag{3.13}$$

where, $\tilde{\mathbf{y}}$ is the prediction vector of the ANN model, m is the number of samples in the data set, n is the number of features in the dataset, $\boldsymbol{x}_j$ is the j^{th} feature vector, $w^{(2)}$ are the weights between hidden layer and the output layer, $w^{(1)}$ are the weights of inputs connected to hidden layer δ_2 is the activation function of the output layer. δ_1 is the activation function of the neurons in the hidden layer. $\mathbf{b}^{(2)}$ and $\mathbf{b}^{(1)}$ are the bias vectors in output layer and hidden layer, respectively. In the proposed ANN

model, two hidden layers with ten neurons each as shown in Figure 3.2 is used due to its superior performance compared to other topologies. The sigmoid activation function is used in hidden layers in order to make ANN "learn" non-linearity of the data (Equation 3.14) and linear activation function is used at the output neuron to obtain a continuous variable as the prediction (Equation 3.15).

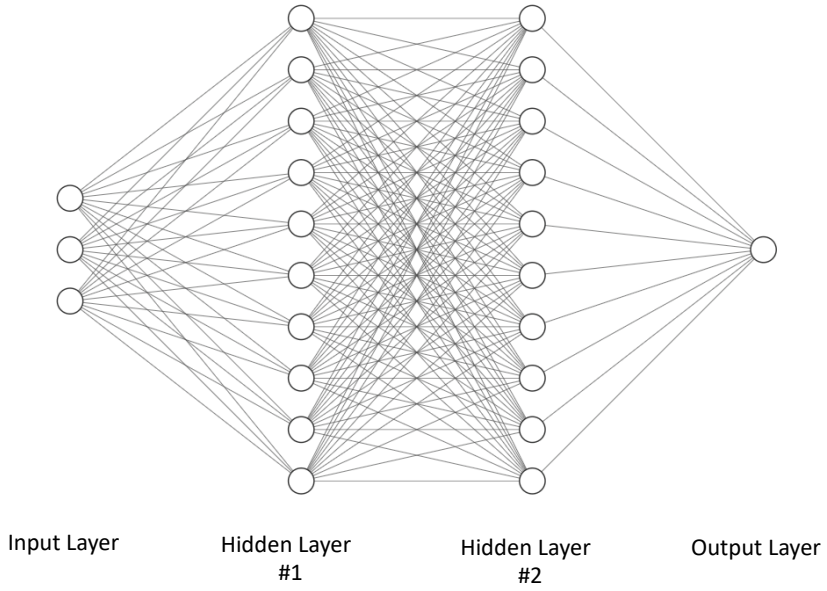


Figure 3.2 Developed ANN structure

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.14}$$

$$f(x) = c * x \tag{3.15}$$

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, L_2 regularized sum of square errors with as cost function and every weight's contribution on the cost (jacobian matrix) are calculated with propagating error back to the corresponding weights and batch-gradient descent algorithm is used to update the weights [98].

3.2.4 Performance evaluation

In order to evaluate the performance of the regression models, 10-fold cross-validation as explained in Section 2.2.5 is used with correlation coefficient (R^2), adjusted correlation coefficient (Adj. R^2), root mean square error (RMSE) and normalized root mean square error (NRMSE) metrics.

$$R^2 = 1 - \left[\frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \overline{y})^2}\right] \tag{3.16}$$

$$Adj.R^2 = 1 - \left[\frac{(1 - R^2)(m - 1)}{m - k - 1}\right] \tag{3.17}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{m}} \tag{3.18}$$

$$NRMSE = \frac{1}{m}\sqrt{\frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\overline{y}}} \tag{3.19}$$

where $\hat{y}_i$ is the prediction vector, $\overline{y}$ is the mean of actual output vector and k is the number of features in the data set.

3.3 Results & Discussion

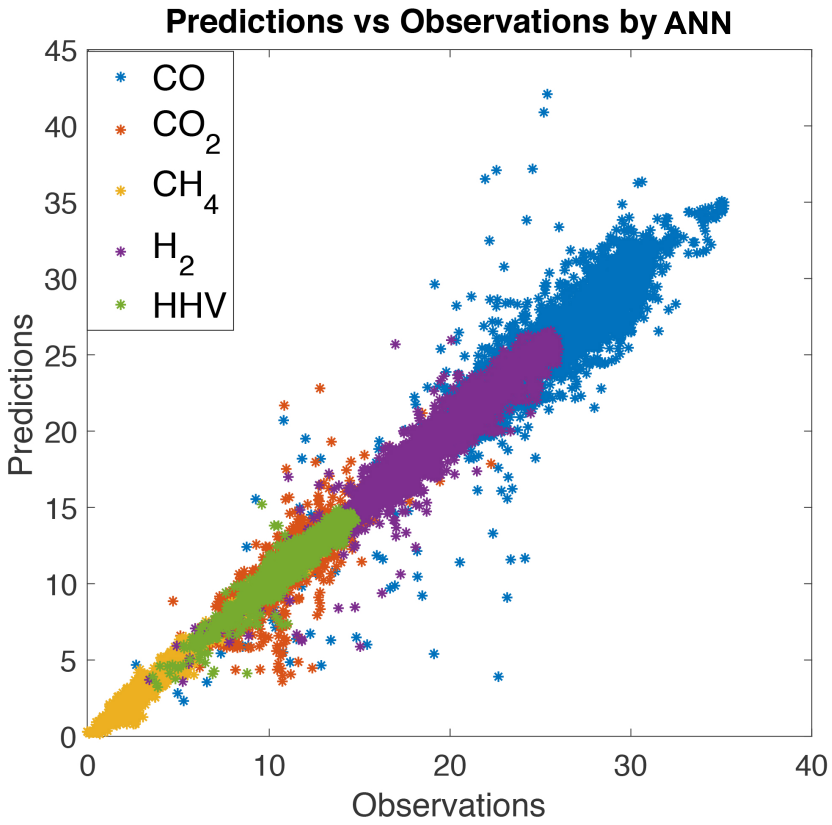
The proposed four regression methods are employed to develop prediction models for five distinct outputs, i.e., i.e., CO, CO_2, CH_4, H_2 and HHV using three features generated by the PCA procedure as mentioned in Section 3.2.2. All prediction

models are trained to predict single output variable and their performances are evaluated according to their test performance in 10-fold cross validation by using R^2, Adj. R^2, RMSE and NRMSE metrics.

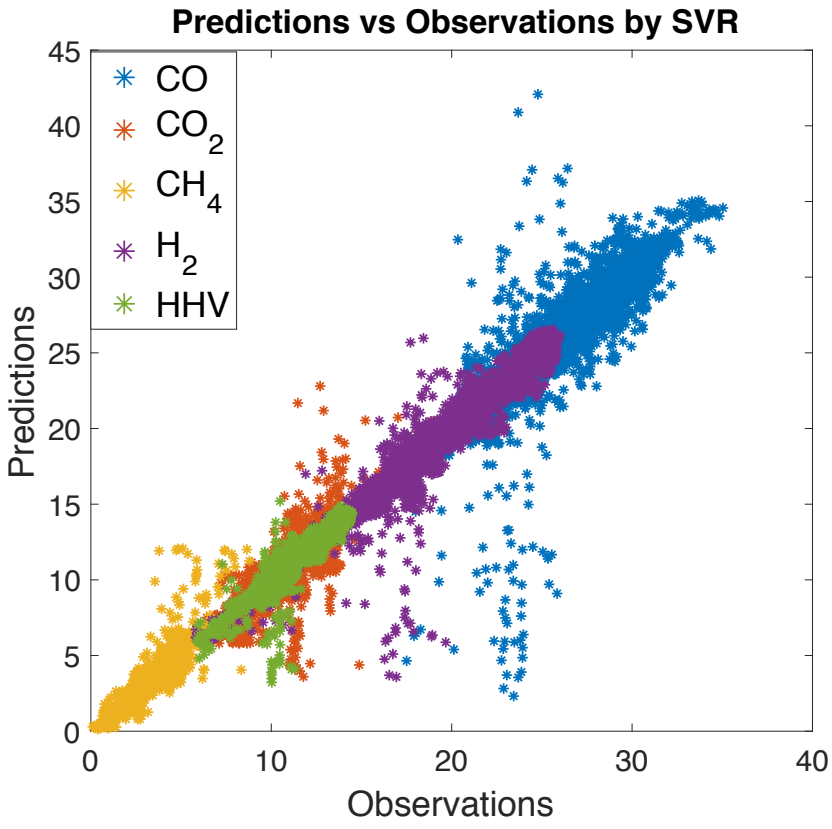
Table 3.2 shows the performance of the proposed techniques in predicting the five distinct outputs variables. For a better understanding of the regression models' performances, the predicted outputs of all 4826 samples versus actual observations plots are created as shown in Figure 3.3. In these plots, the ideal case would be when all points lay on a straight line with slope equal to 1 which indicates that all predictions are equal to observations.

Table 3.2 Performance evaluation of the proposed regression methods

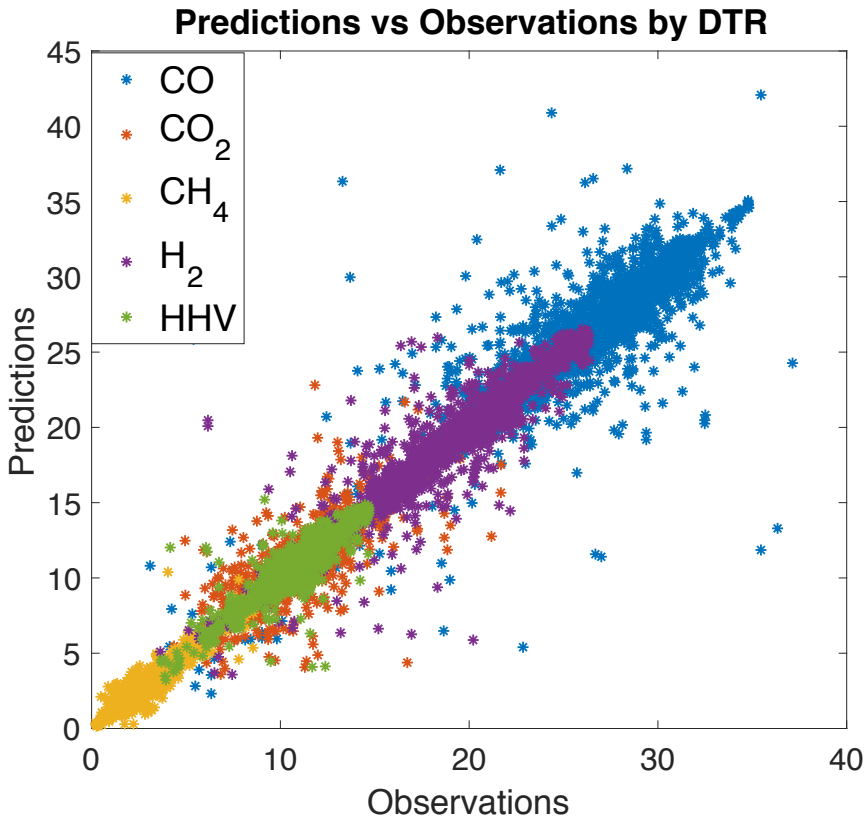
Output	Method	R^2	Adj.R^2	RMSE	NRMSE
CO	SVR	0.713	0.712	1.965	0.278
	DTR	0.807	0.806	1.211	0.099
	PR (Quadratic)	0.541	0.54	2.469	0.341
	PR (Cubic)	0.71	0.709	1.964	0.271
	MLP	0.823	0.822	1.119	0.116
CO_2	SVR	0.812	0.811	0.88	0.117
	DTR	0.852	0.851	0.514	0.092
	PR (Quadratic)	0.567	0.566	1.268	0.175
	PR (Cubic)	0.719	0.718	1.022	0.141
	MLP	0.837	0.836	0.562	0.102
CH_4	SVR	0.827	0.827	0.53	0.219
	DTR	0.944	0.944	0.125	0.18
	PR (Quadratic)	0.682	0.681	0.702	0.582
	PR (Cubic)	0.85	0.849	0.482	0.201
	MLP	0.927	0.927	0.176	0.198
H_2	SVR	0.915	0.915	1.032	0.128
	DTR	0.938	0.938	0.352	0.071
	PR (Quadratic)	0.726	0.725	1.808	0.249
	PR (Cubic)	0.865	0.854	1.269	0.175
	MLP	0.944	0.944	0.328	0.056
HHV	SVR	0.886	0.885	0.552	0.097
	DTR	0.921	0.92	0.286	0.058
	PR (Quadratic)	0.713	0.712	0.866	0.112
	PR (Cubic)	0.858	0.857	0.61	0.101
	MLP	0.931	0.931	0.221	0.045


Predictions vs Observations by ANN
CO
CO2
CH4
H2
HHV
Predictions
Observations
0
5
10
15
20
25
30
35
40
45
0
10
20
30
40

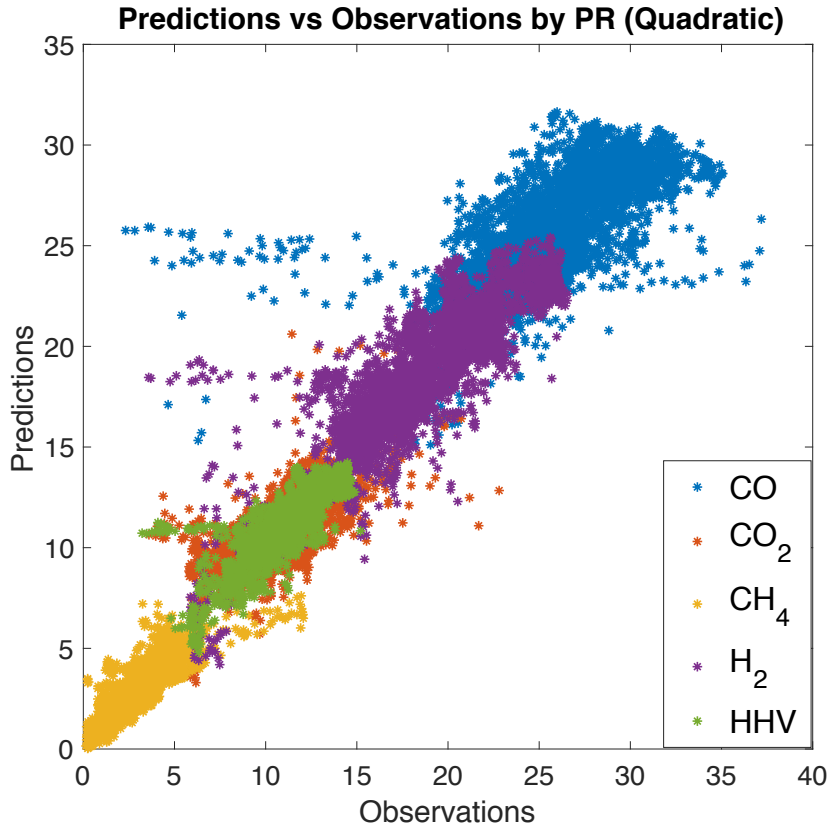
(a)


Predictions vs Observations by SVR
CO
CO2
CH4
H2
HHV
Predictions
Observations
0
5
10
15
20
25
30
35
40
45
0
10
20
30
40

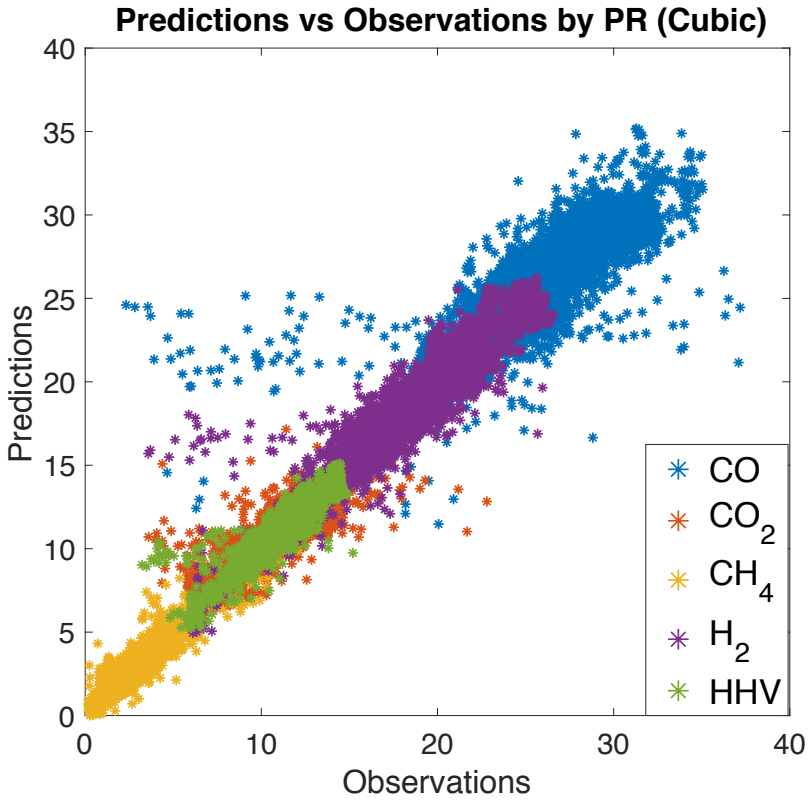
(b)


Predictions vs Observations by DTR
CO
CO2
CH4
H2
HHV
Predictions
Observations
0
5
10
15
20
25
30
35
40
45

(c)


Predictions vs Observations by PR (Quadratic)
CO
CO2
CH4
H2
HHV
Predictions
Observations
0
5
10
15
20
25
30
35
40

(d)



(e)

Figure 3.3 Predictions vs observations plots for all proposed regression methods

As one can see in Table 3.2, the performance of the MLP and DTR was similar and superior to other methods for all output variables due to their higher R^2 and Adj.R^2 scores and less RMSE and NRMSE values. For CO, H_2 and HHV outputs, MLP slightly outperformed DTR, whereas DTR results better for the remaining outputs. When the complex dynamics of the biomass gasification process is concerned, MLP is usually expected to have greater performance compared to other machine learning methods due to its proven success for learning non-linear and complex behaviors exception-ally well [99]. On the other hand, DTR builds a decision-based hierarchical structure using the training data and it is known for creating opportunities for exploratory analysis with its easy to understand the structure. Since the DTR performs similarly to the MLP, it can be an effective tool to make predictions about the biomass gasification process. On the contrary, when Figure 3.3a and Figure 3.3c are examined, DTR's predictions are more scattered compared to MLP's, which means that predictions made by DTR tend to either scatter on the 45-degree line or scatter irrelevantly. This is caused by a strict rule-based learning

approach that the DTR utilizes. When the high preciseness required for the biomass gasification process is concerned, DTR may cause instability problems in practical applications and using it along with other methods might be a better approach. Furthermore, just like MLP, the performance of the DTR proportionally increases with the number of samples in the data which means one can expect greater prediction performance using a data set with more data samples than what we have used in this section [100].

PR is employed in two forms, i.e., quadratic and cubic. Quadratic PR's prediction performance was catastrophically worse compared to other methods according to Table 3.2. Hence, one can conclude that the quadratic combination of features is not sufficient to present any of the outputs and more complex terms are required. For cubic PR, prediction performance is significantly increased compared to quadratic PR. Although cubic terms of features have helped to represent outputs, it was not as successful as DTR or MLP. But when the performance increase with cubic PR is compared to quadratic PR is concerned, increasing the order of the polynomial or introduction of non-linear terms (non-linear regression) may help to increase. Moreover, when Figure 3.3d and Figure 3.3e are examined, one can see that both quadratic and cubic PR's false predictions of CO output lay in a specific range of H_2 and HHV outputs. This is not common behavior for MLP nor DTR where the false predictions spread more uniformly. Therefore, we can say that PR is not able to understand the unique dynamics of CO output with the given feature set and one may need to extract more features for PR methods to achieve greater prediction performance. SVR had similar performance compared to cubic PR for all outputs except for CO_2 and its overall performance was still worse than the DTR and the MLP. One of the reasons for the inferior performance of SVR is the broad range of output variables. Because SVR is the extension of a classification algorithm (SVM), it tends to create a group the output variable during the training phase [101]. As one can see in Figure 3.3b, all predictions are stuck at a certain range, thus, false predictions are also "grouped" in the same range. This behavior can be helpful when the samples of a certain output variable are between some range of values. Otherwise, like it is in biomass gasification, output samples that are out of the range of majority are considered as outliers and highly ignored during the creation of the

support vectors, hence, the overall prediction capability of the model is decreased. Even though the performance of the SVR might increase with the use of different kernels and hyper-parameters, it can be considered as a sub-optimal regression technique for the biomass gasification process compared to others.

In addition to comparing the overall performances proposed machine learning methods, one must investigate the prediction performance for each output variable to explore the viability of machine learning methods for the biomass gasification process more deeply. For HHV output, MLP and DTR were able to achieve satisfactory prediction accuracy with $R^2 = 0.931$ and $R^2 = 0.921$, respectively. Other proposed regression methods also came close except for quadratic PR which means that the given feature set contains enough information for proposed machine learning algorithms to learn and make accurate predictions about HHV. This statement can also be made for H_2 and CH_4 outputs where the proposed methods had also similar performances. On the other hand, for outputs CO_2 and CO, the performance of all methods are declined. It can be deduced that the dynamics of these outputs are more complex than the others and the proposed methods are struggling to understand their complexity. There are several steps one can take to increase performance for these outputs. First, the number of samples in the data set can be increased by collecting more experimental results. This approach can significantly increase the prediction performance of machine learning methods especially DTR and MLP due to their hunger for data [102]. Also, one can extract more feature with respect to the nature of the biomass gasification. But this approach may result in the addition of unnecessary feature which can deceive the models and cause performance decline. Lastly, rather than treating the CO and CO_2 outputs as a part of the biomass gasification process, one can create models specifically designed just for predicting those outputs individually and with using different set of features and hyper-parameter configurations, it would be possible to obtain greater prediction performance for CO and CO_2.

In order to compare the results obtained in this section to other modeling techniques, a comparison table is created as shown in Table 3.3. In this table, RMSE values of stoichiometric and non-stoichiometric modeling approaches from [103], least-square

SVM (LS-SVM) and random forest (RF) classification approach from a previous study that uses same data set are included and compared for CO, CO_2, CH_4, H_2 and HHV outputs of the biomass gasification process [11]. In the table, only the best performing regression methods, MLP and DTR, are included from this section for the sake of simplicity. kinetic or CFD modeling are not included due to its enormous computational requirements which reduces its' practical usability beforehand compared to proposed methods.

Table 3.3 Comparison of RMSE values between this study and other modeling approaches

Output	Method	RMSE	Reference
CO	DTR	1.211	This Study
	MLP	1.119	This Study
	LS-SVM	1.512	[11]
	RF	5.144	[11]
	Stoichiometric	34.99	[103]
	Non-Stoichiometric	3.936	[103]
CO_2	DTR	0.514	This Study
	MLP	0.562	This Study
	LS-SVM	0.64	[11]
	RF	2.8	[11]
	Stoichiometric	20.33	[103]
	Non-Stoichiometric	1.59	[103]
CH_4	DTR	0.125	This Study
	MLP	0.176	This Study
	LS-SVM	0.348	[11]
	RF	1.796	[11]
	Stoichiometric	3.846	[103]
	Non-Stoichiometric	4.32	[103]
H_2	DTR	0.352	This Study
	MLP	0.328	This Study
	LS-SVM	0.595	[11]
	RF	5.041	[11]
	Stoichiometric	2.256	[103]
	Non-Stoichiometric	12.23	[103]
HHV	DTR	0.286	This Study
	MLP	0.211	This Study
	LS-SVM	0.38	[11]
	RF	2.328	[11]
	Stoichiometric	1.53	[103]
	Non-Stoichiometric	0.87	[103]

As one can see from Table 3.3, for all outputs, MLP and DTR regression methods were able to outperform stoichiometric and non-stoichiometric modeling as well as the classification-based approach using LS-SVM and RF in a previous study. These

results also justify the application of PCA to the extracted features which can be utilized in further applications where the data set and the number of features is larger and more computationally efficient training procedure is needed. Moreover, the toe-to-toe performance of DTR compared MLP shows that the use of artificial neural network-based algorithms should not be the only machine learning-based approach to employ. Intuitive and low computational cost algorithms such as DTR can give similar or even better performance and create new ways to deeply understand and create predictions models for the biomass gasification process and many more energy applications. Furthermore, trained models of the two best performing method in this section, i.e., MLP and DTR is provided in Github (https://github.com/furkanelmaz/PredictiveModelingofBiomassGasification). With the provided models, one can simulate outcomes of biomass gasification with different scenarios. Because these pre-trained models have low computational requirement to make predictions, one can also directly embed them to a microcontroller-based embedded system by using MATLAB Coder to utilize it practically such as by feeding real-time sensor measurements to the prediction model and "foresee" the next state of the biomass gasification's outcomes to increase efficiency and to experiment with it.

4.TIME SERIES MODELING AND MODEL PREDICTIVE CONTROL OF BIOMASS GASIFICATION PROCESS

In the previous section, several regression methods are employed to predict outputs of the gasification process. Although results were satisfactory and better than the previously used modeling approaches, prediction were in time-independent fashion. This means the outputs are predicted for a 'snap' of time rather than a series of time. This approach highly limits the practical usability of the models because biomass gasification is a process and previous states of the system have effect on the further ones. Furthermore, not including the previous states also makes impossible to maximize desired outputs throughout the process because time-independent models are fed with certain feature set from unknown time instance, thus, different dynamics caused by the previous states are not acknowledged by them. In this section, this problem is addressed by developing a time series-based ML model for the biomass gasification process. Moreover, a model predictive controller is designed on top of that to maximize certain desired outputs and deeply discuss practical usability perspective. The study in this section is conducted in two steps:

The first step is the development of a time series-based ML model of biomass gasification which can mimic the dynamics of the real process with minimal error rates. For this purpose, same data set that used in the Section 3 is utilized. Then, the nonlinear autoregressive with external input neural network method (NARXNN) modeling approach is taken to create the model due to its time-dependent nature and its previously proven success for many process modeling studies. The second step is to create a model predictive controller that uses only data to create a predictive model and change the ER variable to control desired output variables with "foreseeing" next state of the process. Thus, the aim of this section is to propose a highly accurate time-dependent predictive model of biomass gasification process and model predictive controller to control syngas compositions by manipulating ER variable to create both a realistic simulation platform and practically usable controller design.

4.1 Methods

4.1.1 Nonlinear Auto Regressive with Exogenous Input Neural Networks (NARXNN)

Nonlinear Auto Regressive with Exogenous Input (NARX) is a widely used predictive modeling technique for time series problems [104]. NARX utilizes not only previous values of the output variable it also adds another relevant time series as input (exogenous input) to the model in order to boost prediction performance and generalization capability of the model [105]. A NARX model can be defined as:

$$\tilde{y}_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-dy}, x_{t-1}, x_{t-2}, \ldots x_{t-dx}) + e_k \tag{4.1}$$

where, $\tilde{y}_t$ is the predicted output value at time t, x is the exogenous time series input, d_y and d_x are the number of delays for outputs and inputs, respectively. f is a nonlinear function of previously known d_x and d_y number of inputs and outputs determined during the training of the model, respectively. e_k is the error term (also known as noise term) which assumed to be white gaussian noise in this study.

Even though f in Equation 4.1 can be any arbitrary nonlinear function, this function must be capable of expressing the given nonlinear phenomena in order to obtain a successful prediction model [106]. For this reason, a combination of ANN and NARX, also known as NARXNN introduced in literature [107]. In NARXNN, nonlinear function f given in Equation 4.1 is replaced by a feedforward neural network model [108]. Moreover, when the nonlinear activation functions such as sigmoid is used in hidden layers of this ANN, NARXNN model can learn nonlinearity and can be utilized to develop models for problems with high complexity [109]. Single hidden layer and single output NARXNN model can be defined as:

$$\begin{aligned} \tilde{\boldsymbol{y}}_t &= \psi_2\left(\sum_{i=1}^{m} w_i^{(2)} \psi_1(\boldsymbol{X}_{t-1}) + \boldsymbol{b}^{(2)}\right) + e_k \\ \boldsymbol{X}_{t-1} &= \sum_{j=1}^{dx} \boldsymbol{w}_{xj}^{(1)} \boldsymbol{x}_{t-j} + \sum_{k=1}^{dy} w_{yk}^{(1)} y_{t-k} + \boldsymbol{b}^{(1)} \end{aligned} \tag{4.2}$$

where, m is the number of samples in the data set, $\mathbf{w}^{(2)}$ is the weight vector between the hidden layer and the output layer, ψ_2 is the activation function of the output layer which is selected as linear when the desired output is continuous variable (regression). ψ_1 is the activation function of the neurons in the hidden layer $\boldsymbol{b}^{(2)}$ and $\boldsymbol{b}^{(1)}$ are the bias terms in the output layer and hidden layer, respectively. $\boldsymbol{w}_x^{(1)}$ and $\boldsymbol{w}_y^{(1)}$ are the weight vectors of inputs connected to the hidden layer and weights of delayed outputs connected to the hidden layer, respectively.

Even though Equation 4.2 is defined for single hidden layer and single output architecture, one can easily extend the hidden layer size by recursively defining $\boldsymbol{X}_{t-1}$ for each hidden layer with different weight matrices and can predict more than one output by adding the desired number of neurons in the output layer [110]. One must note that the training of the model is performed with series- parallel NARXNN architecture where the true output is used instead of feeding back the estimated output at each time instance. After the training of the model, architecture is converted to the parallel structure where the predictions from the previous time instance are fed to the model as inputs. Thus, aside from initial conditions, the model is self-sufficient at each sampling instance and training of the model can be performed with conventional feed-forward ANN training algorithms such as backpropagation [111]. NARXNN model is developed to predict five outputs using a single hidden layer. Details of the model are given in Section 4.2.

4.1.2 Model predictive controller

Model predictive controller (MPC) is an established control methodology based on changing the controllable variable(s) to control desired output(s) by predicting the further states of the system (plant) and by performing online optimization at each sampling instant [112]. MPC requires a dynamic model of the system to "forecast" next states, this model must define the variable that aimed to be controlled in terms of inputs (controllable variable) that can dynamically affect the plant [113]. Also, it should mimic the real dynamics of the system as accurate as possible for good control performance [114]. Assume that, mentioned dynamic model is available as $M_t(x)$ which is a function of controllable variable x. One can define a cost function

(J) such as the sum of square errors that includes the reference value at time t as y_{ref} :

$$J = \frac{1}{2}(y_{ref} - M_t(x))^2 \tag{4.3}$$

With the minimization of Equation 4.3, one can obtain the optimal x value (x^*) and feed it to the plant to approximate the output variable to the reference value. Another feature of MPC is the ability to prevent violation of constraints [115]. One can add a desired number of constraints to the Equation 4.3 and perform minimization while satisfying these constraints. This feature is especially useful in energy applications where many physical constraints exist. Moreover, the Equation (3) can be expanded by introducing a move suppression term which is a penalty term that penalize the change in manipulated variable to prevent controller from aggressively try to make the system stabilize at the reference [116]. Even though this term is not required for the case studies conducted in this section, one can need to add this term for different applications of the proposed framework. Similarly, hard limiting of the control signal might be needed to force the duty signal in the meaningful range and this can be achieved by adding the desired range of the control signal as an additional constraint to the optimization function.

Another perspective one should consider is the development of $M_t(x)$. This dynamic model is no different than a conventional machine learning model where the outputs are represented in terms of input variables and this is the reason why MPC is also mentioned as machine learning control [117]. One can argue that the same NARXNN modeling approach which explained in Section 4.1.1 can be employed during the implementation of MPC. Even though this statement is correct, the NARXNN model contains recursively defined layer structures with nonlinear activation functions. Because MPC solves an optimization problem at each sampling instant, minimization of the cost function which includes the NARXNN model would have enormous computational expense especially when it must be solved in real-time. For this purpose, a simplified model of biomass gasification by employing

polynomial regression is used to develop a dynamic model for MPC. Details of the developed MPC model are given in Section 4.3.

4.1.3 Rolling-windows analysis

Unbiased performance evaluation is an important step while developing a prediction model to measure the generalization performance of the model and to investigate possible overfitting problems. For this purpose, several cross-validation methods such as k-fold and hold-out are primarily used in machine learning applications. Even though these methods are suitable for many regression and classification problems as k-fold is used in this thesis, they are suboptimal to be used for time series. Because, these methods are based on random splitting which means they create a test/validation sets by using random samples in the data set, thus, time is not a considered variable [118]. On the other hand, in time series, outputs/inputs in previous time steps are important factors and performance evaluation based on randomly selected subsets would be unreasonable [119]. In order to evaluate the performance of a predictive model developed for the time series problem, rolling-window analysis is successfully used in the literature [120]. In this analysis, the first k number of data points are used to train the model, then, prediction of the model for p number of sequential data points (window) after k (test) are stored. After that, the model is retrained by using the first k + p data points and a prediction vector for the next window is stored. This procedure is repeated until the model predicts the last data point in the data set. Illustration of rolling-windows analysis is given in Figure 4.1.

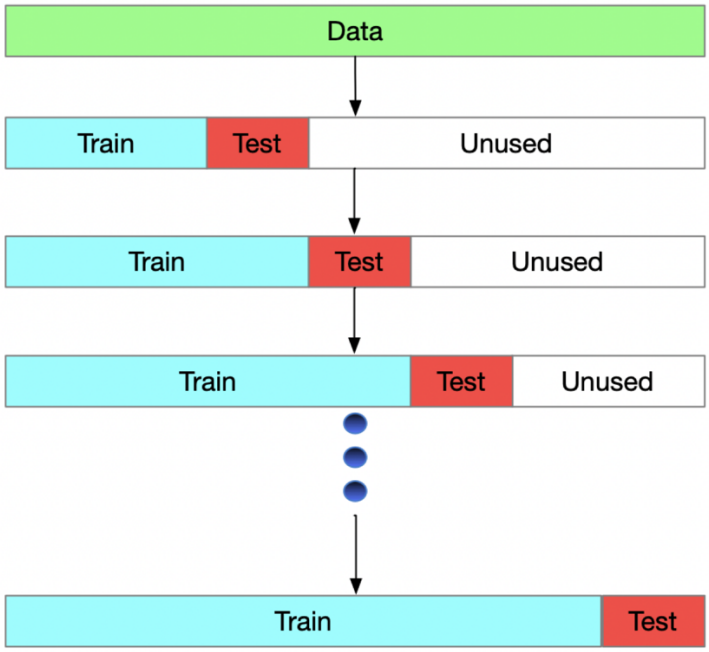


Figure 4.1 Visual demonstration of rolling-windows analysis

After collecting the predictions of the model for all data points except the first k data points, one can use correlation and/or error metrics to compare actual data points (observations) and predicted data points (predictions) to evaluate prediction performance and generalization capability of the model. In this section widely R^2, Adj.R^2, RMSE and NRMSE metrics are used like the previous section to evaluate the performance of both NARXNN and polynomial regression models.

4.2 NARXNN Modeling

During the NARXNN modelling, ultimate analysis results, i.e., C, H, O, N and proximate analysis results, i.e., FC, VM, Moisture, Ash and ER variables are utilized as exogenous inputs. Output HHV, gas concentrations, i.e., CO, CO_2, CH_4, H_2 and T_0 are used as outputs in the NARXNN model. All of these inputs variables except ER are time-independent, thus, no delay is added to them. On the other hand, ER has a time-dependent effect on the outputs which means current values of the outputs are affected by previous ER values, therefore, 2 number of delays including its' real-time value (ER_t, ER_{t-1}, and ER_{t-2},) are added while feeding it to the model. Moreover, outputs with 2 number of delays, excluding their real-time values, are also used as inputs to make the NARXNN model understand the time-dependent structure of the

outputs. For neural network architecture, a single hidden layer with ten neurons with sigmoid activation function on each neuron is used, in the output layer linear activation function is used to predict continuous type variables. The optimal number of neurons in the hidden layer is determined by trying different range numbers and calculating the average R^2 of all outputs. Resulting average number of neurons vs average R^2 plot is provided in the Appendix. The architecture of the proposed NARXNN model is illustrated in Figure 4.2.

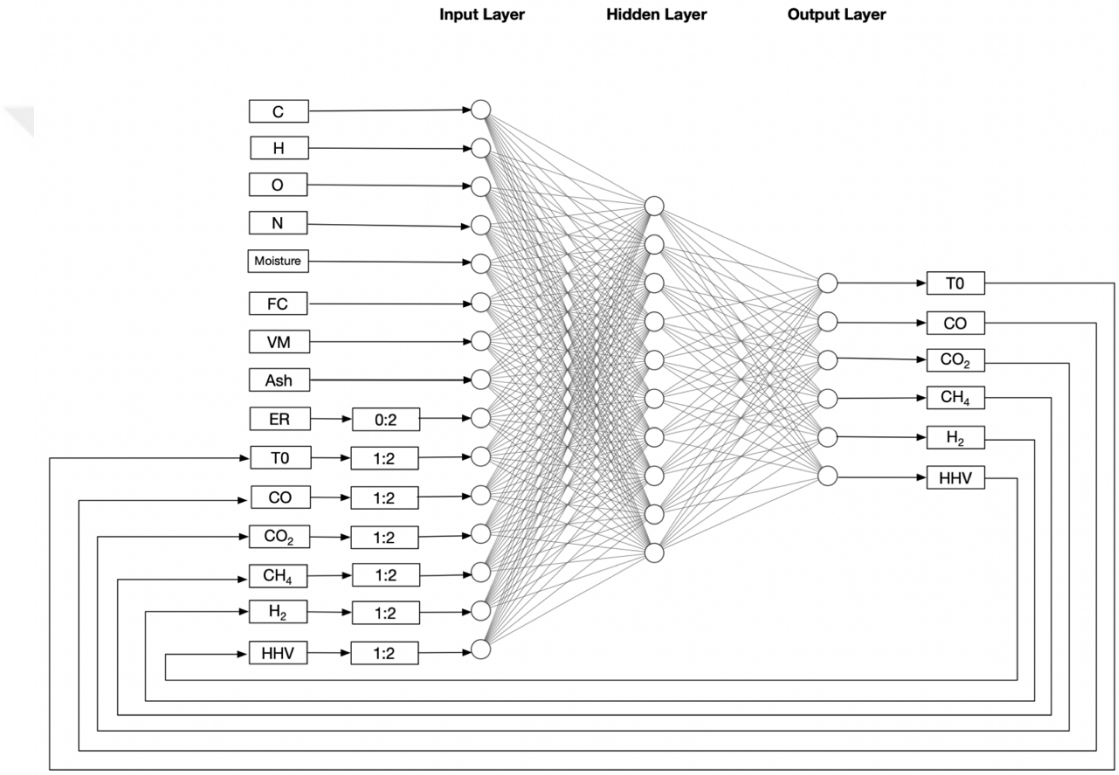


Figure 4.2 Developed NARXNN model

Performance of the proposed NARXNN is evaluated by rolling-windows analysis as explained in Section 4.1.3. The first 1000 data points are used for initial training of the model and rest are predicted as windows that have a size of 100. Performance evaluations are given in Table 4.1, NARXNN predictions and actual time series plot is given in Figure 4.3 for each output variable.

Table 4.1 Performance evaluation of the NARXNN model

Output	R^2	Adj.R^2	RMSE	NRMSE
CO	0.9888	0.9885	0.3233	0.0241
CO_2	0.9853	0.985	0.2138	0.0226
CH_4	0.9839	0.9811	0.1027	0.034
H_2	0.9931	0.9929	0.2605	0.017
HHV	0.9941	0.9939	0.121	0.016
T0	0.9854	0.984	4.5374	0.0246

Figure 4.3 NARXNN prediction and actual values plot, '-P' indicates the predicted values of a certain output

As one can see from Table 4.1, the NARXNN model was able to "learn" the dynamics of biomass gasification quite successfully by reaching $R^2 > 0.98$ for all of the outputs. One must also note that given performance metrics are calculated with rolling-windows analysis which means that the model is challenged to predict

outputs while training samples constantly varying as explained in Section 4.1.3. One can conclude that the proposed NARXNN model was able to dynamically adapt all time periods, excluding the first 1000 data points which are reserved for initial training and predict outputs accordingly. Another perspective one must look at is the model's ability to understand trends in the data set. In the data set, there are several increasing and decreasing trends in all of the outputs. NARXNN model was able to catch those trends strongly and it is almost impossible to distinguish actual values from predicted ones by the naked eye. That situation also justifies the utilization of delayed output variables that are fed to the model, because it is also known that time-independent prediction methods such as support vector machines and decision trees struggle with keeping out with changing trends in a data set [121,122]. When the modeling performance of the proposed NARXNN model is concerned, it is safe to claim that, during the implementation of MPC for controlling outcomes of biomass process, NARXNN model can be treated as real plant and its' prediction would be exceptionally similar to one can have in real gasification system.

4.3 MPC Design

MPC method consists of two layers of implementation; first is the development dynamic model which will be used to predict the next state of the corresponding plant. To obtain such a model, polynomial regression method is employed as discussed in Section 3.2.3.1. Because polynomial regression can only be used to predict single output variable [123], five unique models for five controllable output variables are developed. Previous output variable, y_{t-1}, and ER are utilized as features for all models. The reason why such a low number of features is used, especially compared to the NARXNN model, is to increase computational efficiency. Because, during each sampling instant, MPC solves an optimization problem to determine optimal ER value which minimizes the cost function, thus, each extra term in the polynomial model would increase the computational time and reduce the controller's practical usability with microcontroller-based embedded systems. Furthermore, all polynomial regression models are trained with a batch gradient descent algorithm by minimizing L_2 regularized sum of square errors. Performance metrics resulting from rolling-windows analysis are given in Table 4.2 for each

output. Correlation equations, predicted output and actual output comparison plots are given in Appendix for the sake of clarity.

Table 4.2 Performance evaluation of the polynomial regression models

Output	R^2	Adj.R^2	RMSE	NRMSE
CO	0.8742	0.8733	0.9854	0.0734
CO_2	0.8321	0.8294	0.5076	0.0542
CH_4	0.8352	0.8302	0.3421	0.0936
H_2	0.8422	0.8415	1.2685	0.0841
HHV	0.8591	0.8572	0.4765	0.0644

As one can see from Table 4.2, prediction accuracy is dropped compared to the NARXNN model and the results given in Table 4.1. This situation is surely expected due to the superior learning capability of NARXNN model and significantly more number of features used during the NARXNN modeling. On the other hand, results are still can be considered satisfactorily, due to all of the polynomial regression models achieving over $R^2 > 0.8$. Even though polynomial regression models are simplified models to be used in MPC, they were still able to "learn" the dynamics of the biomass gasification to some degree. Also, as one can see in Appendix, polynomial regression models are able to catch increasing and decreasing trends on the data set. This situation shows that these models can accurately represent the next state of the biomass gasification for MPC to successfully generate optimal ER values. Moreover, these models contain only quadratic combinations of the output value from the previous time sample and ER variable. Therefore, it is reasonable to assume that the online optimization process can be executed in real-time by a computer or even with several microcontroller-based embedded systems.

The second layer of implementation during MPC design is the definition of an optimization problem. This step depends significantly on the control objective and the constraints occur in the nature of the process to be controlled. Moreover, this optimization problem will be solved at each iteration (online optimization) for satisfactory controller performance [124]. As stated in Section 4.2, a highly successful NARXNN model is treated as the real plant of biomass gasification. Another parameter set of the MPC is the prediction and control horizons. The pre-

diction horizon is defined as the number of predictions that the prediction model (polynomial regression models for our case) does at each sampling instant. The control horizon is the expected number of manipulations on the manipulated variable (ER in this case) to set the system to the desired reference level at the interval k where k is between 1 and prediction horizon. There is no certain way to determine these horizons. On the other hand, as both of these horizons increase, the required computational effort significantly increases. Moreover, decreasing the value of the prediction horizon tends to make the controller more aggressive which is another undesirable attribute for this study [125]. In this study, the prediction horizon is set to 10 sampling instant and the control horizon is set to 3 sampling instants in order to obtain a functional controller while not extensively increasing the computational burden. Illustration of complete MPC design with the NARXNN model is given in Figure 4.4.

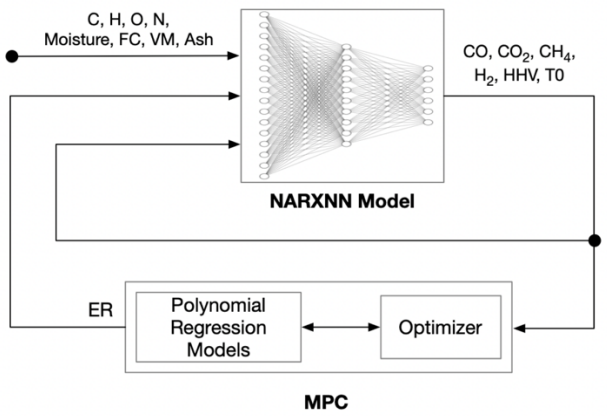


Figure 4.4 Illustration of complete MPC design with NARXNN model.

In order to demonstrate and the proposed complete controller design, an arbitrary reference stabilization problem is defined where the aim is to control and stabilize HHV output at 10 MJ/kg. For this purpose, optimization problem is defined to be minimized as in Equation 4.4 where the square difference between 10 (reference value for HHV) and output of the polynomial model of HHV output. Moreover, because there can't be any negative concentration for product gases in biomass gasification physically, constraints where all output gas concentration have to be greater or equal to 0 are also added. The framework demonstrated in Figure 4.4 has

been run until stabilization of the outputs with the optimization objective given in Equation 4.4. Change in gas concentrations, HHV, T0 and ER variable (generated by MPC) is visually demonstrated in Figure 4.5.

$$J = \frac{1}{2}(10 - M_t^{HHV}(ER))$$
$$s.t.$$
$$M_t^{CO}(ER), M_t^{CO_2}(ER),$$
$$M_t^{CH_4}(ER), M_t^{H_2}(ER), M_t^{HHV}(ER) \geq 0 \quad (4.4)$$

where, $M_t^{CO}(ER), M_t^{CO_2}(ER), M_t^{CH_4}(ER), M_t^{H_2}(ER), M_t^{HHV}(ER)$ indicates the polynomial regression models for CO, CO_2, CH_4, H_2 and HHV outputs, respectively.

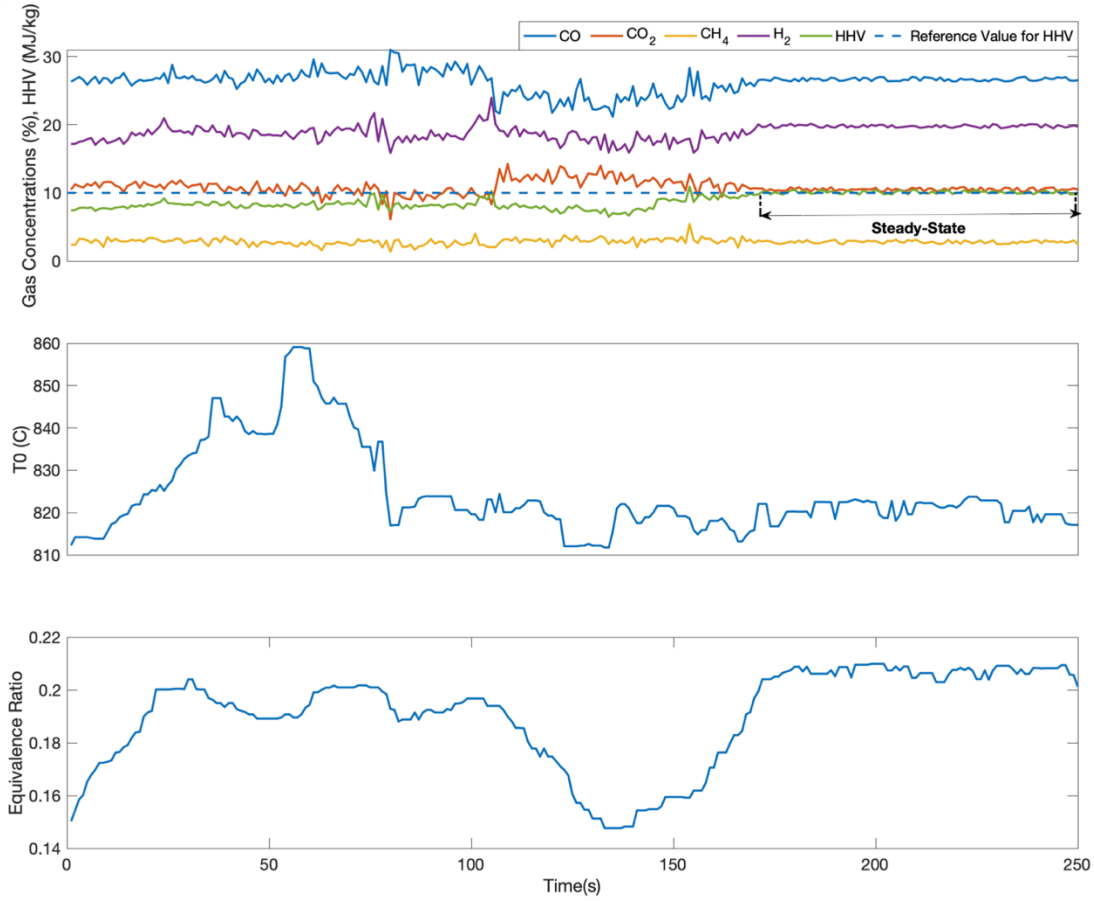


Figure 4.5 Demonstration of MPC design by controlling HHV variable at 10 MJ/kg

When the Figure 4.5 is concerned, ±2 error rate from the reference value which is referred to as steady-state is reached after 172 seconds and constraints given in Equation 4.4 are satisfied throughout the application. Even though the obtained

results look promising, one must also check the viability of these results compared to the experimental data. For this reason, the time period in experimental data where HHV is equal to 10 which is between 1380th and 1581th seconds and we compared experimentally collected outputs, i.e., gas concentrations, T0 and ER variable to the ones obtained during MPC application in that period of time are compared (Table 4.3).

Table 4.3 Range comparison of results obtained from proposed framework with the experimental results for stabilization of HHV.

Output	Experimental	Proposed Framework	Error
CO (vol%)	25.3-26.4	25.9-26.6	$\pm$%1.9
CO_2 (vol%)	10.67-11.33	10.7-11.42	$\pm$%1
CH_4 (vol%)	2.81-3.21	2.76-3.12	$\pm$%2.25
H_2 (vol%)	18.47-19.98	19.45-20.54	$\pm$%4
HHV (MJ/kg)	9.97-10.15	9.98-10.11	$\pm$%0.24
T0 (°C)	802-836	812-819	$\pm$%1.6
ER	0.18-0.2	0.184-0.205	$\pm$%2.35

As one can see in Table 4.3, all output variables as well as the MPC-generated ER variable during MPC application were highly coherent with the experimental data where the error rates for all outputs were $< \pm\%5$. One can assume that the error rates given in Table 4 are acceptable. Therefore, for this specific application, real-time control of the biomass gasification process with the proposed MPC approach performed satisfactorily due to its high consistency with the experimental data and its success as a controller.

In order to test and challenge the proposed MPC design further, rather than stabilizing an output at certain reference value, performance of proposed controller for practical scenarios such as maximization of certain gas concentration are also investigated. Therefore, MPC is implemented to several practical cases, its performance is evaluated and discussed for each case.

4.4 Case Studies

In real-life applications of biomass gasification, a certain tracking of a reference variable would not always be a feasible control objective. One can aim to produce the maximum amount of certain output gas based on the aim of the application. On the other hand, in MPC, a reference value has fed to the cost function in order to control a certain output variable. In order to perform a maximization task, an algorithm is developed that steadily increases the reference value until no significant change in the corresponding output is observed (Algorithm 4.1).

Algorithm 4.1: Increasing reference algorithm for maximization

```
for each 20 seconds do
    increase the reference by 10%
    wait for 19 seconds
    if change in output concentration > ±2%
        then
            continue executing this algorithm
    else
            set reference as final value for Y_ref
            stop executing this algorithm
    end
end
```

One can argue that, rather than increasing the reference value steadily, the polynomial regression model of the desired output can directly be maximized during the optimization. Although this approach can be viable for certain circumstances, direct maximization of the optimization function may cause instability due to the rapidly changing ER variable produced by MPC. Similar instability problems are encountered and acknowledged in the literature [126]. Thus, increasing reference value approach is taken for maximization case studies and Algorithm 4.1 is used for each study case. Average computational expense of the optimization routine for each case study is provided in the Appendix in terms of number of iterations and number of function evaluations. Also, one must note that many different optimization objectives can be defined to utilize the proposed framework and one must develop such objective to their benefits and expectations from the biomass gasification process for energy & valuable chemical production.

4.4.1 Case study 1: maximization of H_2 concentration

Hydrogen is a clean energy carrier can be used in fuel cells to generate electricity, or power and heat. It is possible to utilize hydrogen in almost all sectors such as transportation, industrial production and inside portable energy sources. Due to its importance, the maximum production of it is one of the desired goals during biomass gasification. Therefore, in our first case study, MPC and the optimization problem is adjusted to maximize H_2 production. For this reason, the optimization problem is defined as in Equation 4.5. One must note y_{ref} variable initially defined as the 10% higher than the starting value of H_2 then increases according to the Algorithm 4.1. Resulting outputs vs time and ER vs time plot is given in Figure 4.6.

$$\begin{gathered} J = \frac{1}{2}(Y_{ref} - M_t^{H_2}(ER)) \\ s.t. \\ M_t^{CO}(ER), M_t^{CO_2}(ER), \\ M_t^{CH_4}(ER), M_t^{H_2}(ER), M_t^{HHV}(ER) \geq 0 \end{gathered} \tag{4.5}$$

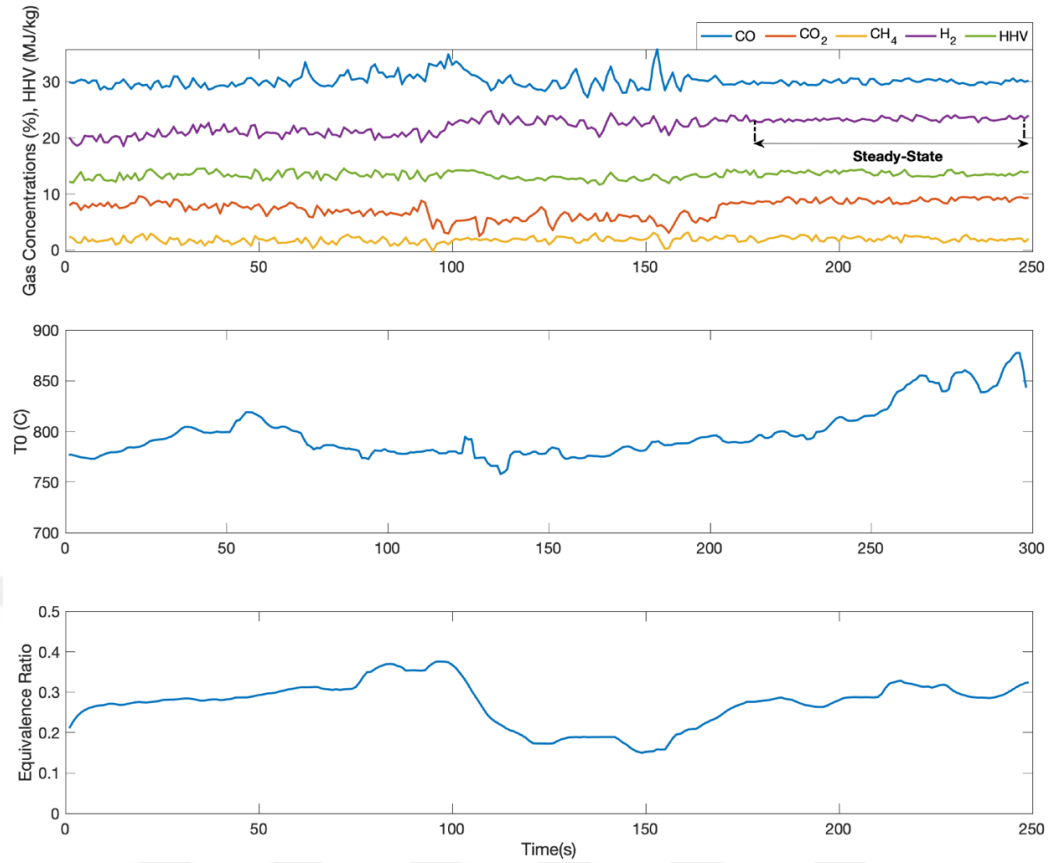


Figure 4.6 Gas concentrations, T0 and ER generated by NARXNN and MPC for maximization of H_2 concentration

As one can see from Figure 4.6, concentration on H_2 was steady from Time = 177 seconds to the end of the simulation. H_2 value was 24 ± %2 during the steady-state which indicates that the maximum amount of H_2 can be obtained at that level according to the MPC and Algorithm 4.1 with the given conditions. When one look at the experimental data to check the viability of these results, the maximum amount of H_2 production was at a time range of 2814-3000 where the H_2 value varied between 24.38% and 24.752%. These results are also compatible to the results provided in Aydin et.al. [127]. Even though these maximum values are somewhat similar, other outputs and ER variable at that time range was checked as given in Table 4.4.

Table 4.4 Range comparison of results obtained from proposed framework with the experimental results for maximization of H_2 concentration.

Output	Experimental	Proposed Framework	Error
CO (vol%)	28.88-29.42	29.51-30.04	±%2.1
CO_2 (vol%)	8.485-9.32	8.732-9.456	±%2.17
CH_4 (vol%)	1.49-2.14	1.36-2.12	±%4.85
H_2 (vol%)	24.38-24.752	24.25-24.412	±%4
HHV (MJ/kg)	13.75-14.31	13.8-14.2	±%0.93
T0 (°C)	797-843	790-864	±%1.72
ER	0.24-0.28	0.24-0.309	±%4.54

When the Table 4.4 is examined, NARXNN and MPC results are coherent with the experimental results up to ±5% error rates. But it is also worth mentioning that, MPC was able to only find and stabilize the maximum H_2 concentration that occurred in the experimental data. One can doubt that is it the actual maximum amount of H_2 concentration that can be produced by the experimental setup that generated the data for this study. Even though there are no references for comparison for the same setup, at worst case, MPC was able to determine the maximum value for H_2 concentration and was able to stabilize it successfully with the given set of data. Moreover, it is logical to assume that the proposed MPC would be capable of find and stabilize the H_2 concentration at different maximum concentrations with the different data set and can be a viable controller method in different biomass gasification applications when the experimental data can be collected.

4.4.2 Case study 2: maximization of CO/CO_2 ratio

Another practical scenario considered in this case study is the maximization of CO/CO_2 ratio. This ratio is proportional to the calorific value of the synthesis gas which is desired to be maximized during biomass gasification. In addition, CO is used in hydroformylation, popularly known as the “oxo” process and it is a building block of many relevant valuable chemical products such as formaldehyde, acetic acid/anhydride, the methyl ethers, methylamine and methyl chloride. Thus, due to the importance of CO/CO_2 ratio, optimization problem that will be solved by MPC is defined as shown in Equation 4.6 to increase this ratio as much as possible.

$$J = \frac{1}{2}(Y_{ref} - (\frac{M_t^{CO}(ER))}{M_t^{CO_2}(ER))})$$
$$s.t. \tag{4.6}$$
$$M_t^{CO}(ER), M_t^{CO_2}(ER),$$
$$M_t^{CH_4}(ER), M_t^{H_2}(ER), M_t^{HHV}(ER) \geq 0$$

It is also worth to mention that, unlike Section 4.4.1, the aim is to maximize a ratio rather than a certain gas concentration in this case study. For this reason, Y_{ref} is initially defined as the initial ratio of CO/CO_2 and the ratio steadily increased according to the Algorithm 4.1. The maximum CO/CO_2 ratio is obtained between Time = 3158 and Time = 3225 in the experimental data where the maximum ratio is equal to 5.411. Moreover, MPC was able to achieve 5.217 and stabilize at that level in steady-state. One must note that maximum ratio that occurred in experimental data was at a smaller time interval compared to other case studies. Experimental outputs and ER are compared to NARXNN and MPC outputs are given in Table 4.5.

Table 4.5 Range comparison of results obtained from NARXX and MPC models with experimental results for maximization of CO/CO_2 ratio

Output	Experimental	Proposed Framework	Error
CO (vol%)	30.81-31.15	30.27-30.46	±1.98%
CO_2 (vol%)	5.721-5.756	5.532-5.559	±3.41%
CH_4 (vol%)	2.351-2.765	2.47-2.832	±3.71%
H_2 (vol%)	24.851-25.312	25.136-25.325	±0.7%
HHV (MJ/kg)	12.22-13.421	12.53-13.07	±2.55%
T0 (°C)	825.5-823	819.98-840.244	±1.37%
ER	0.241-0.266	0.247-0.273	±2.52%
CO/CO_2	5.385-5.411	5.471-5.479	±1.42%

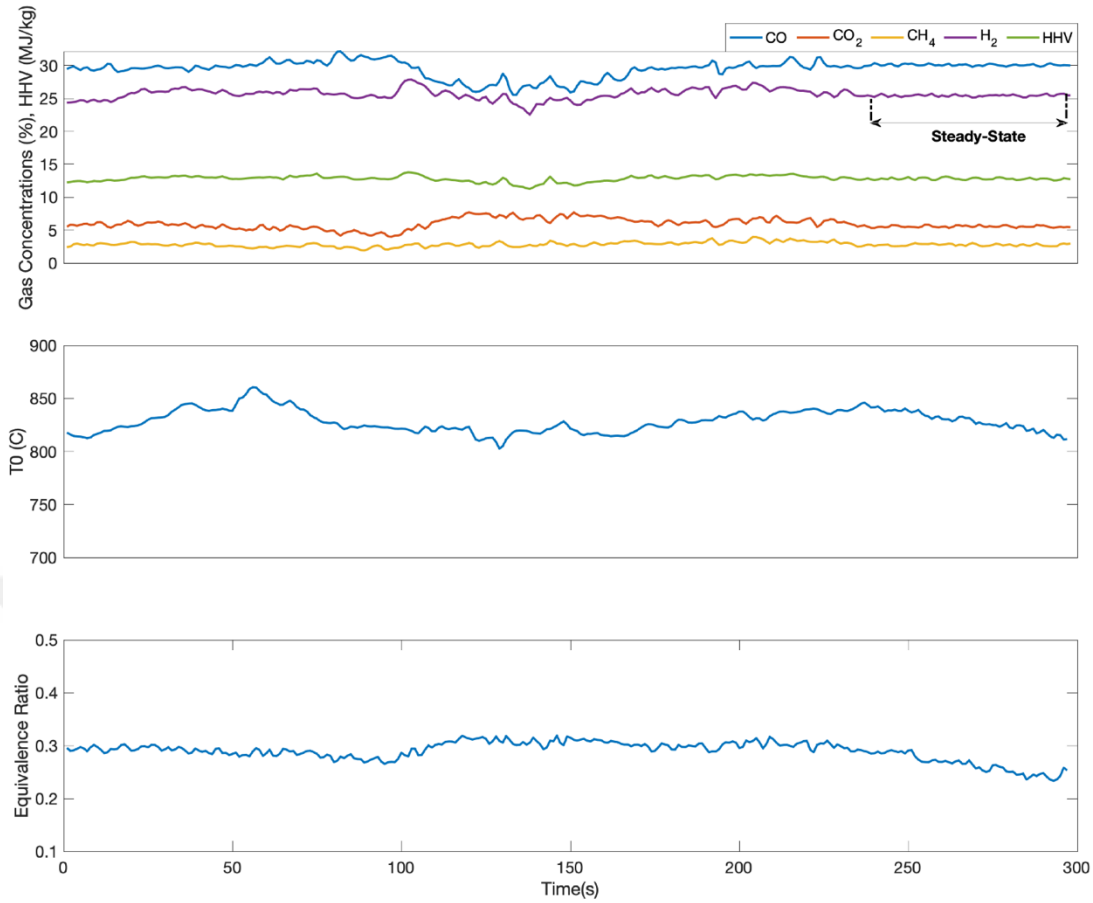


Figure 4.7 Gas concentrations, T0 and ER generated by NARXNN and MPC for maximization of CO/CO_2 ratio

As one can see from Table 4.5 and Figure 4.7, NARXNN and MPC results were coherent with the experimental data up to <±%5 deviation similar to other case studies. And MPC was able to achieve a slightly better CO/CO_2 ratio than the certain interval in experiments where this ratio is at maximum. Even though obtained CO/CO_2 ratios were better in proposed MPC framework, they were no significant change compared to experimental data and MPC was only able to find and stabilize the desired output level occurred in the data set as discussed in Section 4.4.1.

4.4.3 Case study 3: maximization of HHV

HHV, also known as calorific value, is a term generally used for fuels defined as is the amount of heat released during the combustion of a specified amount of the material [127]. It is an important characteristic to determine the energy potential of the corresponding material. In biomass gasification. The amount of HHV obtained throughout gasification indicates the energy potential of the product gases to be used

in further processes such as electricity generation. Thus, it is one of the primary goals of biomass gasification. For this reason, maximization of the amount of HHV is the aim in this case study. Thus, optimization problem is defined almost same as in Section 4.4.1 only for HHV (Equation 4.7). Results are provided in Table 4.1 and Figure 4.8.

$$\begin{gathered} J = \frac{1}{2}(Y_{ref} - M_t^{HHV}(ER)) \\ s.t. \\ M_t^{CO}(ER), M_t^{CO_2}(ER), \\ M_t^{CH_4}(ER), M_t^{H_2}(ER), M_t^{HHV}(ER) \geq 0 \end{gathered} \tag{4.7}$$

Table 4.6 Range comparison of results obtained from NARXX and MPC models with experimental results for maximization of HHV.

Output	Experimental	Proposed Framework	Error
CO (vol%)	29.45-31.063	28.72-29.15	±4.4%
CO_2 (vol%)	7.625-8.145	7.426-7.987	±2.25%
CH_4 (vol%)	2.45-2.58	2.44-2.825	±4.82%
H_2 (vol%)	23.77-25.68	24.06-25.65	±0.66%
HHV (MJ/kg)	14.82-15.02	15.18-15.34	±2.25%
T0 (°C)	795-846	829-855	±2.63%
ER	0.251-0.258	829-855	±2.63%

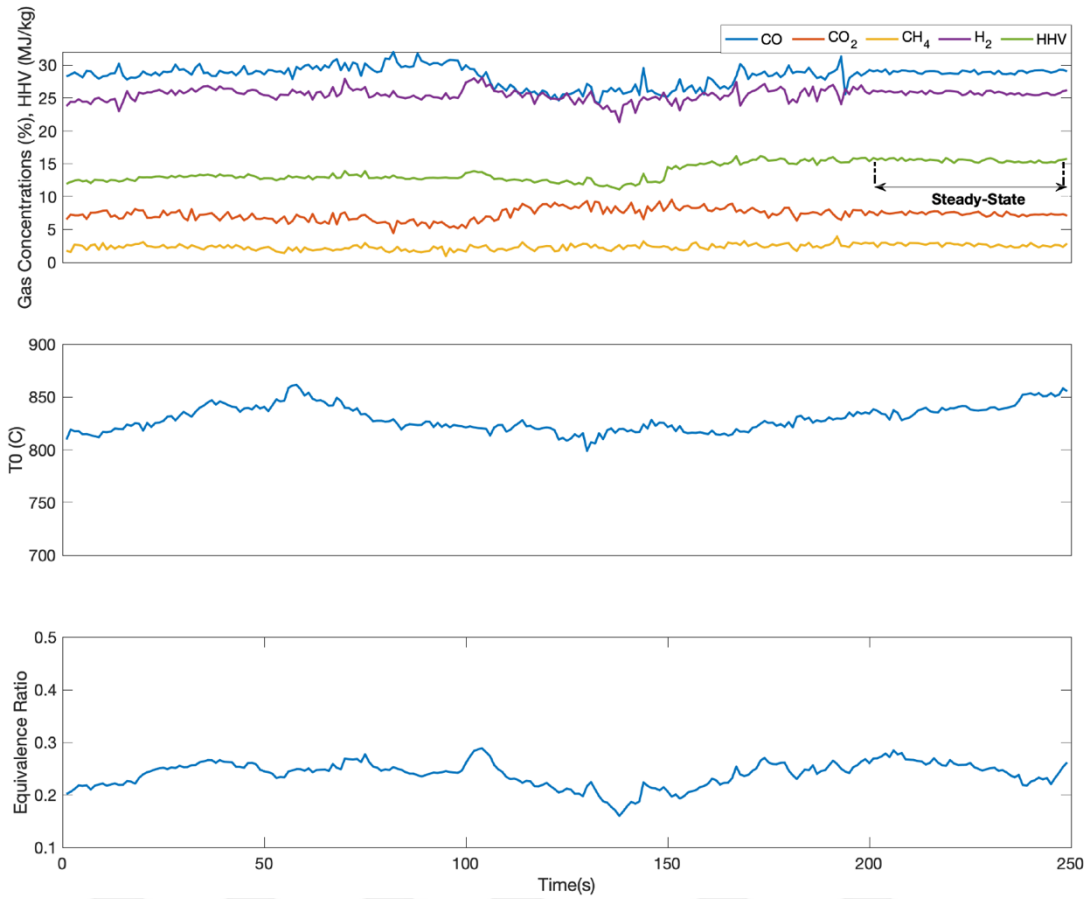


Figure 4.8 Gas concentrations, T0 and ER generated by NARXNN and MPC for maximization of HHV

As one can see from Figure, the maximum amount of HHV obtained during MPC implementation is 15 ± %2 during steady-state, and in experimental data, maximum HHV value occurred during between time = 2798 and time = 2982 while the HHV range was between 14.82 and 15.02. When the Table 4.6 and Figure 4.8 are examined, just like other case studies, deviation between experimental data and proposed model outputs were similar to other case studies which are <±%5 Furthermore, MPC was able to determine the maximum HHV value occurred in the experimental data and managed to stabilize the system at that level successfully.

4.4.4 Case study 4: maximization of CH_4 concentration

CH_4 is another type of energy carrier like H_2 and similarly it has the potential use in almost all industries. Moreover, CH_4 is 4-5 times denser than H_2 which makes it highly attractive from economical perspective. Also, there is an extensive infrastructure for CH_4 (natural gas) in place. So, the no need for new infrastructure

investments makes it a highly desired product. Thus, maximization of CH_4 can be considered as another aim of biomass gasification for certain applications. Therefore, we define a similar optimization problem for other case studies as in Equation (13) to maximize CH_4 concentrations during biomass gasification. Comparison of steady-state outputs of proposed model to the experimental data is given Table 4.7. Resulting gas concentrations, HHV, ER and T0 variables during the study are plotted in Figure 4.1.

$$\begin{gathered} J = \frac{1}{2}(Y_{ref} - M_t^{CH_4}(ER)) \\ s.t. \\ M_t^{CO}(ER), M_t^{CO_2}(ER), \\ M_t^{CH_4}(ER), M_t^{H_2}(ER), M_t^{HHV}(ER) \geq 0 \end{gathered} \tag{4.8}$$

Table 4.7 Range comparison of results obtained from NARXX and MPC models with experimental results for maximization of HHV.

Output	Experimental	Proposed Framework	Error
CO (vol%)	24.51-26.64	24.59-26.11	±1.15%
CO_2 (vol%)	11.12-11.71	11.23-11.27	±2.3%
CH_4 (vol%)	3.08-3.16	3.11-3.32	±3.01%
H_2 (vol%)	17.65-18.11	16.63-17.36	±5%
HHV (MJ/kg)	10.92-11.67	10.78-11.39	±1.84%
T0 (°C)	677-681	623.98-646.85	±5.43%
ER	0.131-0.132	0.125-0.128	±3.75%

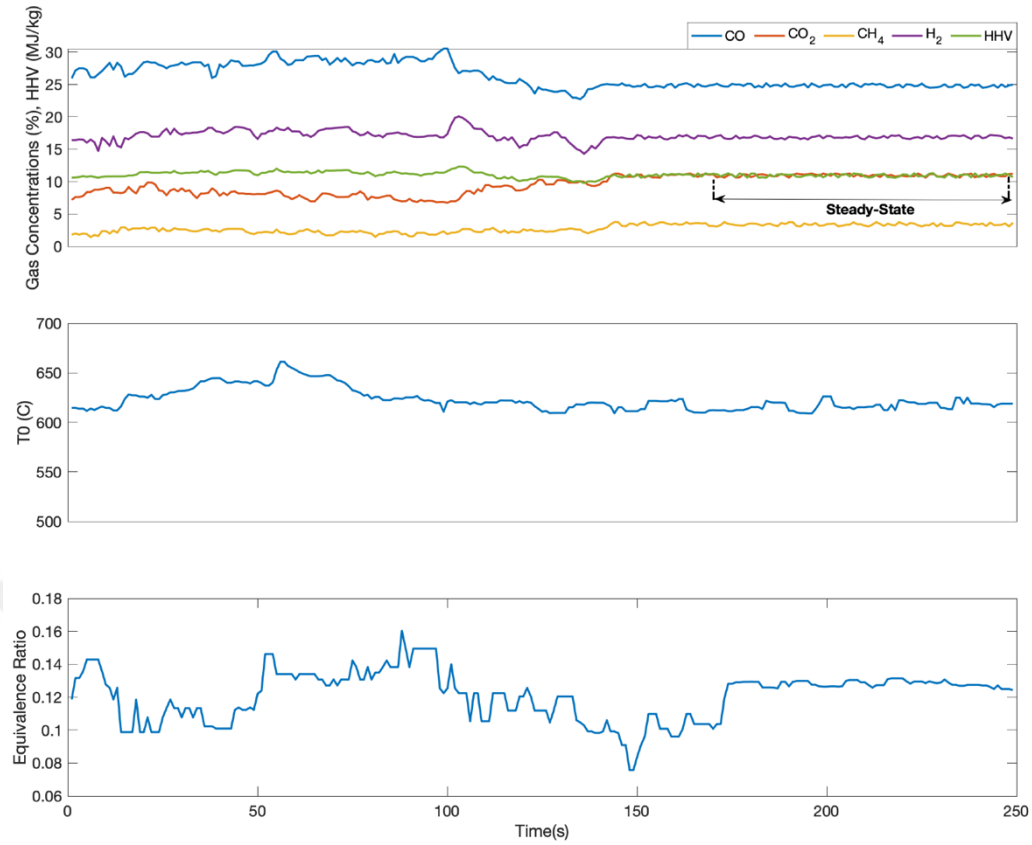


Figure 4.9 Gas concentrations, T0 and ER generated by NARXNN and MPC for maximization of CH4 concentration

When the Table 4.7 is concerned, error rates were again at < ±%5 except T0 output and maximum amount of CH_4 production range in the experiment have reached by MPC in the steady-state which is between 3.08 and 3.16. Just like other case studies, MPC was not able to significantly go out of the maximum level occurred during experimental data collection. On the other hand, CH_4 is the least changing output gas concentration among others during the experiment, and MPC was still able to catch the observed maximum level and managed to stable the CH_4 concentration and adapt accordingly which is a desired feature for a prediction model and for a controller.

4.5 General Discussion

When the overall performance of the NARXNN model is concerned for mimicking the dynamics of the biomass gasification process as well as the MPC for stabilization problem and other maximization-based case studies, there are several observations had to be pointed out. Firstly, NARXNN modeling is proved itself as one of the best if not the best modeling approach for biomass gasification by hitting almost perfect

performance metrics. And due to the utilization of rolling-windows analysis technique, the model is shown to almost perfectly mimic the biomass gasification for data with varying time steps, this performance shows not only good prediction performance but also satisfactory generalization performance. Therefore, it is safe to claim that time-dependency and utilization of output variables from previous time instances are extremely helpful for biomass gasification modeling. Thus, the NARXNN model which is illustrated in Figure 4.2 can be used to accurately predict the output gas compositions, HHV and T0 and can be utilized in simulation studies.

For MPC implementation, polynomial regression is employed to model each output variable to be used during the optimization process. Because the developed polynomial regression models' prediction capability was worse than the NARXNN model due to the low number of features used and polynomial regression being severely less sophisticated algorithm compared to NARXNN. But it was a necessary model simplification to obtain models that have low computational cost due to their part in the optimization procedure of MPC. And because this optimization performed at each sampling instant, these models must require as computation as possible to be used in a practical application where many gasification systems still use microcontrollers that have low to medium computational power. Furthermore, developed MPC is firstly tested in tracking the reference control problem where we aimed to stabilize HHV at 10 MJ/kg. MPC was able to success- fully reach and stabilize HHV at the desired level in the steady-state. Also, other outputs were compatible with the experimental data in the time period where HHV is equal to 10 MJ/kg. In case studies, four different maximization problem is defined with their practical motivation and challenged MPC to solve these problems. Even though results, and obtained maximum concentrations or HHV value, were highly coherent with the experiments, MPC was not able to get out of the boundaries of experimental data for all case studies. This result indicates that, if one desires to control a certain output variable at a certain level and this level is included inside of the training data collected for the development of MPC, it is likely for MPC to find these desired levels and stabilize the output at that level with high success. On the other hand, if that level is not included in that training set as. It can be assumed that MPC may not be able to control the corresponding output at the desired level. The main reason for

this behavior is due to the polynomial regression models' limited modeling capability. If more complex terms are introduced in the models rather than only quadratic combinations of two features, this approach would increase prediction capability but decrease the practical usability due to the higher computational requirement. This trade-off seems to be a primary problem throughout the control applications using MPC in biomass gasification. But this increased computational requirement can be satisfied by using modern solutions such as cloud computing rather than using conventional microcontroller-based circuity. For the proposed MPC design, it seems to be essential for the user to create a data set that includes a wide range of unique samples in order to obtain a controller for different application objectives. For more specific applications, carefully collected and cleaned data set which can describe unique characteristics of the process to some degree would be sufficient to control it with the MPC design we is proposed in this section.

5.CONCLUSION

In this thesis, a ML-based approach is taken to overcome the limitations of the conventional biomass gasification models as well as to expand the practically usable computational frameworks for effectively produce energy via biomass gasification. In Section 2, several prediction models are developed to classify four fuel types, i.e., Coals, Woods, AR and MB with using a dataset collected from literature. Firstly, more conventional approach, flat classification, is taken by employing KNN, RF and SVM classifiers to distinguish four classes at once. Even though all classifiers are managed to classify fuels to some degree, KNN and SVM failed to reach highly satisfactory prediction accuracies. RF, despite its high accuracy level RF, showed stability problems during performance evaluation. Secondly, hierarchical classification methodology is conducted by creating a hierarchical structure between classes by using expert opinion and statistical analyses. Then, classifiers and distinctive features are carefully selected at each stage of the hierarchy and single prediction model is created by merging different classification algorithms. Performance of hierarchical model outperformed flat classification models from both accuracy and stability perspectives by reaching 92% mean accuracy and ±0.17 relative standard deviation in the testing phase. Thus, hierarchical classification proved itself as a viable strategy that can be used in biomass gasification as well as in many fuel applications if it is implemented appropriately. In the Section 3, SVR, DTR, PR and ANN regression methods are employed to predict CO, CO_2, CH_4, H_2 and HHV outputs of the downdraft biomass gasification process by using experimentally collected data set. In order to create prediction models, total of sixteen features are extracted, then reduced to three by using PCA to eliminate possible multi-correlation between features and to reduce computation time during the training phase. MLP and DTR performed significantly better than rest of the methods by achieving $R^2>0.92$ for CH_4, H_2, HHV outputs and $R^2>0.85$ for CO and CO_2 outputs. Performance of all methods declined while predicting CO and CO_2 compared to other outputs, thus, there is a space for improvement for both hyper-parameter and feature selection to increase prediction performance for these outputs. RMSE values obtained from MLP and DTR outperformed the ones obtained from the stoichiometric and non-

stoichiometric modeling and classification-based approaches taken previously that uses the same data set. Obtained results show that machine learning methods can be viable tools to understand, explore and make accurate predictions about the biomass gasification process. MLP and DTR proved themselves as the strongest candidates for this purpose. But due to different complexities of different outputs of the biomass gasification, one method is not sufficient to be used in all of the outputs. The utilization of ensemble learning techniques as well as a different selection of hyper-parameters and features may be a stronger approach. Moreover, developed machine learning-based regression methods, have a low computational requirement to make predictions once they are trained. Thus, these models can be used to make predictions in a simulation environment as well as can be embedded into a microcontroller-based circuitry and utilized in practical applications. In the Section 4, a NARXNN model and MPC for the biomass gasification process is developed by using the same data set that is utilized in Section 3. While creating the NARXNN model, biomass gasification is approached as a time-dependent process (time-series) in order to create a prediction model that can mimic the real dynamics of the process with high precision. Ultimate analysis results, proximate analysis results and ER are used as exogenous inputs, CO, CO_2, CH_4, H_2 concentrations, HHV and T0 values are used as outputs of NARXNN model. Moreover, at each sampling instant, output values from previous sampling instant are fed to the model in order to make NARXNN learn the time-dependent nature of the process. Performance of the NARXNN model evaluated with rolling-windows analysis and $R^2 > 0.98$ is achieved for each output. This concluded that the NARXNN model was managed to understand and mimic biomass gasification even better than the ones proposed in Section 3. Furthermore, MPC implemented to control a certain output variable by treating the NARXNN model as the real plant. For this purpose, polynomial regression models are created for each output variable to make MPC predict the next states of the outputs. Then, optimization problems that are solved at each iteration defined as case studies to test MPC in practical scenarios. The proposed MPC was able to stabilize the outputs at desired levels in all case studies. Although MPC showed several limitations, its' results were highly consistent with the experimental data, thus, MPC can be considered as a strong candidate that can be utilized to

control outputs of the biomass gasification process and possibly in many other energy-based applications for cleaner and more efficient production.

REFERENCES

1. BP. Statistical Review of World Energy [Internet]. Bp. 2019. Available from: https://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-world-energy/electricity.html
2. Panwar NL, Kaushik SC, Kothari S. Role of renewable energy sources in environmental protection: A review [Internet]. Vol. 15, Renewable and Sustainable Energy Reviews. Pergamon; 2011. p. 1513–24.
3. Akia M, Arandiyan H, Yazdani F, Han D, Motaee E. A review on conversion of biomass to biofuel by nanocatalysts. Biofuel Res J. 2015;01(01):16–25.
4. Ud Din Z, Zainal ZA. Biomass integrated gasification-SOFC systems: Technology overview. Vol. 53, Renewable and Sustainable Energy Reviews. 2016. p. 1356–76.
5. Srivastava T. Renewable Energy (Gasification). Adv Electron Electr Eng [Internet]. 2013;3(9):1243–50.
6. Mutlu AY, Yucel O. An artificial intelligence based approach to predicting syngas composition for downdraft biomass gasification. Energy [Internet]. 2018 Dec 15;165:895–901.
7. Demirbas A. Biofuels from agricultural biomass. Energy Sources, Part A Recover Util Environ Eff. 2009;31(17):1573–82.
8. Arabloo M, Bahadori A, Ghiasi MM, Lee M, Abbas A, Zendehboudi S. A novel modeling approach to optimize oxygen-steam ratios in coal gasification process. Fuel [Internet]. 2015 Aug 1;153:1–5.
9. Seo HK, Park S, Lee J, Kim M, Chung SW, Chung JH, et al. Effects of operating factors in the coal gasification reaction. Vol. 28, Korean Journal of Chemical Engineering. 2011. p. 1851–8.
10. Kalina J. Retrofitting of municipal coal fired heating plant with integrated biomass gasification gas turbine based cogeneration block. Energy Convers Manag [Internet]. 2010 May 1;51(5):1085–92.
11. Ali Y, Ozgun Y. An artificial intelligence based approach to predicting syngas composition for downdraft biomass gasification. Energy. 2018;165.
12. Ozgun Y, Mehmet AH. Kinetic modeling and simulation of throated downdraft gasifier. Fuel Process Technol. 2016;(144):145–54.
13. Di Blasi C, Branca C. Modeling a stratified downdraft wood gasifier with primary and secondary air entry. Fuel. 2013;104:847–60.
14. Mendiburu AZ, Carvalho JA, Zanzi R, Coronado CR, Silveira JL. Thermochemical equilibrium modeling of a biomass downdraft gasifier: Constrained and unconstrained non-stoichiometric models. Energy. 2014;71:624–37.

15. Gambarotta A, Morini M, Zubani A. A non-stoichiometric equilibrium model for the simulation of the biomass gasification process. Appl Energy. 2018;227:119–27.

16. Han J, Liang Y, Hu J, Qin L, Street J, Lu Y, et al. Modeling downdraft biomass gasification process by restricting chemical reaction equilibrium with Aspen Plus. Energy Convers Manag. 2017;153:641–8.

17. Aydin ES, Yucel O, Sadikoglu H. Development of a semi-empirical equilibrium model for downdraft gasification systems. Energy. 2017;130:86–98.

18. Jarungthammachote S, Dutta A. Thermodynamic equilibrium model and second law analysis of a downdraft waste gasifier. Energy. 2007;32(9):1660–9.

19. Fani M, Haddadzadeh Niri M, Joda F. A Simplified Dynamic Thermokinetic-Based Model of Wood Gasification Process. Process Integr Optim Sustain. 2018;2(3):269–79.

20. Shayan E, Zare V, Mirzaee I. Hydrogen production from biomass gasification; a theoretical comparison of using different gasification agents. Energy Convers Manag. 2018;159:30–41.

21. Baruah D, Baruah DC. Modeling of biomass gasification: A review. Renew Sustain Energy Rev. 2014;39:806–15.

22. Wang H, Ma C, Zhou L. A brief review of machine learning and its application. Proc - 2009 Int Conf Inf Eng Comput Sci ICIECS 2009. 2009;

23. Ozbas EE, Aksu D, Ongen A, Aydin MA, Ozcan HK. Hydrogen production via biomass gasification, and modeling by supervised machine learning algorithms. Int J Hydrogen Energy. 2019;44(32):17260–8.

24. Xing J, Wang H, Luo K, Wang S, Bai Y, Fan J. Predictive single-step kinetic model of biomass devolatilization for CFD applications: A comparison study of empirical correlations (EC), artificial neural networks (ANN) and random forest (RF). Renew Energy. 2019;136:104–14.

25. Zhu Y, Niu Y, Tan H, Wang X. Short review on the origin and countermeasure of biomass slagging in grate furnace. Front Energy Res. 2014;2(FEB).

26. Uche-Soria M, Rodríguez-Monroy C. An efficient waste-to-energy model in isolated environments. Case study: La Gomera (Canary Islands). Sustain. 2019;11(11).

27. Channiwala SA, Parikh PP. A unified correlation for estimating HHV of solid, liquid and gaseous fuels. Fuel. 2002;81:1051–63.

28. Dunnu G, Maier J, Gerhardt A. Thermal utilization of Solid Recovered Fuels in pulverized coal power plants and industrial furnaces as part of an integrated waste management concept. Appropr Technol Environ Prot Dev World - Sel Pap from ERTEP 2007. 2009;83–91.

29. S. B. Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. Informatica. 2017;31:249–68.

30. Ziegel ER. Handbook of Nonlinear Regression Models. Technometrics. 1991;33(2):240–1.

31. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning 統計的学習の基礎 -データマイニング・推論・予測-. 2017;

32. Guo B, Li D, Cheng C, Lü ZA, Shen Y. Simulation of biomass gasification with a hybrid neural network model. Bioresour Technol. 2001;76(2):77–83.

33. Puig-Arnavat M, Hernández JA, Bruno JC, Coronas A. Artificial neural network models for biomass gasification in fluidized bed gasifiers. Biomass and Bioenergy. 2013;49:279–89.

34. Pandey DS, Das S, Pan I, Leahy JJ, Kwapinski W. Artificial neural network based modelling approach for municipal solid waste gasification in a fluidized bed reactor. Waste Manag [Internet]. 2016 Dec 1;58:202–13.

35. 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 [Internet]. 1995. p. 1137–43.

36. Brown D, Fuchino T, Maréchal F. Solid fuel decomposition modelling for the design of biomass gasification systems. Comput Aided Chem Eng. 2006;21(C):1661–6.

37. Ahmad M, Subawi H. New Van Krevelen diagram and its correlation with t he heating value of biomass. Apex J [Internet]. 2013;2(10):295–301.

38. Galhano dos Santos R, Bordado JC, Mateus MM. Potential biofuels from liquefied industrial wastes – Preliminary evaluation of heats of combustion and van Krevelen correlations. J Clean Prod. 2016;137:195–9.

39. Zhang C, Ho SH, Chen WH, Xie Y, Liu Z, Chang JS. Torrefaction performance and energy usage of biomass wastes and their correlations with torrefaction severity index. Appl Energy. 2018;220:598–604.

40. Zhou H, Long Y, Meng A, Li Q, Zhang Y. Classification of municipal solid waste components for thermal conversion in waste-to-energy research. Fuel. 2015;145:151–7.

41. Ross AB, Jones JM, Kubacki ML, Bridgeman T. Classification of macroalgae as fuel and its thermochemical behaviour. Bioresour Technol. 2008;99(14):6494–504.

42. Rearden P, Harrington PB, Karnes JJ, Bunker CE. Fuzzy rule-building expert system classification of fuel using solid-phase microextraction two-way gas chromatography differential mobility spectrometric data. Anal Chem. 2007;79(4):1485–91.

43. Wang G, Karnes J, Bunker CE, Lei Geng M. Two-dimensional correlation coefficient mapping in gas chromatography: Jet fuel classification for

environmental analysis. J Mol Struct. 2006;799(1–3):247–52.

44. Government of Canada N, Jin J-Z. Development of a national fuel-type map for Canada using fuzzy logic. 2005;406.

45. Qi M, Luo H, Wei P, Fu Z. Estimation of low calorific value of blended coals based on support vector regression and sensitivity analysis in coal-fired power plants. Fuel. 2019;236:1400–7.

46. Saldarriaga JF, Aguado R, Pablos A, Amutio M, Olazar M, Bilbao J. Fast characterization of biomass fuels by thermogravimetric analysis (TGA). Fuel. 2015;140:744–51.

47. Cordero T, Marquez F, Rodriguez-Mirasol J, Rodriguez J. Predicting heating values of lignocellulosics and carbonaceous materials from proximate analysis. Fuel. 2001;80(11):1567–71.

48. Nhuchhen DR, Abdul Salam P. Estimation of higher heating value of biomass from proximate analysis: A new approach. Fuel. 2012;99:55–63.

49. Feng Q, Zhang J, Zhang X, Wen S. Proximate analysis based prediction of gross calorific value of coals: A comparison of support vector machine, alternating conditional expectation and artificial neural network. Fuel Process Technol. 2015;129:120–9.

50. Keller JM, Gray MR. A Fuzzy K-Nearest Neighbor Algorithm. IEEE Trans Syst Man Cybern. 1985;SMC-15(4):580–5.

51. Raghavendra S, Deka PC. Support vector machine applications in the field of hydrology: A review. Appl Soft Comput J. 2014;19:372–86.

52. Lemon SC, Roy J, Clark MA, Friedmann PD, Rakowski W. Classification and Regression Tree Analysis in Public Health: Methodological Review and Comparison with Logistic Regression. Ann Behav Med. 2003;26(3):172–81.

53. Belgiu M, Drăgu L. Random forest in remote sensing: A review of applications and future directions. ISPRS J Photogramm Remote Sens. 2016;114:24–31.

54. Strobl C, Boulesteix AL, Zeileis A, Hothorn T. Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics. 2007;8.

55. Song SL, Bao L, Chen P. Hierarchical text classification and evaluation. Xi Tong Gong Cheng Yu Dian Zi Ji Shu/Systems Eng Electron. 2010;32(5):1088–93.

56. Zimek A, Buchwald F, Frank E, Kramer S. A study of hierarchical and flat classification of proteins. IEEE/ACM Trans Comput Biol Bioinforma. 2010;7(3):563–71.

57. Dekel O, Keshet J, Singer Y. Large margin hierarchical classification. Proceedings, Twenty-First Int Conf Mach Learn ICML 2004. 2004;209–16.

58. Gordon AD. A Review of Hierarchical Classification. J R Stat Soc Ser A. 1987;150(2):119.

59. Nachar N. The Mann-Whitney U: A Test for Assessing Whether Two Independent Samples Come from the Same Distribution. Tutor Quant Methods Psychol. 2008;4(1):13–20.

60. Peterson L. K-nearest neighbor. Scholarpedia. 2009;4(2):1883.

61. Chomboon K, Chujai P, Teerarassammee P, Kerdprasop K, Kerdprasop N. An Empirical Study of Distance Metrics for k-Nearest Neighbor Algorithm. 2015;280–5.

62. Kramer O. K-Nearest Neighbors. 2013;13–23.

63. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20(3):273–97.

64. Abe S. Multiclass Support Vector Machines. 2010;113–61.

65. Anthony G, Gregg H, Tshilidzi M. Image classification using SVMs: One-Against-One Vs One-against-All. 28th Asian Conf Remote Sens 2007, ACRS 2007. 2007;2:801–6.

66. Fletcher T. Support Vector Machines Explained. Online] http//sutikno blog undip ac id/files/2011/11/SVM-Explained.pdf [Internet]. 2009;1–19.

67. Cutler A, Cutler DR, Stevens JR. Random forests. Ensemble Mach Learn Methods Appl. 2012;157–75.

68. Ho TK. Random decision forests. Proc Int Conf Doc Anal Recognition, ICDAR. 1995;1:278–82.

69. Freund, Yoav and LM. The alternating decision tree learning algorithm. Int Conf Mach Learn. 1999;99:124–33.

70. D1Etterich T. Overfitting and Undercomputing in Machine Learning. ACM Comput Surv. 1995;27(3):326–7.

71. Wu DJ, Feng T, Naehrig M, Lauter K. Privately Evaluating Decision Trees and Random Forests. Proc Priv Enhancing Technol. 2016;2016(4):335–55.

72. Kushary D. Bootstrap Methods and Their Application. Technometrics. 2000;42(2):216–7.

73. Ghazi D, Inkpen D, Szpakowicz S. Hierarchical versus flat classification of emotions in text. In: Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text. 2010. p. 140–6.

74. Cesa-Bianchi N, Gentile C, Zaniboni L. Hierarchical classification: combining bayes with svm. In: Proceedings of the 23rd international conference on Machine learning. 2006. p. 177–84.

75. Silla CN, Freitas AA. A survey of hierarchical classification across different

application domains. Data Min Knowl Discov. 2011;22(1–2):31–72.

76. Rodriguez JD, Perez A, Lozano JA. Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE Trans Pattern Anal Mach Intell. 2009;32(3):569–75.

77. Bengio Y, Grandvalet Y. No unbiased estimator of the variance of k-fold cross-validation. J Mach Learn Res. 2004;5(Sep):1089–105.

78. Moore AW. Cross-validation for detecting and preventing overfitting. Sch Comput Sci Carneigie Mellon Univ. 2001;

79. Elmaz F, Yücel Ö, Mutlu AY. Predictive modeling of biomass gasification with machine learning-based regression methods. Energy [Internet]. 2019;116541.

80. Wong T-T. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. Pattern Recognit. 2015;48(9):2839–46.

81. Moreno-Torres JG, Sáez JA, Herrera F. Study on the impact of partition-induced dataset shift on $ k $-fold cross-validation. IEEE Trans Neural Networks Learn Syst. 2012;23(8):1304–12.

82. Tsochantaridis I, Hofmann T, Joachims T, Altun Y. Support vector machine learning for interdependent and structured output spaces. In: Proceedings of the twenty-first international conference on Machine learning. 2004. p. 104.

83. Ghamrawi N, McCallum A. Collective multi-label classification. In: Proceedings of the 14th ACM international conference on Information and knowledge management. 2005. p. 195–200.

84. Khoshgoftaar TM, Golawala M, Van Hulse J. An empirical study of learning from imbalanced data using random forest. In: 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007). 2007. p. 310–7.

85. Gehan EA. A generalized Wilcoxon test for comparing arbitrarily singly-censored samples. Biometrika. 1965;52(1–2):203–24.

86. Sheth PN, Babu B V. Production of hydrogen energy through biomass (waste wood) gasification. Int J Hydrogen Energy. 2010;35(19):10803–10.

87. Jolliffe IT, Cadima J. Principal component analysis: a review and recent developments. Philos Trans R Soc A Math Phys Eng Sci. 2016;374(2065):20150202.

88. Abdi H, Williams LJ. Principal component analysis. Wiley Interdiscip Rev Comput Stat. 2010;2(4):433–59.

89. Büyükçakir B, Elmaz F, Sahin S, Aydin L. Stochastic Optimization of PID Parameters for Twin Rotor System with Multiple Nonlinear Regression. In: 2018 6th International Conference on Control Engineering & Information Technology (CEIT). 2018. p. 1–5.

90. Schmidt M. Least squares optimization with L1-norm regularization. CS542B Proj Rep. 2005;504:195–221.

91. Mohammadi K, Shamshirband S, Anisi MH, Alam KA, Petković D. Support vector regression based prediction of global solar radiation on a horizontal surface. Energy Convers Manag. 2015;91:433–41.

92. Pal M, Mather PM. An assessment of the effectiveness of decision tree methods for land cover classification. Remote Sens Environ. 2003;86(4):554–65.

93. Bhargava N, Sharma G, Bhargava R, Mathuria M. Decision tree analysis on j48 algorithm for data mining. Proc Int J Adv Res Comput Sci Softw Eng. 2013;3(6).

94. Xu M, Watanachaturaporn P, Varshney PK, Arora MK. Decision tree regression for soft classification of remote sensing data. Remote Sens Environ. 2005;97(3):322–36.

95. Jain AK, Mao J, Mohiuddin KM. Artificial neural networks: A tutorial. Computer (Long Beach Calif). 1996;29(3):31–44.

96. Soltani Fesaghandis G, Pooya A, Kazemi M, Naji Azimi Z. Comparison of Multilayer Perceptron and Radial Basis Function in Predicting Success of New Product Development. Eng Technol Appl Sci Res. 2017;7.

97. Noriega L. Multilayer perceptron tutorial. Sch Comput Staff Univ. 2005;

98. Gudise VG, Venayagamoorthy GK. Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. In: 2003 IEEE Swarm Intelligence Symposium, SIS 2003 - Proceedings. 2013. p. 110–7.

99. Kim T, Adali T. Fully complex multi-layer perceptron network for nonlinear signal processing. J VLSI signal Process Syst signal, image video Technol. 2002;32(1–2):29–43.

100. van der Ploeg T, Austin PC, Steyerberg EW. Modern modelling techniques are data hungry: a simulation study for predicting dichotomous endpoints. BMC Med Res Methodol. 2014;14(1):137.

101. Khan FM, Zubek VB. Support vector regression for censored data (SVRc): a novel tool for survival analysis. In: 2008 Eighth IEEE International Conference on Data Mining. 2008. p. 863–8.

102. Ausloos J, Heyman R, Bertels N, Pierson J, Valcke P. Designing-by-Debate: A Blueprint for Responsible Data-Driven Research & Innovation. In: Responsible Research and Innovation Actions in Science Education, Gender and Ethics. Springer; 2018. p. 47–63.

103. Aydin ES, Yucel O, Sadikoglu H. Numerical Investigation of Fixed-Bed Downdraft Woody Biomass Gasification. In: Exergetic, Energetic and Environmental Dimensions. Elsevier; 2018. p. 323–39.

104. Hong X, Mitchell RJ, Chen S, Harris CJ, Li K, Irwin GW. Model selection approaches for non-linear system identification: a review. Int J Syst Sci. 2008;39(10):925–46.

105. Pisoni E, Farina M, Carnevale C, Piroddi L. Forecasting peak air pollution levels using NARX models. Eng Appl Artif Intell. 2009;22(4–5):593–602.

106. Piroddi L. Simulation error minimisation methods for NARX model identification. Int J Model Identif Control. 2008;3(4):392–403.

107. Xie H, Tang H, Liao Y-H. Time series prediction based on NARX neural networks: An advanced approach. In: 2009 International conference on machine learning and cybernetics. 2009. p. 1275–9.

108. Guo WW, Xue H. Crop yield forecasting using artificial neural networks: A comparison between spatial and temporal models. Math Probl Eng. 2014;2014.

109. Aguilar-Lobo LM, Loo-Yau JR, Ortega-Cisneros S, Moreno P, Reynoso-Hernández JA. Experimental study of the capabilities of the Real-Valued NARX neural network for behavioral modeling of multi-standard RF power amplifier. In: 2015 IEEE MTT-S International Microwave Symposium. 2015. p. 1–4.

110. Khamis A, Abdullah S. Forecasting wheat price using backpropagation and NARX neural network. Int J Eng Sci. 2014;3(11):19–26.

111. Gökçen A, \cSahin S. Design of Chaotic System Based Pacemaker on Field Programmable Analog Array Board. In: 2019 Medical Technologies Congress (TIPTEKNO). 2019. p. 1–4.

112. Grüne L, Pannek J. Nonlinear model predictive control. In: Nonlinear Model Predictive Control. Springer; 2017. p. 45–69.

113. Rawlings JB. Tutorial overview of model predictive control. IEEE Control Syst Mag. 2000;20(3):38–52.

114. Del Re L, Allgöwer F, Glielmo L, Guardiola C, Kolmanovsky I. Automotive model predictive control: models, methods and applications. Vol. 402. Springer; 2010.

115. Kothare M V, Balakrishnan V, Morari M. Robust constrained model predictive control using linear matrix inequalities. Automatica. 1996;32(10):1361–79.

116. Dunbar WB. Distributed receding horizon control: Stability via move suppression. Cooperative Control of Distributed Multi-Agent Systems. Wiley Online Library; 2007. p. 63–76.

117. Duriez T, Brunton SL, Noack BR. Machine Learning Control-Taming Nonlinear Dynamics and Turbulence. Springer; 2017.

118. Elmaz F, Yücel Ö, Mutlu AY, Yucel O, Hastaoglu MA, Büyükçakir B, et al. Predictive modeling of biomass gasification with machine learning-based

regression methods. Energy [Internet]. 2019 Jun 30;153(1):847–60.

119. Wang Y, Choi I-C. Market index and stock price direction prediction using machine learning techniques: an empirical study on the KOSPI and HSI. arXiv Prepr arXiv13097119. 2013;

120. Ramasubramanian K, Singh A. Machine Learning Model Evaluation. In: Machine Learning Using R. Springer; 2017. p. 425–64.

121. Kim K. Financial time series forecasting using support vector machines. Neurocomputing. 2003;55(1–2):307–19.

122. Lai RK, Fan C-Y, Huang W-H, Chang P-C. Evolving and clustering fuzzy decision tree for financial time series data forecasting. Expert Syst Appl. 2009;36(2):3761–73.

123. Tangian A, Gruber J. Constructing quadratic and polynomial objective functions. In: Constructing scalar-valued objective functions. Springer; 1997. p. 166–94.

124. Wang Y, Boyd S. Fast model predictive control using online optimization. IEEE Trans Control Syst Technol. 2009;18(2):267–78.

125. Seborg DE, Mellichamp DA, Edgar TF, Doyle III FJ. Process dynamics and control. John Wiley & Sons; 2010.

126. Muske KR, Rawlings JB. Model predictive control with linear models. AIChE J. 1993;39(2):262–87.

127. Aydin ES, Yucel O, Sadikoglu H. Experimental study on hydrogen-rich syngas production via gasification of pine cone particles and wood pellets in a fixed bed downdraft gasifier. Int J Hydrogen Energy. 2019;44(32):17389–96.

APPENDIX

This appendix contains the following related to the Section 4:

- Correlation equations between the output variables' next state and the quadratic combinations of their previous state and ER. These equations and coefficients are generated by the polynomial regression technique which is described in the section. Furthermore, output and actual values vs time plots of rolling-windows analysis are provided to visually demonstrate the prediction performance of each model.
- Number of neurons vs average R^2 value plot
- Average computational expense of the optimization procedures for each case study in terms of number of iterations and number of function evaluations.

Correlation Equations:

1. **Correlation equation and rolling-windows analysis plot of polynomial regression model for *CO* Output**

   Correlation Equation:

$$CO_{t+1} = 0.872 + 0.948CO_t - 0.9ER + (6.126e-04)(CO_t)^2 + (4.77e-02)ER * CO_t - 0.391ER^2 \quad \text{(A.1)}$$

   Rolling-Windows Analysis:

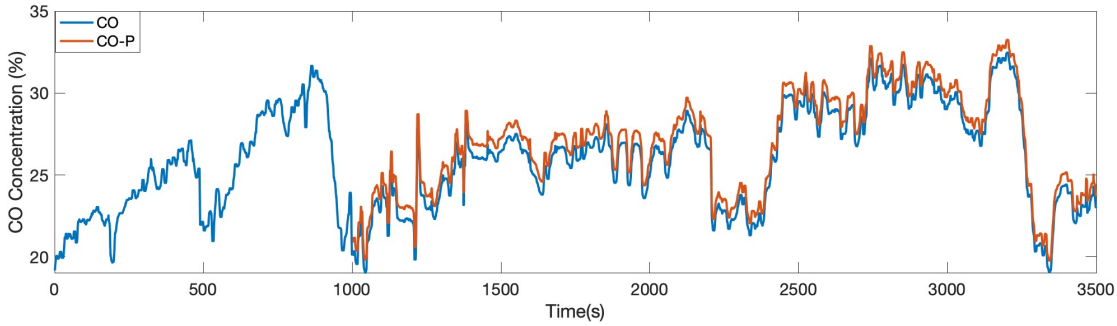


Figure A.1 Polynomial regression predictions vs actual values of CO concentrations

2. **Correlation equation and rolling-windows analysis plot of polynomial regression model for CO_2 output.**

Correlation Equation:

$$CO_{2_{t+1}} = -0.606 + 1.141CO_{2_t} - 0.93ER - (4.038 - 03)(CO_{2_t})^2 - (0.328)ER * CO_{2_t} + 10.44ER^2 \quad \text{(A.2)}$$

Rolling-Windows Analysis:

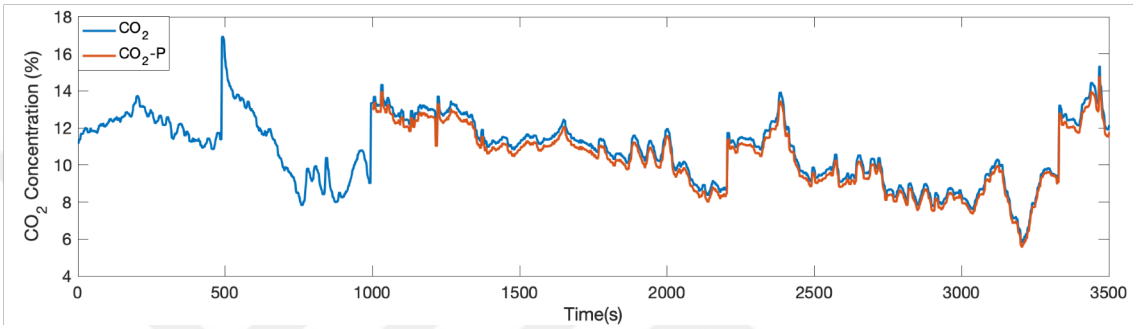


Figure A.2 Polynomial regression predictions vs actual values of CO_2 concentrations

3. **Correlation equation and rolling-windows analysis plot of polynomial regression model for CH_4 output**

Correlation Equation:

$$CH_{4_{t+1}} = -0.01 + 1.08CH_{4_t} + 1.835ER - (2.897e - 03)(CH_{4_t})^2 - (0.62)ER * CH_{4_t} - 5.25ER^2 \quad \text{(A.3)}$$

Rolling-Windows Analysis:

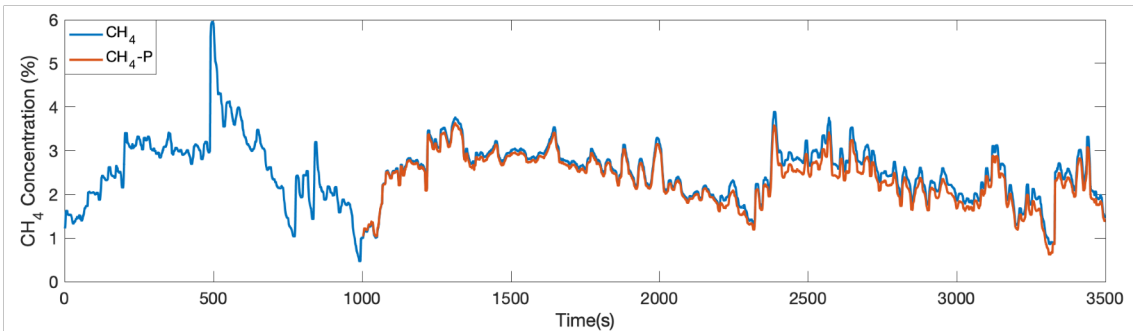


Figure A.3 Polynomial regression predictions vs actual values of CH_4 concentrations

4. **Correlation equation and rolling-windows analysis plot of polynomial regression model for H_2 output**

Correlation Equation:

$$H_{2_{t+1}} = -0.346 + 0.948H_{2_t} + 11.576ER - (9.353e-03)(H_{2_t})^2 + (4.295)ER * H_{2_t} - 38.32ER^2 \quad \text{(A.4)}$$

Rolling-Windows Analysis:

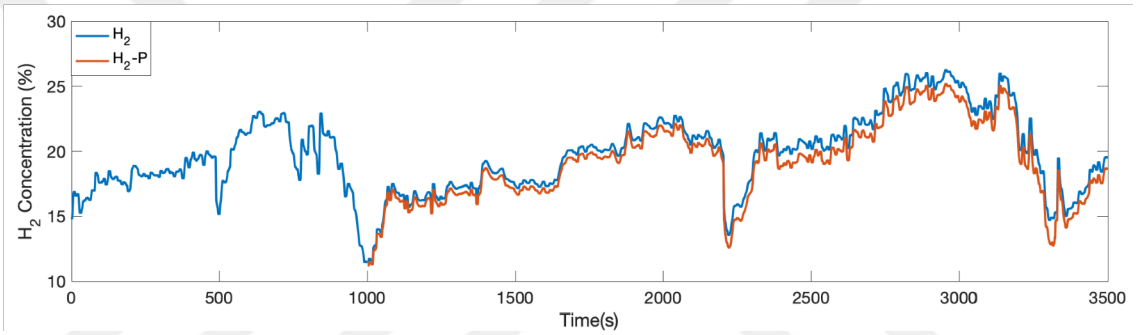


Figure A.4 Polynomial regression predictions vs actual values of H_2 concentrations

5. **Correlation equation and rolling-windows analysis plot of polynomial regression model for *HHV o*utput**

Correlation Equation:

$$HHV_{t+1} = 0.176 + 0.9HHV_t + 5.33ER + (3.535e-04)(HHV_t)^2 + (4.18e-02)ER * HHV_t - 18.32ER^2 \quad \text{(A.5)}$$

Rolling-Windows Analysis:

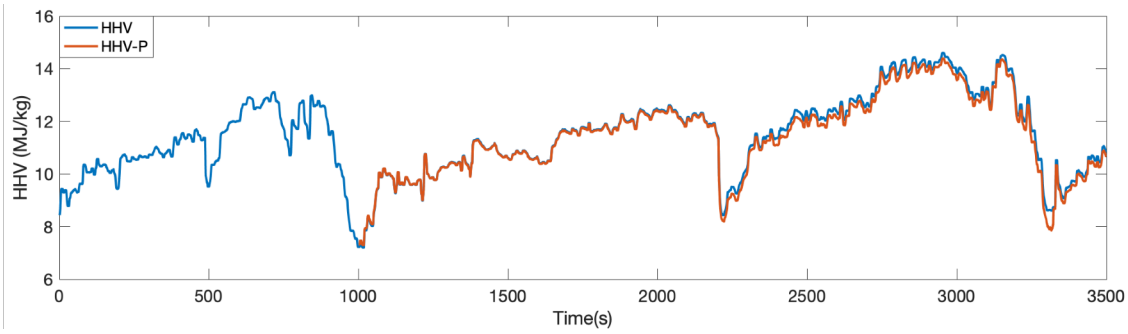


Figure A.5 Polynomial regression predictions and actual values of HHV

Number of neurons on the hidden layer vs average R^2 values plot:

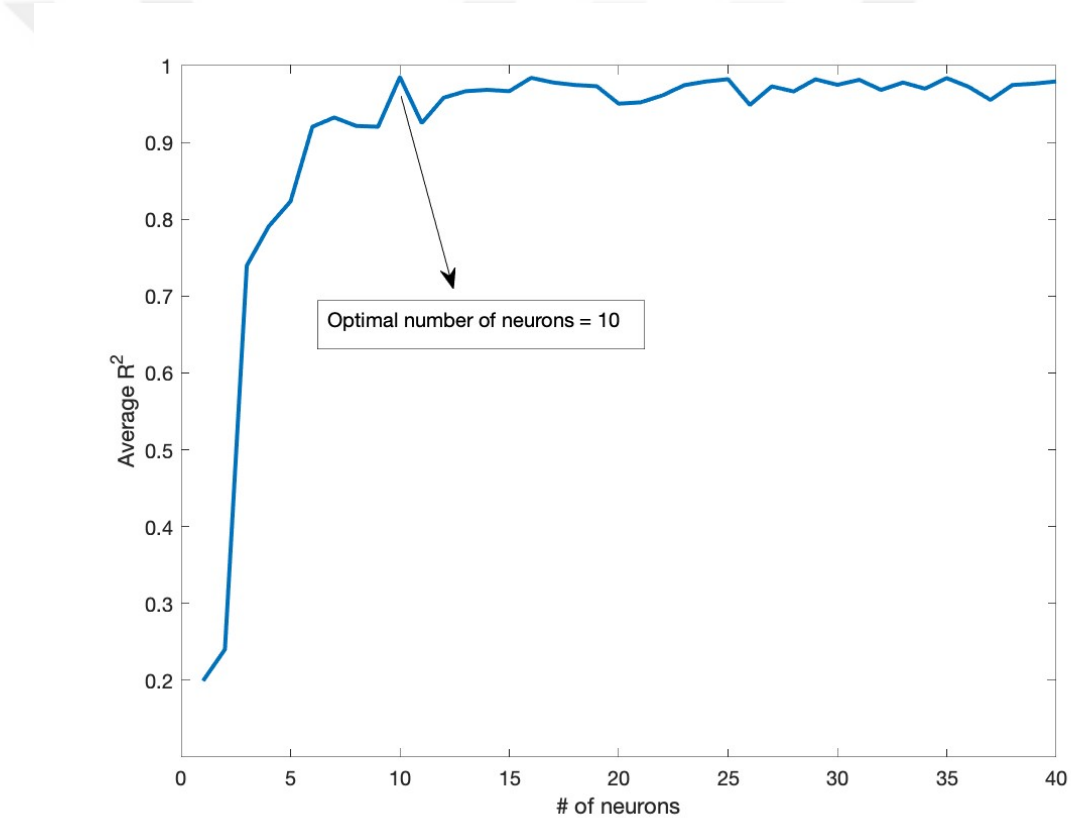


Figure A.6 Number of neurons on the hidden layer vs average R^2 values

Average computational expense of the optimization routine for each case study table:

Table A.1 Average computational expense of the optimization routines

Case Studies	Number of Iterations	Number of Function Evaluations
Case Study 1	17	34
Case Study 2	16	32
Case Study 3	16	32
Case Study 4	15	30

CURRICULUM VITAE

Furkan Elmaz was born in Istanbul, Turkey on 11 November 1996. He completed his B.Sc. degree at Izmir Katip Celebi University in Electrical and Electronics Engineering major with outstanding achievement award. He mainly studied control, embedded programming and machine learning. He was granted by The Scientific and Technological Research Council of Turkey (TUBITAK) undergraduate project and completed it successfully. After graduation, he got accepted to the Master's Degree program of Izmir Katip Celebi University. During his M.Sc. studies, he specialized exclusively on machine learning and its applications on control, energy and health. He contributed in a TUBITAK and a Scientific Research Projects Coordination of Izmir Katip Celebi University-funded projects as a grant-funded researcher. His passion for machine learning drives him to develop unique solutions for many areas of science. He completed his M.Sc. coursework with 4.00/4.00 GPA and on his way to completing his M.Sc. thesis.

List of Contributed Research Projects:

1. ***Forecasting the Outcomes of Biomass Gasification Process by Using Artificial Intelligence-based Methods.***
   Funded by: Scientific Research Projects Coordination of Izmir Katip Celebi University, Turkey by grant 2018-GAP-MUMF-0009

2. ***Epileptic Seizure Classification in EEG Recordings Using Signal Decomposition Techniques.***
   Funded by: TUBITAK by grant 117E137

3. ***Control of Multi-Input Multi-Output Twin Rotor Set with LattePanda-Based Embedded System.***
   Funded by: TUBITAK by grant 1919B11700424

List of Publications:

[1] **Elmaz, F.**, Yücel, Ö. (2020). Data-driven identification and model predictive control of biomass gasification process for maximum energy production. Energy, 117037.

[2] **Elmaz, F.**, Büyükçakır, B., Yücel, Ö., & Mutlu, A. Y. (2020). Classification of solid fuels with machine learning. Fuel, 266, 117066.

[3] **Elmaz, F.,** Yücel, Ö., & Mutlu, A. Y. (2020). Predictive modeling of biomass gasification with machine learning-based regression methods. Energy, 191, 116541.

[4] **Elmaz, F.,** Yücel, Ö., & Mutlu, A. Y. (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 .

[5] Büyükçakir, B., **Elmaz, F.**, Sahin, S., & Aydin, L. (2018). Stochastic Optimization of PID Parameters for Twin Rotor System with Multiple Nonlinear Regression. In 2018 6th International Conference on Control Engineering & Information Technology (CEIT) (pp. 1-5). IEEE.