MODELING AND OPTIMIZATION OF CHEMICAL AND BIOCHEMICAL SYSTEMS USING STATE-OF-THE-ART APPROACHES

A THESIS SUBMITTED TO

SAVITRIBAI PHULE PUNE UNIVERSITY
(SPPU)

FOR AWARD OF DEGREE OF

DOCTOR OF PHILOSOPHY
(Ph.D.)

IN THE FACULTY OF **CHEMICAL ENGINEERING**

SUBMITTED BY

Mr. Makarand Y. Naniwadekar

UNDER THE GUIDANCE OF

Dr. Sanjeev S. Tambe

CHEMICAL ENGINEERING AND PROCESS DEVELOPMENT

DIVISION CSIR-NATIONAL CHEMICAL LABORATORY

Dr. Homi Bhabha Road Pune-411008,

INDIA

November 2018

राष्ट्रीय रासायनिक प्रयोगशाला
(वैज्ञानिक तथा औद्योगिक अनुसंधान परिषद)
डॉ. होमी भाभा मार्ग पुणे – 411 008. भारत
NATIONAL CHEMICAL LABORATORY
(Council of Scientific & Industrial Research)
Dr. Homi Bhabha Road, Pune - 411 008. India.

Certificate of the Guide

This is to certify that the work incorporated in the thesis titled **“Modeling and Optimization of Chemical and Biochemical Systems Using State-of-the-art Approaches”** submitted by **Mr. Makarand Y. Naniwadekar**, for the Degree of Doctor of Philosophy, in Chemical Engineering at Savitribai Phule Pune University (SPPU), Pune, was carried out by the candidate under my supervision in the Chemical Engineering and Process Development Division, National Chemical Laboratory, Pune – 411 008, India. Such material as has been obtained from other sources has been duly acknowledged in the thesis.

Date: Novemer 30, 2018 **Dr. Sanjeev S. Tambe**

(Research Guide)

Declaration by the Candidate

I, hereby declare that the thesis entitled **“Modeling and Optimization of Chemical and Biochemical Systems Using State-of-the-art Approaches”** submitted by me for the degree of Doctor of Philosophy, in Chemical Engineering at Savitribai Phule Pune University (SPPU), Pune, is the record of work carried out by me during the period from January-2012 to November-2018 at the Chemical Engineering and Process Development Division of CSIR-National Chemical Laboratory, Pune-411008, India, under the guidance of Dr. Sanjeev S. Tambe. The thesis has not formed the basis for the award of any degree, diploma, associateship, and fellowship, titles in this or any other University or other institutions of Higher Learning. I further declare that the material obtained from other sources has been duly acknowledged in the thesis.

Date: November 30, 2018 **Mr. Makarand Y. Naniwadekar**

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude to my mentor and architect of the thesis **Dr. Sanjeev S. Tambe** for introducing me to the unexplored and fascinating area of artificial intelligence. His positive approach during my struggling phase, time management skills, strong determination, exemplary commitments, and detailed scrutiny of work at every stage always encouraged and strengthened me to achieve the goals with perfection. His creative thinking and administrative skills continue to inspire me a lot during my Ph. D. work.

I feel privileged to express my thanks to **Dr. Ashwinikumar Nagia**, Director, CSIR-National Chemical Laboratory for allowing me to carry out research work and extending all the necessary infrastructural facilities during my research work at National Chemical Laboratory. My heartfelt thanks are due to **Dr. B. D. Kulkarni**, Distinguished, Scientist, CSIR-National Chemical Laboratory for inspiring me to pursue higher studies and encouraging me to face all the technical challenges with patience and zeal. I am also thankful to **Dr. Sunil Joshi**- Head, CEPD, CSIR-National Chemical Laboratory and **Dr. N. A. Mali**, Scientist, CEPD, CSIR- National Chemical Laboratory for their humble support. My sincere thanks to **Dr. S.P. Kamble**, CEPD, CSIR- National Chemical Laboratory for helping me to verify some of the optimized results experimentally.

My special gratitude goes out to my late **father** and my uncle **Rangakaka**, who instilled in me a desire to learn, strong work ethics, and the value of hard work to achieve my goals.

I am thankful to all my colleagues and staff at CSIR- National Chemical Laboratory, Pune who were quite supportive throughout my research work. Particularly I am indebted to **Shishir Tiwary** who was my closest colleague in most of my research work. Special thanks also to **Dr. Renu Vyas, Dr. Suhas Ghugare, Dr. Vinadevi Patil, Purva, Pratima, Sanket, Aditi, Nikhil, Bharat, Pravin and Mr. Rohit Shetty** for their valuable help as and when required.

I would like to express my sincere thanks towards **management** of my parent organization **All India Shri Shivaji Memorial Society, Pune** for permitting me to pursue higher studies, and the entire staff of Department of Chemical Engineering, for their kind cooperation and support.

Sincere thanks to my wife and son for their unfailing support and caring attitude. I am fortunate to be gifted with our loving daughter whose presence is a great motivation at difficult times.

Thank you all once again.

(Mr. Makarand Y. Naniwadekar)

Dedicated to

MY PARENTS, MY GUIDE AND

ARTIFICIAL INTELLIGENCE SYSTEMS GROUP

(AISG) NCL, Pune

TABLE OF CONTENTS

PREDICTION OF RATE CONSTANT OF PHOTOCATALYTIC DEGRADATION OF PHARMACEUTICAL POLLUTANTS BY MACHINE LEARNING BASED FORMALISMS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

APPENDIX 4

APPENDIX 5

APPENDIX 6

ABSTRACT

Mathematical modeling is needed in all fields of science and engineering systems. It is defined as the task of development of mathematical relation(s) between the dependent/ output/ response and independent/ input/predictor variables in a way such that these are capable of predicting accurately the value(s) of the former. For example, there exists a relation between Cetane number (output) and the biodiesel fuel properties such as flash point, higher heating value, kinematic viscosity and density (predictors). In engineering systems, there exist a number of influential process variables and parameters that need to be related to the variables defining the performance of the system, quality of the product, etc.

Having developed an acceptable and practically usable mathematical model of a system, it can be used to optimize its performance. This is achieved by employing optimization strategies. Optimization is the task of determination of the best solution(s) from among multiple possible ones that would either maximize (minimize) a certain pre-defined profit (loss) function. Typical optimization objective in chemical engineering/technology are for example, minimization of operational costs of a process or maximization of yield, etc.

A sensibly accurate, robust, and reliable mathematical process model saves experimental and operating costs significantly while the optimization improves process performance. Owing to these advantages, a great deal of effort has been expended over the last five decades toward mathematically modeling of chemical and biochemical systems.

Modeling paradigms: There exist a number of modeling approaches such as phenomenological, stochastic, Monte Carlo, cellular automata, statistical, empirical and black-box. Among these phenomenological and empirical models are widely used in practice. The phenomenological or ‘‘first principles’’ models can be constructed from the knowledge of mass, momentum, and energy balances, as well as from other chemical engineering principles such as thermodynamics.

To acquire the necessary information for phenomenological modeling is in general time-consuming, costly, tedious, and often requires extensive experimentation. A common feature of most chemical/biochemical/polymer systems is their non-linear behavior which leads to complex nonlinear models that in most cases are not amenable to analytical solutions; and thus, computationally intensive numerical methods must be utilized for obtaining their solutions.

The above-described multiple difficulties encountered during the construction and solution of the phenomenological models necessitate exploration of alternative modeling formalisms. A commonly employed alternative to the phenomenological process modeling is empirical modeling. Traditionally, this approach entails construction of single or multi-variable linear/nonlinear regression models from the process data. Here, first principles underlying the process are not invoked and modeling is done exclusively using the historical process data. A fundamental deficiency of the empirical modeling approach is that the model structure (form) must be specified a priori. Fulfilling this requisite, especially for nonlinear processes is a time-consuming and cumbersome task since it requires choosing heuristically a suitable model form from numerous alternatives. It is thus seen that due to the difficulties associated with the phenomenological and empirical modeling, there exists a need for approaches that do not require detailed knowledge of first principles underlying a process and at the same time do not require pre-specification of the process data fitting function.

In recent years, state-of-the-art computational intelligence (CI) based methodologies, namely, Artificial Neural Networks (ANNs), Support Vector Regression (SVR), Genetic Programming (GP), Fuzzy Logic (FL) and Genetic Algorithm (GA) have offered a number of attractive avenues for modeling and optimization of engineering processes. These CI-based methods have numerous attractive properties to overcome a number of drawbacks of the traditional phenomenological and empirical modeling techniques. Accordingly, in the present thesis ANNs, SVR, and GP formalisms have been employed for modeling chemical and biochemical systems. Also, GA has been used in obtaining optimal conditions for a reaction involving Photocatalytic Degradation of Pharmaceutical Pollutants.

Each of the above stated CI-based formalism possesses some unique features making them useful in certain specific modeling tasks in chemical and biochemical systems presented in this thesis. The thesis is organized in seven chapters. A brief overview of each chapter is provided below.

Chapter 1 provides a brief outline of the contents of the thesis along with its rationale. The chapter introduces the types of models, their advantages and disadvantages over the others in brief. This chapter also gives the motivation to the application of CI-based formalisms along with its importance to chemical and biochemical engineering systems. The chapter also addresses the application of CI-based methodologies to solve complex real-world problems in brief along with their application areas.

Chapter 2, we introduce the CI-based modeling and optimization methods. This chapter provides a comprehensive report of the conventional methods such as phenomenological, empirical, etc. of modeling chemical and biochemical processes. Further it also gives the advantages and drawbacks of these methods. The chapter also lays the foundation for adopting computational intelligence-based methods and other traditional methodologies for modeling and optimization of chemical and biochemical processes employed in this thesis. The methodologies are classified according to their areas of application such as Modeling Methodologies, for example, Multilayer perceptron (MLP) neural networks, genetic programming (GP), and support vector regression (SVR); Optimization Methodology, for an example, Genetic algorithm (GA). The chapter also gives Principal Component Analysis (PCA) as a Dimensionality reduction method and Steiger's z- test for model discrimination.

Chapter 3 presents the development of artificial neural networks (ANN), support vector regression (SVR) and genetic programming (GP) based models for predicting the photo catalytic degradation (PCD) rate constants from experimental and structural inputs. The models, are developed for three pharmaceutical pollutants considered here namely, Ciprofloxacin, Naproxen, and Paracetamol. To develop the stated models four experimental conditions were considered: time of exposure of pharmaceutical pollutant to the solar radiation, pH of wastewater, concentration degradation during the measured time, and the ratio of degraded concentration to the initial concentration of the pollutant. In the model's input space, the three pharmaceutical pollutants were differentiated using a number of attributes related to their molecular structures. This imparted the model's wider applicability. The molecular structural parameters were considered as model inputs in addition to the above stated four reaction condition parameters. The structural parameters were obtained from Screening Assistant 2 software package and represent the following: number of aromatic atoms, number of atoms, number of bonds, molecular weight, number of heavy atoms, unit stripped logarithm of the solubility, sum of atomic polarizabilities, number of hydrophobic atoms, topological surface area, logarithm of the octanol/ water partition coefficient obtained from linear atom type model, Weiner path, number of rings, logarithm of the octanol/ water partition coefficient of the given structure , molecular refractivity and number of moles.

A sensitivity analysis (SA) was performed to rank the structural inputs according to their influence on the model output. This way top ten most significant parameters were identified. In this study, PCA was performed on the fourteen-dimensional input space consisting of the four experimental parameters and the ten influential structural parameters selected accordingly to the SA. The MLP-, GP- and SVR-based models predicting the rate constant of the PCD, proposed in this study, exhibit an excellent prediction accuracy and generalization performance.

Additionally, the chapter also presents few results of optimization with the objective of obtaining maximum possible degradation of pollutants under milder conditions.

In Chapter 4, the Artificial Intelligence based models have been introduced for the prediction of cetane number of biofuels. It is well-known that the fuel properties of biodiesel, namely, cetane number (CN), kinematic viscosity (KV), density (D) and higher heating value (HHV), play an important role in its deployment for the combustion process. This chapter presents the development of exclusively data driven models using two Artificial Intelligence (AI)-based formalisms, namely, GP and MLP for predicting the cetane number (CN) of biofuels. For predicting CN of biofuels two separate inputs (predictors) are employed- fatty acid methyl ester (FAME) composition and fuel properties. Using the relevant data, two types of CN prediction models are developed in this study: (a) Type I models: FAME composition - 171 biodiesel samples from various research articles and, (b) Type II models: Properties of biofuels - 67 biodiesel samples from various resources.

The CN prediction accuracy and generalization performance of the AI-based models show that the FAME composition-based GP and MLP models exhibit an excellent prediction and generalization performance. The type II models i.e. Fuel property-based models exhibit a reasonably good prediction accuracy and generalization performance.

In Chapter 5, CI-based formalisms have been proposed to predict coal's ash fusion temperatures for the global coal ash samples data from different geographies/ territories. The ash fusion temperatures (AFTs) are significant characteristics of coal ashes and used routinely in the operations of coal-based gasification and combustion processes. Various known models developed for predicting the corresponding AFT values from the chemical composition of coal ashes are complex, tedious, costly, time-consuming, and belonging to mostly a single geographical region. A study of an extensive coal ash chemical composition database from different geographies for each ash phase namely, Initial Deformation Temperature (IDT), Softening Temperature (ST), Hemispherical Temperature (HT) and Flow Temperature (FT) shows the nonlinear relationships between the AFT and weight percentages of some of the chemical and mineral constituents. These nonlinearities are captured using three computational intelligence (CI) based exclusively data-driven formalisms, namely, genetic programming (GP), multi-layer perceptron (MLP) neural network, and support vector regression (SVR) to develop the models for the prediction of four AFTs. In this study the GP possesses several novel and attractive characteristics though it remains much less utilized data-driven modeling technique compared to ANNs and SVR. The predictors for developing the data-driven models are the major oxide constituents of the chemical and mineral composition of the coal ash namely, SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO, TiO_2, and K_2O+Na_2O. Here PCA have been used to reduce the seven-dimensional input space of the AFT models and the pruned dataset was used in the modeling. In the results, the relationships between the seven oxides in the coal ashes and the corresponding AFTs give nonlinear forms by all the four best fitting GP-based models. This study further compares the performance of GP, MLP and SVR models in predicting IDT and ST magnitudes and that of HT and FT values. The CI-based models developed in this study have a potential due to their high AFT prediction accuracy and generalization performance with wider applicability for predicting the AFT values of coal ashes from different geographies in the world.

Chapter 6 reports a study of prediction of initial deformation temperatures (IDT) of Indian coals using CI- based models. IDTs refers to the temperature at which ash just initiates to flow. The currently mined coals in India have a high ash content which are largely used in the power generation industry. This study presents the models for IDT magnitudes for Indian coals with inherently high ash content. The CI-based models developed for IDTs here are likely to be useful in operating coal-based thermal power stations in India. The contents of the seven major oxides found in coal ashes (i.e. SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO, TiO_2, Na_2O+K_2O), were used as inputs. The experimental data on Indian coals were sourced from Central Institute of Mining and Fuel Research (CIMFR), Dhanbad, India. Three CI-based modeling formalisms, namely MLPNN, SVR and GP were employed to develop IDT prediction models. The IDT prediction accuracy and generalization performance of these models was evaluated for three CI-based modeling formalisms- MLPNN, SVR and GP in terms of CC and RMSE values. In this study, the SVR-based model was found to be superior as compared to GP and MLPNN-based models.

Finally, Chapter 7 concludes and summarizes the principal findings of the studies presented in the thesis and provides suggestions for the future work.

CHAPTER 1

INTRODUCTION TO COMPUTATIONAL INTELLIGENCE AND MACHINE LEARNING FORMALISMS

1.1 RATIONALE AND SIGNIFICANCE OF THE STUDY

Chemical reactions/reactors/plants convert raw materials or chemicals into useful valuable forms/products. The chemical industry encompasses numerous sub-areas; the major ones are as given below.

- Bulk chemicals
- Petroleum, petrochemicals and coal/coke-based industry
- Detergents
- Dyes, dyestuff, and paints
- Fertilizers
- Insecticides and pesticides
- Resins, polymers, plastics, and synthetic rubber
- Synthetic fibers and filaments
- Biotechnology and pharmaceuticals
- Food Manufacturing
- Electronics
- Energy generation
- Specialty chemicals

The size of the global chemical industry in revenue terms in the year 2017 was USD 4,378 billion. In the above industries some of the important jobs chemists and chemical engineers perform are:

- Invent, design, and develop new processes and products,
- Construct equipment, instrument and facilities for safe, environment friendly and economical process operation
- Plan and operate facilities.
- Control and optimize the process for efficient, cost-effective and profitable use of the resources

In all these activities modeling of reactions/ reactors/ processes plays an important role.

"*The objective of modeling is to construct, from theoretical and empirical knowledge of a process, a mathematical formulation which can be used to predict the steady-state or dynamic behavior of the process* (Constantinides, 1987)."

Modeling can also be considered as

"*The task of development of mathematical relationship(s) between the dependent/ output/ response and the independent/ input/ predictor variables/ parameters variables of a process in a manner such that these in a generalized way are capable of predicting accurately the value(s) of the former*."

Mathematical modeling pervades all fields of science, engineering and technology. It provides a description of a system/process by employing the mathematical language, principles and procedures. The mathematical model of a process is at best an approximation of the phenomena underlying the real process. In general, models get complex (multiple variables interacting either in a linear or a nonlinear manner) as they describe microscopic phenomena. Typically, the dynamic behavior of a chemical process is described in terms of ordinary and/or partial differential equations, which may be augmented with algebraic equations.

In the present-day chemical reactions/reactors/plants there exists a number of influential process variables and parameters that need to be accounted for during modeling. The principal advantage of modeling are as follows:

- To replace as much as possible the expensive hardware or laboratory analyzers with less expensive and easier-to-maintain software.
- An accurate reactor/ reaction/ process model can be used for predicting process performance under varying sets of operating conditions. This helps significantly in the design, development, optimization and control of the chemical processes and saves the efforts involved in conducting experiments.

In chemical engineering/technology, process models are used mainly for following purposes:

- Steady-state and dynamic process modeling
- Process optimization
- Process monitoring
- Process identification
- Soft-sensor development
- Model based nonlinear process control
- Fault detection and diagnosis
- Nonlinear principal component analysis

Commonly, models for chemical processes involve following inputs (predictor/ causal variables and parameters) and outputs (dependent/ response) variables.

Inputs:

- Reactant, catalyst, promoter, and inhibitor concentrations, inlet temperature, thermodynamic and kinetic parameters, etc.

Outputs:

- Conversion, yield, turn over number, process temperature, operating cost, etc.

1.2 COMMONLY EMPLOYED PROCESS MODELING PARADIGMS AND A NEED FOR THEIR ALTERNATIVES

There exist a number of modeling approaches such as phenomenological, *stochastic*, *cellular automata, statistical*, *Monte Carlo*, *empirical,* and *black-box* for modeling of chemical and chemical engineering/technology systems. Among these phenomenological and empirical are widely used in practice. The phenomenological (also termed ‘‘first principles’’ or “mechanistic”) models can be constructed from the knowledge of the physico-chemical phenomena underlying the system under consideration. This typically involves taking in consideration the mass, momentum, and energy balances, physical property relationships, as well as other chemical engineering principles such as thermodynamics. The advantages of the first principles models are: (a) Their structure can be interpreted in terms of the phenomena they describe, and (b) they can be used in extrapolation and scale-up. On the other hand, first principles models suffer from following drawbacks (Czop et al., 2011; Nandi et al., 2002).

- Their development involves a significant amount of experimental effort and the associated costs and time for securing the mechanistic and other information and data such as kinetic parameters and heat and mass transfer coefficients.

- The modern-day chemical processes are often complex and nonlinear and consists of a plethora of interconnected equipment. This poses significant difficulties in getting a good understanding of their steady-state and dynamic behavior, which is needed for the development of phenomenological models.

- A large number of chemical and chemical engineering/technology systems behave in a nonlinear manner. Rather, nonlinear behavior is a law rather than an exception for these systems. This results in multivariable nonlinear models involving ordinary and partial differential equations that in most cases cannot be solved analytically. In such cases computationally intensive numerical methods are needed for their solutions.

- Complexity of a mechanistic model varies with the underlying assumptions and, therefore, these must be must be outlined beforehand.

Owing to the difficulties in developing first principles models, an alternative namely empirical modeling is commonly employed in process modeling. Most often this approach comprises constructs single or multi-variable linear/nonlinear models using corresponding regression techniques and the process data. It may be noted here that unlike first principles modeling, the empirical modeling rarely invokes the actual physico-chemical phenomenon underlying the chemical process. That is, modeling is conducted exclusively using the historical process data. The major drawback of the empirical process modeling is that the specific structure (form) of the data-fitting function (model) needs to specified before estimation of the function parameters could be attempted (Nandi et al., 2002). Invariably, owing to the nonlinear and complex nature of the modern-day chemical processes, pre-specification of the data-fitting function turns out to be a time-consuming, and cumbersome task especially since it comprises selecting by trial and error a suitable model form from a large number of alternatives (Nandi et al., 2001).

Owing to the difficulties pertaining to the phenomenological and empirical modeling approaches there arose a need for methods that do not (a) require the detailed knowledge of the physic-chemical phenomena underlying a process, and (b) compel pre-specification of the linear/nonlinear form of the data-fitting function.

1.3 COMMON PROCESS OPTIMIZATION PARADIGMS AND A NEED FOR THEIR ALTERNATIVES

Optimization is the task of determination of the best solution(s) from among multiple possible ones that would either maximize (minimize) a certain pre-defined profit (loss) function. Optimization is done to determine what values of process inputs should be used to obtain the desired process output. Like calibration, optimization involves substitution of an output value for the response variable and solving for the associated predictor variable values. The process model is again the link that ties the inputs and output together. Typical optimization objective in chemical sciences and engineering/technology are, for example, (a) minimization of operational costs of a process, (b) maximization of product quality and quantity, (c) minimization of energy costs during process operation, (d) maximization of conversion/yield and selectivity of the desired product, and (e) minimization of resources such as raw materials/ feed, labor and utilities.

A reasonably accurate model saves experimental and operating costs significantly while optimization of model improves process performance leading to, for example, minimization of operating cost, energy savings, reduction in pollution, better product quality etc. Consider, for instance, a typical furnace or a boiler system wherein process optimization involves determination of the optimum operating and feed conditions so as to obtain maximum heat energy from minimum amount of reactants/fuel/feed. In order to conduct this optimization, an accurate boiler model that predicts the heat energy as a function of the operating and feed conditions is necessary.

Conventionally, process optimization is conducted using deterministic gradient based algorithms. These algorithms such as the widely employed gradient descent and conjugate gradient need computation of the derivative of the objective function (to be minimized/maximized) at each step towards an optimal solution. Most of these algorithms also suffer from following drawbacks.

(a) Invariably, in the case of nonlinear objective functions, these methods get stuck into a local minimum instead of reaching the global minimum on the objective function surface.
(b) Require the objective function to be smooth, differentiable and continuous.

Since more often than not, the optimization problems encountered in chemical engineering / technology consists of nonlinear objective functions possessing multiple minima, the standard gradient descent based deterministic function maximization/minimization algorithms do not yield optimum solutions. Also, while the objective function may be continuous and differentiable, there is no guarantee that it will be smooth. Owing to these difficulties in respect of the conventionally used gradient descent methods there was a need to explore alternative methods that are lenient towards the nature of the objective function and capable of reaching the deepest or the global minimum on the objective function surface.

In recent years, state-of-the-art computational intelligence (CI) based methodologies, namely, *Artificial Neural Networks* (ANNs), *Support Vector Regression* (SVR), *Genetic Programming* (GP), *Fuzzy Logic* (FL) and *Genetic Algorithm* (GA) have offered a number of attractive features for modeling and optimization of chemical, biochemical and polymer processes. These methods have several attractive properties and overcome a number of drawbacks of the conventional phenomenological and empirical modeling techniques as also commonly used gradient based optimization methods.

The principal tenets of the modern day's chemical industry practice involve safe, "Green", energy efficient production with improved productivity. Prediction of the steady-state and dynamic reaction/reactor/process behavior, prediction of the phase diagrams, preventative maintenance and asset management, optimization of process conditions, soft-sensor development, load forecasting and fault detection and diagnosis, are some of the techniques available for improving the productivity of today's chemical plants. A brief description of these tasks is given below (Krishnan, 2017; Tambe et al., 1996). In a number of these tasks, the state of art artificial/computational intelligence formalisms namely ANNs, GP, SVR and GA, have been utilized for performing modeling and optimization.

1. **Prediction of the steady-state and dynamic process behavior**: An enormous amount of steady-state and dynamical data regarding the process operating (input) conditions (temperatures, reactant and catalyst concentrations, flow-rates, etc) and the corresponding performance (output) variables (conversion, selectivity, yields, etc) are measured and

archived. These data can be used to establish appropriate linear/nonlinear mathematical relationships between the operating conditions (predictors) and the corresponding performance (response) variables using a variety of artificial and computational intelligence formalisms. These models can be developed online as also offline and subsequently used in tasks such as process optimization, fault detection and diagnosis and model-based control.

2. **Phase diagram prediction**: Using models describing the phase diagram, it is possible to predict the conditions at which thermodynamically distinct phases occur, and also coexist at an equilibrium.

3. **Preventative Maintenance and Asset Management**: Multiple sensors are fitted on the chemical process equipment such as reactors, compressors, boilers and heat exchangers which measure a continuous stream of data that gets archived. Big data analytics which utilizes various techniques including artificial intelligence ones makes use of the stated data for identifying/classifying patterns and thereby predict beforehand the potential malfunctions/breakdowns of the equipment and prevent equipment failures.

4. **Process optimization**: The goal of process optimization is determination of values of process inputs (operating variables & parameters) to be used in the process operation to obtain an optimal process performance. Some optimization goals in process engineering are maximization of conversion and or product yield, maximization of conversion and or product yield, maximization of process profit, minimization of operating loss, minimization of production cost, and minimization of selectivity of the undesirable products/energy requirement. In recent years artificial intelligence-based methods have been extensively employed in performing process optimization.

5. **Soft-sensors:** Soft-sensors are inferential mathematical models capable of continuous estimation of properties that are infrequently measured in a laboratory or where online property measurement is not possible. Their advantages are: (i) lower maintenance and capital investment of analyzers, (ii) operate the plant “without-being-in-dark” between lab measurements, and (iii) reduced lab sampling frequency. In a typical chemical plant, a number of dynamic operations take place concurrently. This poses difficulties in selecting the optimal operating conditions. The state of the art “Smart” technologies utilizes soft sensors for enabling control of the nonstandard process variables to improve energy efficiency.

6. **Load Forecasting**: Artificial/computational intelligence methods have been used to forecast the demand of power and energy requirement of chemical plants that varies depending upon the time of the day, season, weather conditions, etc. Typically, the historic demand-supply data are used to develop load forecasting models which are significantly useful also in planning the inventory of the raw-materials such as oil, natural gas, coal, air, water, metals, minerals, etc. necessary to fulfill the demand of power and energy.

7. **Fault detection and diagnosis**: Chemical processes comprise a complex network of equipment handling and processing materials, and energy. Over time, the process performance degrades albeit slowly owing to the wear and tear of the process equipment and components. It is extremely important to detect and diagnose faulty/abnormal process behavior from the viewpoint of plant and personnel safety as well as avoiding catastrophic events and associated economic loss. In recent years, artificial neural networks have found a niche in performing fault diagnosis and detection (FDD) in chemical processes. Specifically, their tremendous power in recognizing patterns in the data is exploited for conducting FDD. ANNs can efficiently perform pattern recognition (clustering) and this

ability is used to differentiate between the “normal” and “abnormal” process behavior based upon the sensor data.

The major advantages of the AI/CI-based modeling methods are as follows

- They are exclusively data-driven methods, that is, modeling is conducted on the basis of historical, and/or current process data. Unlike phenomenological modeling, the data-driven modeling does not require information such as mass, momentum, and energy balances, physical property relationships, and thermodynamics and therefore model development needs mush less efforts and time.
- Given adequate and representative process data (example set) consisting of predictor (model inputs) and response (model outputs) variables, ANNs are **capable of approximating the relationships between the inputs and the output to an** arbitrary degree of accuracy (Girosi and Poggio, 1990).
- While ANNs and SVR are ideal for approximating nonlinear input-out relationships, GP can efficiently model both linear as well as nonlinear inter-dependencies.
- Unlike commonly employed regression models, it is not necessary to pre-specify the form (structure) of the ANN, SVR and GP based models thus saving the time-consuming and tedious trial-and-error approach involved in the selection of the data-fitting model.

The data driven models also have some negative features although the advantage stated above outweighs them: (a) a large number of unknown parameters may be associated with the data-driven models, which need to be estimated from the process data (termed “example set”) consisting of the predictor and response variables, (b) in general, these models cannot be used for extrapolation since being exclusive data-driven, their prediction and generalization performance is reliable only in the range of the predictor-response variable data considered in the model development; this essentially means that the data-driven models are good for interpolation, (c) being black-box models, it is difficult to interpret the structure and parameters of the ANN and SVR models in terms of the example data or the physico-chemical phenomena that they describe, and (c) the structure of a data-driven model is subjective; that is different data-driven modeling paradigms can come up with process models possessing different forms (structures) with variable number of related parameters although their prediction and generalization performance is comparable.

Owing to their numerous advantages, in the present thesis ANNs, SVR, and GP formalisms have been employed for modeling selectively chemical and biochemical systems. Also, GA has been used in obtaining optimal conditions for a reaction involving Photo-catalytic Degradation of Pharmaceutical Pollutants. The characteristic features of the CI-based modeling methods used in the thesis are provided below.

1.3.1 Artificial neural networks

Artificial neural networks (ANNs) are devised on the basis of the functioning of the naturally occurring network of neurons in the human brain. Specifically, these are based on the concept that a highly interconnected system of simple processing elements (called neurons or node) can approximate complex nonlinear relationships existing between input and output variables to an arbitrary degree of accuracy (Patel et al., 2006). The ANN paradigm namely *Multilayer Perceptron Neural Networks* used in the thesis is an exclusively data-driven nonlinear function approximation technique. This means that the phenomenal information processing capability of MLP arises from its multilayer architecture housing artificial neurons (processing elements/nodes) that are linked using weighted synaptic connections. When the flow of information occurs in forward direction only, it is a feed-forward structure MLP. Most often, it contains three layers namely *input, hidden,* and *output* layers. It is also possible to consider multiple hidden layers in an MLP architecture. Each node in its hidden layer processes incoming information using *logistic sigmoid*, a nonlinear transfer function to compute the output. The much-desired nonlinear input-output mapping capability of an MLP is due to the said nonlinear processing performed by its hidden layer nodes. Given an example data, consisting of input and the corresponding output vectors, an MLP can learn intricate nonlinear input-output relationships. The training or learning is directed using a appropriate learning algorithm (such as the *error-back-propagation* algorithm (Rumelhart et al., 1986) that optimizes the interlayer connection weights such that the error between MLP computed output and its desired (target) magnitude (known as *prediction error*) is minimized. The notable features of MLP can be summarized as:

- It can be developed solely from the process input output data without invoking process phenomenology.

- Even multi-input multi output relationships (MIMO) can be approximated easily.

- The MLP-based models possess generalization ability owing to which the model can accurately predict the outputs corresponding to a new set of inputs that were not part of the data used for constructing the MLP model.

- The MLP-based models are robust and fault tolerant. They can, therefore, recall full patterns from incomplete, partial or noisy patterns.

- The MLPs can process information in parallel, at high speed, and in a distributed manner. Their notable application domains can be listed as: nonlinear function approximation, supervised pattern recognition and image processing.

- MLPs have shown remarkable progress in the recognition of visual images, handwritten characters, printed characters, speech and other PR based tasks.

- Control Systems: A large class of computer products, chemical plants, robots, space science and technology, military applications, mechatronics etc. are the current examples of MLPs applications in today's scenario.

A representative tabulation of MLP applications in chemical engineering/ technology is provided in **Table 1.1.**

Table 1.1: A representative survey of ANN/MLP–, SVR-, GP-, and GA-based modeling and classification applications

CI Formalism	**Application**	**Reference**
ANNs/ MLPs	Modeling of catalytic manufacturing process	Chitra (1993)
	Nonlinear multivariate calibration	Bos et al (1993)
	Prediction of brewing fermentation	Syu et al (1994)
	Modeling of coke fractionators	Blaesi and Jensen (1992)
	Identification of gases	Niebling (1994)
	Estimation of GCV of coal	Patel et al (2007)
	Modeling of High Ash Coal Gasification in a Pilot Plant Scale Fluidized Bed Gasifier	Patil-Shinde et al (2014)
	Counter-propagation Neural Networks for Fault Detection and Diagnosis	Vora et al (1997)
	Prediction of All India Summer Monsoon Rainfall Using Error-back-propagation Neural Networks	Venkatesan et al (1997)
	Modeling and monitoring of batch processes using PCA assisted GRNN	Kulkarni et al (2003)

MLPs exhibit a phenomenal capability to predict situations from past trends. In addition to Chemical engineering MLP neural networks have been utilized for example in meteorology (Mielke, 1984), stock market (Hassan et al., 2007), banking (Li and Ma, 2010) and econometrics (Kuan and White, 1994) with high success rates.

1.3.2 Support vector regression

Support vector machine (SVM) is a statistical learning-based formalism for conducting supervised nonlinear classification (Vapnik, 1995). To conduct the alleged classification, SVM maps the coordinates of the objects into a high dimensional feature space using nonlinear functions (termed *features* or *kernels*) wherein two classes can be separated with a linear classifier as done typically. Support vector regression utilizes same principles albeit for performing a nonlinear regression. That is, it attempts to find a multiple input – single output (MISO) function with following attributes: (a) it is located with the maximum deviation from all the training data, and (b) is as flat as possible. A function with a small weight vector assures its flatness. The salient features of the SVR are given below (Sharma and Tambe, 2014).

- The problems to find the solution in presence of multiple local minima can be avoided by SVR. For this task, SVR minimizes a quadratic function with a single minimum.
- Assures good generalization ability, robustness of the solution and sparseness of the regression function, and an automatic control of the solution complexity.
- A precise understanding of *support vectors*, play a foremost role in defining the regression function assists in the exposition of the regression model in terms of the training data.

The noteworthy applications of SVMs can be seen to solve various real-world problems:

- SVMs are helpful in information science, library science and computer science for text and hypertext categorization.
- The hand-written characters/ signatures can be recognized using SVM
- The SVM algorithm has been widely applied in the medical, biological, environmental and other sciences.
- SVM have been proved best for images classification.
- In some applications of soft sensors SVM have proved best performance results.

The significant applications of SVM/ SVR in chemical engineering/ technology are specified in **Table 1.2**

Table 1.2: A representative survey of SVR-based modeling and classification applications

CI Formalism	**Application**	**Reference**
SVM/ SVR	Soft-sensor development - fed-batch bioreactor	Desai et al (2006)
	Catalyst Development	Valero et al (2009)
	Applications of SVR in chemistry	Ivanciuc (2007)
	Analysis of LDA time-series	Gandhi et al (2008)
	Modeling and Optimization of pilot plant study of cumene synthesis	Nandi et al (2004)
	Forecasting of coal seam gas content using SVR based on particle swam optimization	Meng et al (2014)
	High Ash Char Gasification in Thermo-Gravimetric Analyzer and Prediction of Gasification Performance Parameters	Patil-Shinde et al (2016)

	A novel modeling approach to optimize oxygen-steam ratios in coal gasification process	Arabloo et al (2015)
	Selecting Predictor Variables and Modeling in Process Identification	Verma et al (2016)
	Optimization Techniques for Improving the Performance of Information Retrieval System	Badhe et al (2014)

1.3.3 Genetic Programming

Genetic Programming (GP), belongs to a class known as “evolutionary algorithms” that follow the principal tenet—commonly paraphrased as “survival of the fittest” — of Darwin’s theory of evolution along with the genetic propagation of characteristics. Initially, GP was proposed [Koza, 1992] to develop automatically computer programs doing pre-specified tasks. It’s another application known as “symbolic regression (SR),” is of interest to this study. The novel attribute of the GP-based SR is, provided an example data set having function inputs and the corresponding outputs, it has the capability of searching as also optimizing the specific structure (form) of an appropriate linear or nonlinear data-fitting function and all the related parameters. And, unlike MLP neural network and SVR formalisms, the GP approach performs the stated search and optimization without making any assumptions about the structure of the linear or nonlinear data-fitting function. GP is a CI-based exclusively data-driven modeling paradigm, for automatically generating computer programs that perform pre-defined tasks (Koza, 1990). Another important application of GP is known as “symbolic regression (SR)” which is used in the thesis for modeling purposes. The characteristics of GP based SR are given below.

- Generates an initial population of candidate expressions/models with pure stochastic manner. That is, unlike similar techniques performing data-driven modeling, namely multilayer perceptron (MLP) neural network, and support vector regression (SVR), GPSR does not make any assumptions about the form and parameters of the data-fitting models/expressions.

- Perpetually, GPSR searched models are parsimonious (i.e. lesser complexity) than the analogous MLP and SVR models. Therefore, these models are easier to use, grasp, and deploy in a practical setting.
- The automatic search and optimization of the form and parameters of the data-fitting function performed by GPSR obviates the trial and error approach allied with the conventional linear or nonlinear regression analysis.

The GPs applications are wide worthy found in today's real-world scenario:

- GP have been used for selection of predictor variables and modeling in process identification
- It works better in coal/ coal gas applications where most the data is observed as nonlinear.
- GP applications are found to best suitable in thermodynamics such as vapor-liquid equilibrium (VLE).
- The quantitative structure property relationship (QSPR) is powerful analytical method for breaking down a molecule into a series of numerical values describing its relevant chemical and physical properties (e.g. charge density, hydrophobic surface area, etc.). While modeling the QSPR, GP have proved best results as it models, searches the optimized space and gives the best result.
- Apart from these GP have deep applications in the field of - Nanotechnology, Chemical Kinetics, Control Engineering, Bioinformatics, Gene Expression Profiling Analysis, Financial Mathematics, Economics, Deep Space Network, Civil and Mechanical Engineering, Marketing Mix Analysis, Multimodal Optimization, Multidimensional Systems, Wireless sensors, Quality control, etc.

A brief representative survey of GP-applications in chemical engineering/ technology is provided in **Table 1.3** below.

Table 1.3: A representative survey of GP-based modeling and classification applications

CI Formalism	**Application**	**Reference**
GP	Softsensor Model for Styrene Polymerization Process and its Application in Model based Control	Ghugare et al (2016)
	Prediction of HHV of coals of different ranks and from diverse geographies	Ghugare and Tambe (2017)
	Optimization of Glucose to Gluconic Acid Fermentation	Cheema et al (2002)
	Non-linear principal components analysis using genetic programming	Hiden et al(1998)
	Fault Detection	Zhang et al (2005)
	Robust soft sensors based on integration of genetic programming, analytical neural networks, and support vector machines	Kordon et al. (2002)
	Selecting Predictor Variables and Modeling in Process Identification	Verma et al (2016)
	Genetic programming-based models for prediction of vapor-liquid equilibrium	Patil-Shinde et al (2018)

1.3.4 Evolutionary and Genetic Algorithms

Evolutionary algorithms: Evolutionary algorithms are based upon the theory of biological evolution, namely that proposed by Charles Darvin in 1859. It is a branch of artificial intelligence termed "evolutionary computation." EAs have found numerous applications for solving complex nonlinear optimization problems. Given an objective function, the said optimization typically involves finding optimal values of the decision variables such that the objective function is maximized/minimized. There exist mainly three types of EAs; these are:

- Genetic algorithms
- Evolutionary programming
- Evolutionary strategies

Although their applications differ, the underlying implementation are similar. To begin, an EA generates a random population of candidate solutions to the given optimization problem. This is unlike the classical gradient based optimization methods which works on a single solution towards finding an optimal one. Next, the population is subjected to a selection process in which those members of the population which are in inferior in fitness are eliminated, while the fitter individuals are permitted to not only survive to the next generation but also produce offspring candidate solutions using an operation termed "crossover." This is akin to the "survival of the fittest" principle of Darwinian evolution theory. The fitness of a candidate solution essentially represents how well the solution fares at the optimization task, that is, objective function maximization/minimization. Typically, fitness of a candidate solution is computed using a fitness function. The offsprings candidates may be subjected to mutation which completes one cycle of evolutionary process. The iteration continues by subjecting the offspring to the fitness evaluation, crossover and mutation until a best solution is searched.

Two major benefits of the evolutionary algorithms are (a) they are well-suited for solving nonlinear optimization problems with multiple minima, and (b) invariably they provide a solution which corresponds to the global minimum. In the present thesis, genetic algorithms have been employed for obtaining optimal solutions for a problem involving Photocatalytic Degradation of Pharmaceutical Pollutants. A detailed description of GAs is provided below.

GA (Goldberg, 1989) are evolutionary algorithms based on the principles of "Darwinian evolution" of biologically evolving species and propagation of genetic characteristics from one generation to the next. Their application area is stochastic nonlinear optimization that has a number of advantages over the traditional gradient-based optimization methods. Specifically, GA can be used for maximization/minimization of an objective function that is not smooth, differentiable and continuous. Another important characteristic of GA is that they need measurements of objective function only, and not the measurements (or direct calculation) of the gradient (or higher order derivatives) of the objective function.

The applications of GA are Linguistic analysis, Protein folding, vehicle routing problem, software engineering, Filtering and signal processing, Mobile communications infrastructure optimization, etc. Most of the GAs applications are similar to GP; the difference is that GAs are bit string operations while GPs are tree structures.

A representative survey of the CI–based modeling and classification applications in chemical and biochemical engineering is tabulated below.

Table 1.4: A representative survey of GA-based modeling and classification applications

CI Formalism	**Application**	**Reference**
GA	Function Optimization	DeJong (1975); Petridis et al (1998)
	System Identification	Das and Goldberg (1988)
	Chemical flow-shop sequencing	Cartwright and Long (1993)
	Solving non-convex trim loss problem	Ostermark (1999)

	Biodegradable iron chelates for H_2S abatement	Hamid et al (2014)
	Optimization Techniques for Improving the Performance of Information Retrieval System	Badhe et al (2014)

As can be seen, each of the above stated CI-based formalism possesses some unique features making them useful in certain specific modeling tasks in chemical and biochemical systems presented in this thesis.

1.4 BRIEF SUMMARY OF THE THESIS

The thesis is organized in seven chapters. A brief overview of each chapter is provided below.

Chapter 1: Abstract of the thesis/Introduction

This chapter provides a brief outline of the contents of the thesis along with its rationale.

Chapter 2: Computational Intelligence based Modeling and Optimization methods

This chapter provides a detailed account of the conventional methods such as phenomenological, empirical, etc of modeling chemical and biochemical processes and their advantages and drawbacks. The chapter also gives the rationale for adopting computational intelligence-based methods for modeling and optimization of chemical and biochemical processes. Next, the chapter presents the state-of-art computational intelligence based and other traditional methodologies employed in the thesis. The methodologies are classified according to their areas of application:

1. *Modeling Methodologies*: Multilayer perceptron (MLP) neural networks, genetic programming (GP), and support vector regression (SVR).
2. *Optimization Methodologies*: Genetic algorithm (GA).
3. *Dimensionality Reduction Method*: Principal Component Analysis (PCA).
4. *Statistical method for model discrimination*: Steiger's z- test.

Chapter 3: Prediction of Rate Constant of Photocatalytic Degradation of Pharmaceutical Pollutants by machine learning based formalisms

This chapter presents the development of artificial neural networks (ANN) and genetic programming (GP) based models for predicting the photo catalytic degradation (PCD) rate constants from experimental and molecular structural inputs. The models, are developed for three pharmaceutical pollutants, namely, *Ciprofloxacin*, *Naproxen,* and *Paracetamol*. To predict the rate constant of the PCD of pharmaceuticals the stated models use following four experimental conditions: time of exposure of pharmaceutical pollutant to the solar radiation, pH of wastewater, concentration degradation during the measured time, and the ratio of degraded concentration to the initial concentration of the pollutant.

The MLP- and GP-based models predicting the rate constant of the PCD, proposed in this study, exhibit an excellent prediction accuracy [*correlation coefficient* (*CC*) > 0.9] and generalization performance. The machine learning based modeling strategy introduced here can be fruitfully used in the development of models for a variety of other pharmaceutical degradation reactions.

In addition to the modeling the chapter also reports a few results of optimization that has been performed with the objective of obtaining maximum possible degradation of pollutants under milder conditions.

Chapter 4: Prediction of Cetane number of biofuels using Artificial Intelligence based models

It is well-known that the fuel properties of biodiesel, namely, *cetane number* (CN), kinematic viscosity (KV), density (D) and higher heating value (HHV), play a significant role in its utilization for the combustion process. Accordingly, this chapter presents the development of exclusively data driven models predicting the Cetane Number (CN) of biofuels. The models are constructed using two Artificial Intelligence (AI)-based formalisms, namely, GP and MLP and employ fatty acid methyl ester (FAME) composition and fuel properties as inputs (predictors). The carbon chain length and degree of unsaturation are two significant factors of FAME that affect the property of CN of biodiesel (Tong et al., 2011) and thus contribute to the ignition quality of the biodiesel (Sivaramakrishnan and Ravikumar, 2012).

Based on the relevant data, two types of CN prediction models developed in this study using following inputs:

- Type I models: FAME composition—Capric (C10:0), Lauric (C12:0), Myristic (C14:0), Palmitic (C16:0), Palmitoleic (C16:1), Stearic (C18:0), Oleic (C18:1), Linoleic (C18:2), Linolenic (C18:3), Arachidic (C20:0), Paullinic (C20:1) and Erucic (C22:1).
- Type II models: Properties of biofuels, namely density (kg/l), flash point (°C), higher heating value (MJ/kg) and kinematic viscosity (mm^2/s).

For developing the two types of AI-based models, a dataset consisting of FAME composition and fuel properties of 171 and 67 biodiesel samples, respectively, and the corresponding CNs was compiled from various research articles (Ahmad et al., 2014; Martı´nez et al., 2014; Miraboutalebi et al., 2016). Principal component analysis (PCA) was performed on the twelve-dimensional input space of the FAME composition-based GP and MLP-based models with a view to reduce the input dimensionality of the models. In the case of fuel property-based models the input dimensionality was not high and therefore PCA was not performed on the input space of the corresponding models.

The CN prediction accuracy and generalization performance of the AI-based models indicate the following.

(a) FAME composition-based GP and MLP models exhibit an excellent prediction accuracy [correlation coefficient (CC) > 0.9 and root mean squared error (RMSE) < 5] and generalization performance.

(b) Fuel property-based models exhibit a reasonably good prediction accuracy and generalization performance [CC > 0.8 and RMSE < 4] for both training and test set data.

The notable characteristics of this study is that (i) to our knowledge as gained from the current literature, biodiesel property based nonlinear CN prediction models have been developed for the first time, and (ii) the novel GP method has been employed for the first time for developing CN prediction models. The AI-based models introduced here due to their excellent prediction and generalization performance have the potential to replace the existing biofuel Cetane Number prediction models in practical applications.

Chapter 5: Prediction of Coals Ash Fusion Temperatures using computational intelligence-based models

The ash fusion temperatures (AFTs) are important characteristics of coal ashes and used routinely in the operations of coal-based combustion and gasification processes to avoid/lessen occurrences of undesired slagging and fouling phenomena and their deleterious effects such as pressure drop, channel burning, corroding of furnace components, and unstable operation. Various models (predominantly linear) have been developed for predicting the corresponding *AFT* values from the chemical composition of coal ashes. However, estimation of AFT from chemical composition is a complex and relationship between the numbers of interacting factors is precisely unknown, thus, the stated approach becomes tedious, costly and time-consuming. Also, the stated AFT prediction models are based on the data from coals belonging to mostly a single geographical region although coal properties widely based upon their geographical origin. A scrutiny of an extensive coal ash chemical composition database for each ash phase namely, *Initial Deformation Temperature* (*IDT*), *Softening Temperature* (ST), *Hemispherical Temperature* (HT) and *Flow Temperature* (FT) suggests that the relationships between the AFT and weight percentages of some of the chemical and mineral constituents could be nonlinear. For capturing these nonlinearities and thereby developing models possessing better *AFT* prediction accuracies, this work uses three computational intelligence (CI) based exclusively data-driven formalisms, namely, genetic programming (GP), multi-layer perceptron (MLP) neural network, and support vector regression (SVR) for developing models for prediction of four AFTs. Among three CI-based methods, the GP possesses several novel and attractive characteristics; yet it remains much less utilized data-driven modeling technique compared to ANNs and SVR. A notable feature of this study is that a large number of chemical and mineral constituents data and the corresponding AFT values pertaining to the global as well as Indian coal ashes have been utilized in the model development. These characteristics have imparted a wider applicability to the CI-based models. The inputs to the models are the major oxide constituents of the chemical and mineral composition of the coal ash namely, SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO, TiO_2, and K_2O+Na_2O. The seven-dimensional input space of the AFT models was reduced by using the principal component analysis and the pruned dataset was used in the modeling. All the four best fitting GP-based models developed in this work for the prediction of four ash phase temperatures possess nonlinear forms. This result clearly shows that the relationships between the seven oxides in the coal ashes and the corresponding AFTs are nonlinear. A comparison of the prediction accuracy and generalization performance of the three CI-based models shows that (a) the performance in predicting *IDT* and *ST magnitudes* of GP, MLP and SVR models is comparable, and (b) the performance of the MLP based models, in the case *HT* and *FT predictions,* is better than GP and SVR models. Therefore the CI-based models developed in this work have a potential due to their wider applicability, high AFT prediction accuracy and generalization performance for predicting the AFT values of coal ashes from different geographies in the world.

Chapter 6: Prediction of *Initial Deformation Temperatures* of Indian coals using computational intelligence-based models

The property that governs the deposition of ash in various coal-utilizing process equipment is termed *ash fusion temperature* (AFT). The phenomena responsible for the said ash deposition are termed *slagging* and *fouling*. One of the four commonly used AFTs in practice is termed *Initial deformation temperature* (*IDT*). It refers to the temperature at which ash just begins to flow. In India, the currently mined coals have a high ash content and these coals are predominantly used in the power generation industry. It is therefore important to develop models predicting IDT magnitudes specifically for Indian coals with intrinsically high ash content. Such models are expected to be useful in operating coal-based thermal power stations in India. With this objective, computational intelligence (CI) based IDT prediction models were developed in this study wherein the contents of the seven principal oxides found in coal ashes (i.e. SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO, TiO_2, Na_2O+K_2O), were used as predictors. The experimental data on Indian coals were sourced from *Central Institute of Mining and Fuel Research* (CIMFR), Dhanbad, India. These data correspond to 91 coal ash samples and contain information of the above stated seven oxides and the corresponding IDTs. Three CI-based modeling formalisms, namely MLPNN, SVR and GP were employed to develop IDT prediction models. The IDT prediction accuracy and generalization performance of these models was evaluated in terms of *coefficient of correlation* (*CC*) and *root mean square error* (*RMSE*) values pertaining to the experimental and model predicted magnitudes of the IDT of Indian coal ashes. Among the three models constructed using the GP, MLPNN and SVR methods, the IDT model prediction and generalization performance of the SVR-based model was found to be superior to the corresponding GP and MLPNN-based models.

Chapter 7: Conclusions

This chapter summarizes the principal findings of the studies presented in the thesis and provides suggestions for the future work.

1.5 REFERENCES

Ahmad J., Yusup S., Bokhari A., Kamil R. (2014). Study of fuel properties of rubber seed oil-based biodiesel. *Energy Conversion and Management*, **78**, pp. 266-275.

Arabloo M., Bahadori A., Ghiasi M., Lee M.A., Zendehboudi S. (2015). A novel modeling approach to optimize oxygen–steam ratios in coal gasification process, *Fuel*, **153**, pp.1-5.

Badhe Y.P., Bhat R., Boate P. (2014). Optimization techniques for improving the performance of information retrieval system, *International Journal of Research in Advent Technology*, **2(2)**, pp. 263- 267. E-ISSN: 2321-9637.

Blaesi J. and Jensen B. (1992). Can neural network compete with process calculations? , *INTECH*, pp. 34-37.

Bos M., Bos A., Van der Linden W. (1993). Data processing by neural networks in quantitative chemical analysis, *Analyst*, **118**, pp. 323-328.

Brooks R., Iyengar S., Chen J. (1996). Automatic correlation and calibration of noisy sensor readings using elite genetic algorithms, *Artificial Intelligence*, **84**, pp.339-354.

Cartwright H., Long R. (1993). Simultaneous optimization of chemical flow-shop sequencing and topology using genetic algorithms, *Ind. Eng. Chem. Res.*, ***32*** **(11)**, pp 2706–2713.

Cheema J., Sankpal N., Tambe S.S., Kulkarni B.D. (2002). Genetic programming assisted stochastic optimization strategies for optimization of glucose to gluconic acid fermentation, *Biotechnology Progress*, **18**, pp. 356−1365.

Chitra S. (1993). Use neural networks for problem solving, *Chemical Engineering Programming*, pp. 44-52.

Constantinides A. (1987). *Applied Numerical Methods with Personal Computers*, McGraw Hill Chemical Engineering Series, pp. 528-534.

Czop P., Kost G., Sławik D., Wszołek G. (2011). Formulation and identification of First-Principle Data-Driven models, *Journal of Achievements in Materials and Manufacturing Engineering*, **44(2)**, pp. 179-186.

Darwin C. (1859). *On the Origin of Species by Means of Natural Selection; or, the Preservation of flavored Races in the Struggle for Life*, John Murray, London.

Das R., Goldberg D. (1988). Discrete-time parameter estimation with genetic algorithms. *19th Annual Pittsburgh Conference on Modeling and Simulation*, **19**, pp. 2391-2395.

DeJong T. (1975). A comparison of three diversity indices based on their components of richness and evenness, *OIKOS*, **26**,pp. 222-227.

Desai K., Badhe Y.P., Tambe S. S., Kulkarni B.D. (2006). Soft-sensor development for fed-batch bioreactors using support vector regression, *Biochemical Engineering Journal*, **27**, 225–239.

Gandhi A., Joshi J., Kulkarni A., Jayaraman V., Kulkarni B.D. (2008). SVR-based prediction of point gas hold-up for bubble column reactor through recurrence quantification analysis of LDA time-series, *International Journal of Multiphase Flow*, **34(12)**, pp. 1099-1107.

Ghugare S.B., Tambe S.S. (2016). Development of genetic programming based softsens or model for styrene polymerization process and its application in model-based control, *Indian Control Conference (ICC)*, IIT Hyderabad, India.

Ghugare S.B., Tambe S.S. (2017). Genetic programming based high performing correlations for prediction of higher heating value of coals of different ranks and from diverse geographies, *Journal of the Energy Institute*, **90(3)**, pp.476-484.

Girosi F., Poggio T (1990). Networks and the best approximation property, *Biological Cybernetics*, **63(3)**, pp.169–176.

Goldberg D. (1989). *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Hamid A., Deshpande A.P., Badhe Y.P., Barve P., Tambe S.S., Kulkarni B.D. (2014). Biodegradable iron chelates for H_2S abatement: Modeling and optimization using artificial intelligence strategies, *Chemical Engineering Research and Design,* **92**, pp.1119–1132.

Hassan M. R., Nath B., Kirley M. (2007).A fusion model of HMM, ANN and GA for stock market forecasting, *Expert Systems with Applications*, **33(1)**,pp. 171-180.

Hiden H., Willis M., Tham M., Turner P., Montague G. (1997). Non-linear principal components analysis using genetic programming, *IEE*, Conf. Publication No.446, pp.302-307.

IBM SPSS Neural Networks 20 manual, IBM Chicago, 2011.

Ivanciuc O., Lipkowitz K., Cundari T. (2007). *Reviews in computational chemistry-applications of support vector machines in chemistry*, Chapter 6, Wiley VCH, **23**, pp.291-378.

Khandelwal M., Armaghani D., Majid M., Yagiz S. (2017). Function development for appraising brittleness of intact rocks using genetic programming and nonlinear multiple regression models, *Engineering with Computers,* **33**, 13-21.

Kordon A., Smits G., Jordaan E., Rightor E.D. (2002). Robust soft sensors based on integration of genetic programming, analytical neural networks, and support vector machines, *Proceedings of 2002 Congress on Evolutionary Computation*, **2**, pp. 896-901.

Koza J. (1990). Genetically breeding populations of computer programs to solve problems in artificial intelligence, *Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence*; 819–27.

Krishnan V.V. (2017). Recent developments in metal-supported solid oxide fuel cells, *WIREs Energy Environment*, **6**, pp. 1-35. DOI: 10.1002/wene.246

Kuan C. M., White H. (1994). Artificial neural networks: an econometric perspective, *Econometric Reviews*,**13(1)**, pp. 1-91.

Kulkarni S., Chaudhary A., Nandi S., Tambe S.S., Kulkarni B.D. (2004). Modeling and monitoring of batch processes using principal component analysis (PCA) assisted generalized regression neural networks (GRNN), *Biochemical Engineering Journal*, **18**, pp.193–210

Le G., Arrault A., Colliandre L., Bourg S., Vayer P., Morin A. (2012). Mining collections of compounds with screening assistant 2. *Journal of Cheminformatics,* **4**, pp.20.

Li Y. and Ma W. (2010). Applications of Artificial Neural Networks in Financial Economics: A Survey, *International Symposium on Computational Intelligence and Design*, pp. 211-214. DOI: 10.1109/ISCID.2010.70

Martı´nez G., Sanchez N., Encinar J., Gonza´lez J. (2014). Fuel properties of biodiesel from vegetable oils and oil mixtures- Influence of methyl esters distribution, *Biomass Bioenergy*, **63**, pp.22-32.

Meng Q., Xiaoping M., Zhou Y.(2014). Forecasting of coal seam gas content by using support vector regression based on particle swarm optimization, *Journal of Natural Gas Science and Engineering*, **21**, pp. 71-78.

Mielke P. W. (1984).34 Meteorological applications of permutation techniques based on distance functions, *Handbook of Statistics*, **4**, pp. 813-830.

Miraboutalebi S., Kazemi P., Bahrami P. (2016). Fatty acid methyl ester (FAME) composition used for estimation of biodiesel cetane number employing random forest and artificial neural networks: A new approach. *Fuel*, **166**, pp. 143-151.

Nandi S., Badhe Y.P., Lonari J., Sridevi U., Rao B., Tambe S.S., Kulkarni B.D. (2004). Hybrid process modeling and optimization strategies integrating neural networks/support vector regression and genetic algorithms: study of benzene isopropylation on H-beta catalyst, *Chemical Engineering Journal*, **97**, pp.115–129.

Nandi S., Ghosh S., Tambe S.S., Kulkarni B.D. (2001). Artificial neural-network-assisted stochastic process optimization strategies, *AICHE journal*,**47(1),** pp.126-141.

Nandi S., Mukherjee P., Tambe S., Rajivkumar, Kulkarni B.D. (2002). Reaction Modeling and Optimization Using Neural Networks and Genetic Algorithms: Case Study Involving TS-1-Catalyzed Hydroxylation of Benzene, *Ind. Eng. Chem. Res.*,**41**, pp. 2159-2169.

Niebling G. (1994). Identification of gases with classical pattern-recognition methods and artificial neural networks, *Sensors and Actuators*, **18-19**, pp. 259-263.

Ostermark R. (1999). Solving a nonlinear non-convex trim loss problem with a genetic hybrid algorithm, *Computers & Operations Research*, **26(6)**, pp.623-635.

Patel S., Jeevankumar B., Badhe Y.P., Sharma B., Saha S., Biswas S., Chaudhury A., Tambe S.S., Kulkarni B.D. (2007). Estimation of gross calorific value of coals using artificial neural networks, *Fuel*, **86**, 334–344.

Patil-Shinde V., Kulkarni T., Kulkarni R., Chavan P., Sharma T., Sharma B., Tambe S.S., Kulkarni B.D. (2014). Artificial intelligence-based modeling of high ash coal gasification in a pilot plant scale fluidized bed gasifier, *Ind. Eng. Chem. Res*., **53 (49)**, pp.18678–18689. DOI: http://dx.doi.org/ 10.1021/ie500593j

Patil-Shinde V., Saha S., Sharma B., Tambe S.S., Kulkarni B.D. (2016). High ash char gasification in thermo-gravimetric analyzer and prediction of gasification performance parameters using computational intelligence formalisms, *Chemical Engineering Communications* , **203(8)**, pp.1029-1044. DOI: https://doi.org/ 10.1080/00986445.2015.1135795

Patil-Shinde V., Tambe S.S. (2018). Genetic programming based models for prediction of vapor-liquid equilibrium, *Calphad*, **60**, pp. 68-80.

Petridis V. (1998). A hybrid neural-genetic multi-model parameter estimation algorithm, *IEEE TRANSACTIONS ON NEURAL NETWORKS*, **9(5)**, pp.862-876.

Saltelli A., Tarantola S., Chan K. (1999). A quantitative model-independent method for global sensitivity analysis of model output. *Technimetrics*, **41,** pp. 39-56.

Sharma A., Saikia A., Khare P., Dutta D., Baruah B. (2014). The chemical composition of tertiary Indian coal ash and its combustion behavior – a statistical approach: Part 2, *Journal of Earth System Science,* **123(6)**, pp.1439-1449.

Sivaramakrishnan K., Ravikumar P. (2012). Determination of cetane number of biodiesel and its influence on physical properties. *Journal of. Engineering and Applied Sciences*, **7(2)**, pp. 205-211.

Syu M., Tsao G., Austin G., Celotto G., D'Amore T. (1994). Neural network modeling for predicting brewing fermentations, *Journal of the American Society of Brewing Chemist,***52**, pp. 15-19.

Tambe S.S., Kulkarni B.D., Deshpande P.B. (1996). *Elements of artificial neural networks with selected applications in chemical engineering, and chemical & biological sciences*, Simulation & Advanced Controls, Louisville KY.

Tong D., Hu C., Jiang K., Li Y. (2011). Cetane number predictions of biodiesel from the composition of the Fatty Acid Methyl Esters. *J. Am. Oil Chem. Soc.*, **88,** 415-423.

Valero S., Argente E., Botti V., Serra J., Serna P., Moliner M., Corma A. (2009). DoE framework for catalyst development based on soft computing techniques, *Computers & Chemical Engineering*, **33(1)**, pp.225-238.

Vapnik V. (1995). *The nature of statisitcal learning theory*, Springer Verlag, New York.

Venkatesan C., Raskar S., Tambe S.S., Kulkarni B.D., Keshavamurty R. (1997). Prediction of all India summer monsoon rainfall using error-back-propagation neural networks, *Meteorology and Atmospheric Physics*, **62**, pp. 225-240.

Verma D., Goel P., Patil-Shinde V., Tambe S.S. (2016). Use genetic programming for selecting predictor variables and modeling in process identification, *Indian Control Conference (ICC)*, IIT Hyderabad, India, pp.230-237.

Vora N., Tambe S.S., Kulkarni B.D. (1997). Counter-propagation neural networks for fault detection and diagnosis, *Computers and Chemical Engineering*, **21(2)**, pp.177-185.

Vyas R., Goel P., Tambe S.S. (2015). *Genetic programming applications in chemical sciences and engineering*, in: A.H. Gandomi, H. Amir, Alavi, C. Ryan, (Eds.), Handbook of Genetic Programming Applications, Springer International Publishing, Switzerland, pp. 99–140. DOI: http://dx.doi.org/10.1007/978-3-319-20883-1.

Zhang L., Jack L., Nandi A. (2005). Fault detection using genetic programming, *Mechanical Systems and Signal Processing*, **19(2)**, pp. 271-289.

CHAPTER 2

COMPUTATIONAL INTELLIGENCE BASED MODELING AND OPTIMIZATION METHODS

Abstract

Modern chemical processes are complex in character with numerous reactors, unit processing equipment, raw materials, controllers, etc. Also, it is absolutely essential to operate them efficiently, cost-effectively, causing minimum or no damage to the environment, and at the same time manufacturing products that are safe to use and of high quality. Process modeling and optimization plays an important role in achieving these objectives. There are a number of different strategies for modeling and optimization of chemical reactions, reactors and processes. These include phenomenological, stochastic, Monte Carlo, cellular automata, statistical, empirical and computational intelligence-based ones. Each of these methods possesses certain advantages and drawbacks. Since they overcome some serious deficiencies of the commonly used phenomenological modeling and optimization methods, in this thesis, state-of-the-art exclusively data-driven computational intelligence (CI) based methods have been employed for modeling and optimization of chemical and biochemical systems. The modeling methods used are *artificial neural networks* (ANNs), *support vector regression* (SVR) and *genetic programming* (GP), while genetic algorithms (GA) have been used for process optimization. This chapter provides a detailed account of these methods, with a view of presenting their origin, functional aspects, implementation procedure and applications.

The methodologies described here are classified according to their areas of application:

1. *Modeling Methodologies*:
    - Multiple input – single output (MISO) and Multiple input – multiple output (MIMO) data-driven modeling: Multilayer perceptron neural networks (MLPNN), genetic programming (GP), and support vector regression (SVR).
2. *Optimization Methodology: Genetic algorithm (GA).*

In addition to the CI methods, the studies presented in the thesis employ a method known as *Principal Component Analysis* (PCA) for reducing the dimensionality

of the data used as model inputs. Accordingly, a short description of PCA is also provided in this chapter. Also presented in this chapter are the statistical measures namely *coefficient of correlation* (CC) and *root mean square error* (RMSE) and a test termed Steiger's z-test that are used to measure the performance of the individual models and to discriminate between multiple models and select the best one.

2.1 PROCESS MODELING

The knowledge of the “cause-and-effect” relationships is vital in the solution of problems in all fields of endeavor (Tambe et al., 1996). The first phase of modeling of any scientific/engineering/technology system comprises collection of the information and systematic knowledge of the system (Naddeo et al, 2008). It entails the *a priori* knowledge regarding a given phenomenon that comes from the analysis of all possible linkages with other phenomena and physical laws, preceding the modeling. In process modeling, the *a priori* knowledge about the system is of significant importance although its availability is always limited by the complexity of the physical/chemical system (Buchacz and Zolkiewski, 2007). Even if the physical and/or chemical principles underlying a system are known, it is usually difficult to formulate the specific relationships and obtain particular values of the parameters such as the kinetic rate constants, heat and mass transfer coefficients and relevant thermodynamic quantities. The extent of the a priori knowledge available for modeling decide the following: (i) the final type of the model that is possible to be developed, (ii) model accuracy, (iii) the type of algorithm to be used in model construction, (iv) the complexity of the model and, (v) the cost of model construction (Tambe et al., 1996).

On the basis of their temporal behavior, two types of chemical process models are commonly developed, namely, s*teady-state* and *dynamic* models. A steady-state model describes the time-invariant behavior of a chemical process. On the other hand, dynamic models describe the time-dependent behavior of the process and in general are more complex to solve than the steady-state models. These models are used in the control system design for analyzing the process response for monitoring the disturbances, set-point changes, etc. as also for tuning the controllers. Depending upon the quantum of the a priori knowledge available about the system, either a first principles or an empirical is constructed.

Commonly, process models are constructed using *phenomenological* and *data-driven* (*empirical*) approaches.

2.1.1 Phenomenological Modeling Approach

The phenomenological modeling approach, also termed as "first-principles" or “mechanistic” modeling approach, describes the process behavior in terms of the appropriate mass, momentum and energy balance equations collectively with the pertinent chemical engineering principles. Such models contain a number of system parameters and quantitatively articulate the *cause-and-effect* relationships in the form of algebraic or differential or algebraic-differential equations. For example, a phenomenological model for a non-isothermal continuous stirred tank reactor (CSTR) (Carberry, 2001; Dutta, 2017) uses information such as the reactant inlet concentration(s), kinetic rate constants, flow rates, etc. to compute the magnitudes of the outlet temperature and concentration. In the first principles modeling approach, a mathematical model describing the physico-chemical phenomena underlying in the process is first formulated and the model fitting is conducted by estimating the values of the unknown model parameters by utilizing actual process input-output data. For data- fitting purposes, the linear / nonlinear regression techniques based on the least squares’ minimization formalisms are commonly employed.

The distinct advantages of the phenomenological process models are: (i) they account for the underlying physico-chemical phenomena as closely as possible, and (ii) can be used in extrapolation (scale-up) i.e., a phenomenological model can be utilized even outside the range spanned by the experimental process input-output data used in the model fitting. A significant drawback of this type of modeling is that in most real-life situations the complete understanding of the physico-chemical phenomena is not available thus making the model development a tedious, costly and difficult task. Moreover, most chemical processes behave nonlinearly and multiple nonlinear dependencies exist between system/process variables. Such models cannot be solved using analytical techniques and, therefore, require rigorous computationally intensive numerical methods for their solutions.

2.1.2 Empirical Modeling Approach

The empirical approach to process modeling involves construction of a single or multi-variable linear/nonlinear regression models. Here, an approximate or a random guess is made regarding the structure of the data-fitting function and its parameters are ascertained using a suitable linear/nonlinear parameter estimation method and the process data. Empirical modeling approach is relatively less tedious compared to the first principles modeling although the guesswork for choosing an appropriate fitting function and the consequent parameter estimation does involve a trial and error procedure. The critical task of specifying the structure (form) of the fitting function requires multiple trials particularly when the relationships between model inputs and the outputs are nonlinear. The reason for numerous trials is that a large number of fitting functions compete for the data-fitting. Additional constraint of an empirical model is its validity being limited over the ranges of the data used in its construction. That is, in general these models are poor at extrapolation and therefore cannot be used in scale-up. Also, empirical modeling requires a statistically well distributed data for them to possess good prediction accuracy and generalization performance. A model possessing good generalization ability is capable of predicting outputs accurately for new sets of inputs that were not part of the dataset used in the model construction. The most significant advantage of the empirical modeling approach is that a detailed knowledge of the physico-chemical phenomena underlying the process behavior is not required in the model construction thus saving the enormous efforts required in obtaining the related information.

2.1.3 Alternative modeling approaches

The above-stated limitations of the phenomenological and empirical modeling approaches necessitated a paradigm shift in the approach towards modeling of scientific and engineering/technology systems. Specifically, the new modeling approach should possess the following attributes so as to prove its suitability when phenomenological and empirical modeling are found to be infeasible.

- A method that doesn't call for an in-depth knowledge of the physicochemical phenomena underlying a process; that is, it should be essentially process data-driven.
- The modeling formalism should not require pre-specifying the form (structure) of the data-fitting function. Ideally, it should automatically choose the form of the data-fitting function without making any assumptions about the form.
- The method should be capable of developing models possessing good prediction accuracy, and equally importantly an excellent generalization capability.

Post-1980s, a number of *Computational Intelligence* (CI) based approaches possessing above-stated attributes were proposed and these have offered an attractive alternative not only for modeling of chemical systems/processes but also for their optimization. A wide variety of engineering problems exist for which phenomenological and conventional empirical modeling poses difficulties. For such problems, the CI-based approaches can bring efficient and speedy solutions. The CI-based modeling approaches possess following advantages.

(i) The models are constructed exclusively from the process data consisting of the dependent (also termed *model outputs/response variables*) and independent variables (termed *model inputs/predictor/causal* variables) without invoking the first principles underlying the system/process.

(ii) Being robust they can handle the system/process uncertainties (noise) efficiently.

(iii) They are particularly suitable for modeling complex nonlinear systems/processes.

(iv) These methods do not require explicit specification of the form of the data-fitting function.

Owing to their above-stated several attractive characteristics, in the present thesis steady-state models have been developed of chemical and biochemical systems using computational intelligence methods.

2.2 PROCESS OPTIMIZATION

All chemical processes can be improved to various extents and therefore scientists, engineers and technologies strive to optimize them. In chemical engineering/technology optimization typically finds applications in process design and operation, product and model development, real-time (online) optimization, etc. The task of process optimization refers to optimizing process operating variables and parameters in a manner such that while satisfying the imposed constraints, specific measures of process performance are significantly improved. In actual process optimization practice, an objective function representing the performance measure is either maximized or minimized. The essential attributes of chemical process optimization are as follows (Biegler, 2010).

(a) A typical optimization tasks involves maximization of, for instance, reaction conversion, yield, selectivity's of the desired product and profit, or minimization of the operating cost, energy requirement, selectivity of the undesired product, etc. For example, optimization of a typical distillation column operation comprises determination of the optimum column-operating and feed conditions in order to acquire maximum product yield from minimum amount of reactants in feed, minimum reboiler energy and reflux.

(b) A predictive model describing the behavior of the process and possessing good prediction accuracy and generalization performance is essential. This model, which represents the objective function can be phenomenological, empirical or black-box (exclusively data-driven one such as CI-based). Often, a solution to an optimization problem needs to satisfy a constraint.

(c) Depending upon the optimization task, the decision variables appearing in the objective function are optimized such that the function is maximized or minimized while satisfying the imposed constraints. The constraints essentially specify feasible regions of system operation. Often, process optimization provides multiple sets of operating variables, leading to an improved process performance.

2.2.1 Conventional Methods of Process Optimization

The two types of process optimization methods are namely *deterministic* and *stochastic*. In the case of unconstrained multi-variable optimization problems, the following conventional deterministic optimization methodologies are used (Edger et al., 2001).

- *Gradient based methods*: Newton's method, Cauchy's steepest descent method, Generalized Reduced Gradient (GRG), Nelder-Mead algorithm simplex method, sequential linear programming (SLP), sequential quadratic programming (SQP), and Levenberg-Marquardt (LM) method.

- *Direct search methods*: Simplex search, Pattern search, Conjugate direction.

Among these, the gradient-based methods are employed commonly. In their implementation, the solution to an optimization problem is represented in the form of a vector consisting of the values of decision variables at which the gradient of the objective function with respect to the decision variables becomes zero. Thus, gradient computation is an integral feature of such optimization methods. The gradient-based methods are iterative. For example, the Newton's method requires computing the first as well as the second order gradient (in the form of a matrix termed “Hessian” of second order partial derivatives) of the objective function with respect to the individual decision variable. A majority of the above-stated gradient based optimization methods, suffer from the following disadvantages.

- The objective function to be maximized/minimized needs to be continuous, smooth, and differentiable over the feasible region of the decision variables.
- For a constrained optimization problem, a feasible region must be of convex shape.
- Computation of the gradients (either first order and/or second order) at each step is computationally expensive.
- Higher probability of solutions getting get stuck in a local optimum, thus leading to a sub-optimal solution.
- Convergence to a solution depends on the chosen initial conditions/guess.
- Ineffective in treating discrete search space problems.

In a number of real-life optimization problems—especially those involving objective functions derived using exclusively data-driven modeling methods such as the computational intelligence-based ones—the objective functions are nonlinear, noisy, and non-smooth. In such cases, the conventional deterministic gradient based optimization methodologies are found to be inefficient and unsuitable, which necessitates exploration of alternate strategies for process optimization.

In the last two decades stochastic optimization algorithms have found ever growing applications in almost all scientific and engineering/technology branches as also finance and business. They posses several attractive attributes and thus have emerged as a smart alternative to the deterministic optimization methods. Many of these methods have now become “industry standard” approaches for solving challenging optimization problems (Spall, 2004).

A majority of the stochastic optimization methods are not gradient based. A noteworthy feature of the stochastic optimization technique is that emphasis is given on sampling the search space as widely as possible while trying to locate the promising regions for further search. This is achieved by searching the solution space randomly. Thus, implementation of stochastic optimization methods involves random (probabilistic) step(s). In contrast to the conventional deterministic optimization techniques, which operate on a single candidate solution, the stochastic methods function on a population of candidate solutions. The size of the population depends on the specific problem under consideration. These characteristics makes it possible for the stochastic techniques to search several areas of the solution space simultaneously. In their implementation, randomly generated initial population of solutions is constantly refined to find improved solutions.

A number of computational intelligence (CI) based stochastic optimization techniques have been proposed, such as genetic algorithms (GA) (Goldberg, 1989; Holland, 1975), memetic algorithms (Moscato, 1992; Rechenberg, 1989), particle swarm (Kennedy and Eberhart, 1995), differential evolution (Aluffi-Pentini et al., 1985) and recent artificial immune systems (AIS) (De Castro and Zuben, 2002; Jerne, 1974). The stochastic optimization techniques have following advantages (Nandi et al., 2001; Spall, 2004).

- They are well-suited for dealing with noisy objective functions or highly nonlinear, high dimensional systems for which classical deterministic methods of optimization are unsuitable.
- Can efficiently solve optimization problems involving non-differentiable, discontinuous and noisy objective functions as also those involving discrete search spaces.
- They need measurements of the objective function only, and not the measurements (or direct evaluation) of the gradient (or higher order derivatives) of the objective function.

- Invariably, for problems involving minimization, converge to the global minimum or the deepest local minimum on the objective function surface.

A significant drawback of the stochastic optimization methods is that since they work upon a population of candidate solutions simultaneously, their implementation is computationally expensive. This drawback can be easily overcome by employing parallel processing computers since a number of steps in their implementation are amenable to parallelization.

In the present thesis, a widely used stochastic optimization technique, namely, Genetic Algorithm (Goldberg, 1989; Holland, 1975) has been employed for the optimization of the photo-catalytic degradation of pharmaceutical pollutant process (See Chapter 3). A detailed description of GA is provided in Section 2.5.

2.3 CI-BASED MODELING METHODOLOGIES

Owing to their several attractive features, three CI-based modeling methodologies, namely Artificial *Neural Networks* (ANNs), *Genetic Programming* (GP), and *Support Vector Regression* (SVR) have been used extensively for reaction/process modelling in this thesis. In what follows, a detailed description of their origin, functional attributes and applications is presented.

2.3.1. Artificial Neural Networks (ANNs)

Artificial neural networks are information-processing formalisms based on the mechanisms followed by the highly interconnected network of the cellular structure of the human brain. The billions of cells forming this network are known as *neurons*. Figure (2.1) shows the basic structure of a biological neuron. Individually, a neuron performs simple signal processing; however, the neural network formed by billions of neurons possesses immense information processing capability that forms the basis of the human intelligence and other brain functions. This intelligence imparting ability of the biological neural network is copied in an ANN in an oversimplified manner. ANNs can be viewed as computational algorithms mimicking the behavior of a biological neural network for accomplishing various tasks such as classification, function approximation, nonlinear principal component analysis (NPCA), noise reduction, clustering, pattern recognition, and memory association.

The artificial neuron works in a manner analogue to a biological neuron. As shown in Figure (2.1), signals coming in through various inlet connectors, known as *dendrites*, are summed in the nucleus of the cell. The summed signal is further propagated through the *axon* connectors to the downstream neurons. The axons of a neuron are connected to hundreds of downstream neurons through a separating nerve cell gap called the *synapse*. A synapse acts as an activation function for the electrochemical signals that are transferred onwards. The information processing/transfer that occurs within the network essentially takes place via the electrochemical signals. The signal received by the connected downstream neurons is processed in a similar fashion by each linked neuron. These numerous neurons working in assembly of the network finally generate the processed signal at the output nodes of the network.

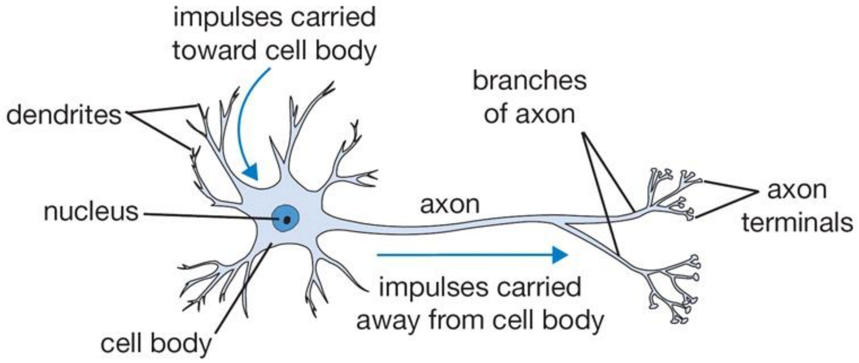


Figure 2.1: A biological neuron.

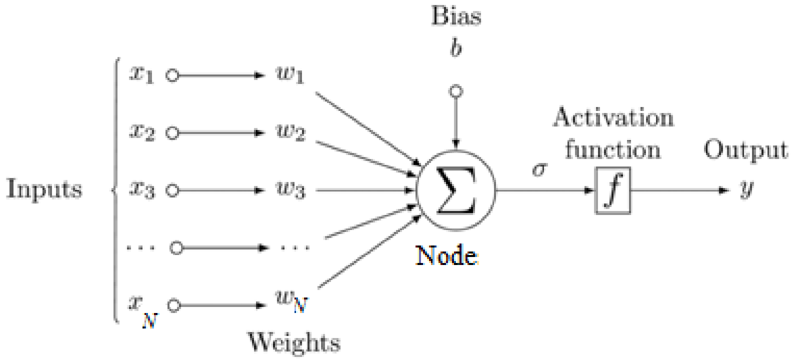


Figure 2.2: An artificial neuron

An artificial neuron (also termed processing element or node) and the information flow (via signals) associated with it is depicted in Figure 2.2, where $x_{1,} x_{2,} x_{3,}, \ldots., x_N$ are the 'N' inputs to the neuron and $w_{1,} w_{2,} w_{3,}, \ldots., w_N$ are the weights attached to the input signals. As shown in the figure, if an N-dimensional input vector, $X = [x_{1,} x_{2,} x_{3,}, \ldots., x_N]^T$, is presented to the neuron, then each of the input x_i is weighed by a weight w_i and is passed to the summing junction. In order to fire the neuron a unity input with a weight 'b', known as *bias* is also fed to the summing junction. Here, all the weighed inputs are summed according to Eqn. (2.1) and the resulting output is transformed by an appropriate transfer function (known as an activation/ threshold function) to produce the desired output 'y'.

$$y = \sum_{i=1}^{N} w_i \, x_i + b \tag{2.1}$$

It is the use of nonlinear transfer functions that imparts ANNs their ability of nonlinear classification/regression. The commonly used transfer functions are logistic sigmoid, hyperboloid tangent and linear and Gaussian functions.

Table 2.1: Well-known Artificial Neural Network architectures

ANN Architecture			
Feedforward		Feedback	
Supervised learning	Unsupervised learning	Supervised learning	Unsupervised learning
• Adaline/Madaline	• Learning Matrix (LM)	• Fuzzy Cognitive Map (FCM)	• Additive Grossberg (AG)
• Multilayer Perceptron (MLP)	• Fuzzy Associative Memory (FAM)	• Brain-state in-a-box (BSB)	• Adaptive Bidirectional Associative Memory (ABAM)
• Counter-propagation (CP)	• Drive- Reinforcement Learning (DR)	• Recurrent Neural Networks (RNL)	• Discrete Bidirectional Associative Memory (DBAM)
• Perceptron	• Kohonen Self-Organizing Map (SOFM)		• Continuous Hopfield (CH)
• Radial Basis Function (RBFN)	• Learning Vector Quantizer (LVQ)		• Discrete Hopfield (DH)

Commonly, ANNS are classified on the basis of how the information flow occurs in their structure. Accordingly, they are classified either as “feed-forward” or “feedback” ANNs. In the former category, the information flow occurs in a single (i.e. forward) direction from the input layer to the output layer whereas in the second category, inter-layer or intra-layer feedback loops are present. Table (2.1) lists various types of ANNs along with their information flow and learning modes (Tambe et al., 1996).

2.3.1.1 Multilayer perceptron (MLP) neural networks

An ANN termed *Multilayer Perceptron Neural Network* (MLPNN) is the most widely employed ANN for performing exclusively data-driven nonlinear function approximation. The unique information processing capability of MLP arises from its multilayer structure housing artificial neurons (processing nodes/ elements) that are linked using weighted synaptic connections. When a MLP possess a feed-forward arrangement meaning the information flow take place only in the forward direction. Generally, it contains three layers namely *input*, *hidden*, and *output* layers (see Figure 2.3). In some cases, MLPNN also houses multiple hidden layers in its structural design. Every node in its hidden layer processes incoming information using a nonlinear transfer function, for example the logistic sigmoid and hyperbolic tangent (tanh), to calculate its output. The much-desired nonlinear input-output mapping ability of a MLPNN is due to the said nonlinear processing performed by its hidden layer nodes. Given an example data, consisting of the inputs (independent/predictor/causal variables) and the corresponding outputs (dependent/response variables), an MLPNN is capable of learning complex nonlinear input-output relationships. This training (learning) is performed by employing an appropriate learning algorithm [e.g. *error-back-propagation* algorithm (Rumelhart et al., 1986) , the conjugate gradient algorithm (Moller,1993) and Levenberg-Marquardt method (Levenberg,1944; Marquardt, 1963)] that updates the weights on the connections linking the inter-layer nodes such that a measure of the error (termed "prediction error") between the MLPNN computed outputs and their desired (target) magnitudes is minimized. A detailed description of the MLPNNs is available at numerous books (e.g. Bishop, 1995; Tambe et al., 1996; Zurada, 1992). The reviews and research articles (see for example, Rumelhart et al., 1986 and Zhang et al., 1998). For the sake of brevity an in-depth description of MLP training is provided below.

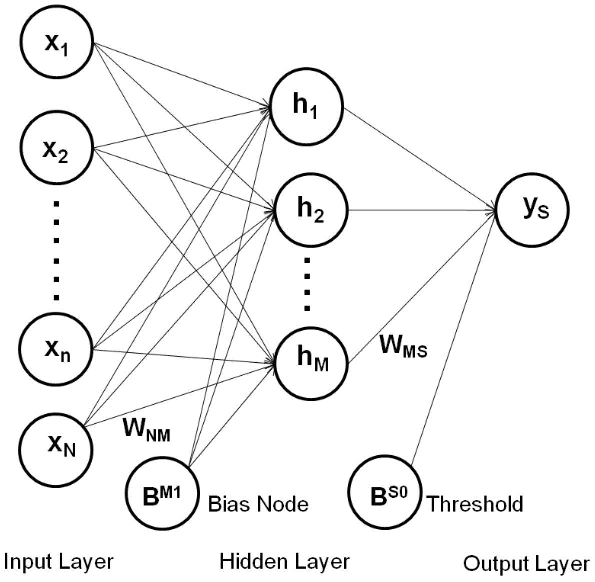


Figure 2.3: Schematic of a single hidden layer multiple input-single output (MISO) MLP network

An MLPNN works in the either of the following two modes:

1. *Training/learning Mode*: This step is implemented after readying the dataset known as the "example set" comprising the inputs and the corresponding outputs/targets to be fitted using the MLPNN. The network training can be performed using either *on-line* or *batch* mode. In the case of on-line training mode, a single input-output example (data-point) from the dataset is used to compute the outputs of the hidden and the output layer nodes and the MLPNN weights are updated on the basis of the corresponding prediction error. This procedure is repeated for all the input-output examples in the example set multiple times till the MLPNN endowed with the desired regression abilities (good prediction accuracy and generalization performance) is secured. In the batch mode operation, all the input-output examples in the example set are presented successively to the MLPNN, and the mean

prediction error is calculated, which is utilized for the weight updation. This procedure is repeated till an MLPNN model with the above-stated regression abilities is obtained. The most widely used weight adjustment algorithm is known as “error-back-propagation (EBP)” algorithm proposed by Rumelhart et al. (1986). It is based on the *generalized delta rule*. The field of ANN virtually exploded with a huge number of applications after this algorithm was proposed.

2. *Prediction Mode*: Once the training mode is successful, the trained MLPNN model can be used to predict the outputs of a new set of inputs. This input set is not a part of the original set used in the training mode. The ability of accurately predicting the outputs of the input set used in training is termed “recall” ability while the ability to accurately predict the outputs of a new input set is called “generalization” ability. It is at most important that the trained MLPNN is capable of generalization.

2.3.1.2 Error-back-propagation (EBP) Algorithm for Training MLPNN

Consider a single hidden layer multiple input-single output (MISO) MLP network as shown in Figure 2.3. The MISO example data set that is available for training the MLP network model is given as:

$$J = \ (\boldsymbol{x_r}, y_r) \tag{2.2}$$

It consists of *R* input-output patterns ($r = 1, 2, \ldots, R$); $\boldsymbol{x}$ (= $[x_1\ , x_2, \ldots, x_M]^{\mathrm{T}}$) refers to an *M*-dimensional vector of independent variables/ predictors, and the corresponding scalar output y_r. An overview of the generalized delta rule and EBP-algorithm based training of an MLPNN is given below.

According to the generalized delta rule, the weights of MLPNN are adjusted in proportion to the negative of the error gradient. i.e,

$$\Delta w_{ij} = -\eta \frac{\partial E_k}{\partial w_{ij}} \tag{2.3}$$

$$\mathrm{w}_{\mathrm{ij}}(t+1) = \mathrm{w}_{\mathrm{ij}}(t) + \Delta \mathrm{w}_{\mathrm{ij}}(t) \qquad (2.4)$$

where,

Δw_{ij} = magnitude by which any weight between units *i* and *j* in two successive layers is to be changed, η = learning rate ($1 \geq \eta > 0$) and E_k = prediction error dependent error function evaluated when *k* th input pattern is applied to the network function defined as

$$E_k = \frac{1}{2}\sum_{l=1}^{s}(y_{kl} - \hat{y}_{kl}^{o})^2 \qquad (2.5)$$

Irrespective whether the destination neuron *j* belongs to the hidden or an output layer, the weight updation rule follows the same basic principle given as: $\Delta \mathrm{w}_{\mathrm{ji}}(t)$

$$\begin{bmatrix} Magnitude\ \ of\ weight \\ correction\ at\ training \\ iteration, t,\ \ (\Delta w_{\mathrm{ij}}\ (t)) \end{bmatrix} = \begin{bmatrix} learning \\ rate, \eta \end{bmatrix} \times \begin{bmatrix} Scaled\ error \\ w.r.t. \\ ith\ node \end{bmatrix} \times \begin{bmatrix} Output\ of\ jth \\ node \end{bmatrix}$$

(2.6)

Error-back-propagation (EBP) based stepwise procedure of MLPNN training

Step 1. Initialize hidden and output layer weights to small random values, e. g., between -1 and +1. The MLP architecture contains N, *M*, and *S* number of inputs, hidden and output nodes. The logistic sigmoid transfer function is considered at both hidden and output later nodes.

Step 2. Apply k^{th} input pattern $\boldsymbol{x_k} = [x_{k1}, x_{k2,\ldots,}\ x_{kR}]$ to the network input layer.

Step 3. Compute the weighted-sum of inputs (activation level) for the individual nodes in the hidden layer according to

$$net_{kj}^{h} = \sum_{i=0}^{N} w_{ji}^{h} x_{ki}\ ;\ j = 1,2,\ldots,M \qquad (2.7)$$

where, net_{kj}^{h}denotes the weighted-sum of inputs for the hidden node *j* when *k*th input pattern is presented to the network.

Step 4. Transform the weighted-sum using the logistic sigmoid transfer function to compute the hidden node outputs as

$$\hat{y}_{kj}^{h} = \frac{1}{1+\exp(-net_{kj}^{h})}; \ j = 1,2,...,M \tag{2.8}$$

Step 5. Compute the weighted-sum of inputs for the individual output layer nodes $(l = 1,2, ... , S)$ as

$$net_{kl}^{o} = \sum_{j=0}^{M} w_{lj}^{o}\hat{y}_{kj}^{h}; \ l = 1, 2,...,S \tag{2.9}$$

Step 6. Transform the net activations of the output layer units using the logistic sigmoid transfer function. They form the network outputs.

$$\hat{y}_{kl}^{o} = \frac{1}{1+\exp(-net_{kl}^{o})}; \ l = 1,2,...,S \tag{2.10}$$

Step 7. Compute the scaled-error for the output layer units as

$$\delta_{kj}^{o} = (y_{kl} - \hat{y}_{kl})\hat{y}_{kl}(1-\hat{y}_{kl}^{o}); \ l = 1,2,...,S \tag{2.11}$$

where y_{kl} refers to the desired output of neuron *l* when input vector $\boldsymbol{x}_k$ is applied to the input nodes.

Step 8. Compute the scaled-error for the neurons in the hidden layer according to

$$\delta_{kj}^{h} = \hat{y}_{kj}(1-\hat{y}_{kj})\sum_{l=1}^{S}\delta_{kl}^{o}w_{lj}^{o}; \ j = 0,1,...,M \tag{2.12}$$

Step 9. Update the weights between the output and hidden layer nodes as

$$w_{lj}^{o} = w_{lj}^{o}(t) + \eta\delta_{kl}^{o}y_{kj}^{h} + \alpha[w_{lj}^{o}(t) - w_{lj}^{o}(t-1)]; \ j = 0,1,...,M; l = 1,2,...,S \tag{2.13}$$

Step 10. Update the weights between hidden and input layers as given below

$$w_{ji}^{h}(t+1) = w_{ji}^{h}(t) + \eta\delta_{kj}^{h}x_{ki} + \alpha[w_{ji}^{h}(t) - w_{ji}^{h}(t-1)];\ \ i = 1,2,\ldots,N;\ \ j = 1,2,\ldots,M \tag{2.14}$$

Step 11. Repeat steps (2-10) with all the input patterns over several iterations till the preselected convergence criterion is satisfied.

2.3.1.3 Securing an MLPNN model with best prediction and generalization performance

The prediction and generalization ability of a MLPNN model is examined using three statistical metrics, namely *coefficient of correlation* (*CC*), *root mean square error* (*RMSE*) (%), and *mean absolute percent error* (*MAPE*). These are calculated using the desired output values and the subsequent model predicted output magnitudes. The *RMSE* is evaluated as follows:

$$\mathrm{RMSE_r} = \sqrt{\frac{\sum_{r=1}^{R}(y_r - \hat{y}_r)^2}{R}} \tag{2.15}$$

where, $RMSE_r$ $(r = 1,2,\ldots.,R)$ refers to the *RMSE* pertaining to the *R* input-output patterns.

The *MAPE* and *CC* are evaluated according to the following expressions (2.16) and (2.17) respectively.

$$MAPE_r(\%) = \frac{1}{R}\sum_{r=1}^{R}\left|\frac{y_r - \hat{y}_r}{y_r}\right| \times 100 \tag{2.16}$$

$$CC_r = \frac{\sum_{r=1}^{R}(y_r - \overline{y}_r)(\hat{y}_r - \hat{y}_r')}{\sqrt{\sum_{r=1}^{R}(y_r - \overline{y}_r)^2}\sqrt{\sum_{r=1}^{R}(\hat{y}_r - \hat{y}_r')^2}} \tag{2.17}$$

where, r denotes the index of the input-output patterns in the range $(r = 1,2,\ldots.,R)$; $MAPE_r$ and CC_r refers to the *MAPE* and CC values pertaining to the R input-output patterns $(r = 1,2,\ldots.,R)$ respectively; y_r and $\overline{y}_r$ are the corresponding desired (target) output value and its mean value respectively; $\hat{y}_r$ and $\hat{y}_r'$ are the magnitudes of the model predicted output and its corresponding mean value respectively when r^{th} input pattern is used to calculate the output of the r^{th} model predicted output.

The following factors related to the network structure and the EBP-specific training algorithm need to be chosen judiciously for securing an MLPNN model possessing good prediction accuracy and generalization capability.

- Structural factors: (i) number of hidden layers, (ii) number of nodes/neurons in each hidden layer, and (iii) the transfer function at hidden layer nodes.
- Error Back Propagation training algorithm related parameters: learning rate (η), training cycles and momentum coefficient (μ).

Commonly, these factors are varied systematically and the effect of this variation on the prediction accuracy and generalization performance of the network model is rigorously assessed. The set of factors and parameters leading to an MLPNN model with optimal prediction and generalization performance is selected. The greater details of the MLPNN training procedure and the related issues have been provided for example, in Zurada (1992), Bishop (1994) and Tambe et al. (1996).

During training of an MLPNN, it is observed that as the number of training cycles increases, the magnitude of *CC* increases with an associate decrease in the *RMSE* value. However, training an MLPNN over a large number of iterations has a drawback that despite resulting in high *CC* and low *RMSE* magnitudes, the resultant model is incapable of predicting correctly the outputs of a new set of inputs. This phenomenon is known as “overtraining” owing to which the MLPNN performs poorly at generalization (ability to predict correctly outputs of new set of inputs). Similar loss of generalization is witnessed if there exists more-than-necessary number of nodes/neurons in the hidden layer (s) of an MLPNN architecture, which is known as “over-parameterization.” The result of overtraining and/or over-parameterization is an “overfitted” MLPNN model unsuitable for any use since it does not possess the much-desired generalization capability. The loss of generalization by an over-trained MLPNN model occurs since it captures the micro details such as noise in the data, at the cost of learning smooth trends in the data.

Over-fitting can be avoided by partitioning the available example dataset into *training*, and *test* subsets. The training set is utilized to train the MLPNN model while the test set is used for checking the generalization performance of network at each iteration. In some cases, an additional set termed “validation set” is carved out from the example set and used in validating the model being trained. During training, the above-stated MLPNN’s structural and EBP algorithm related parameters are optimized in a manner such that the (a) the coefficient of correlation (*CC*) magnitudes pertaining to both training and test sets are high and comparable, and (b) the *MAPE/ RMSE* magnitudes pertaining to both training and test sets are low and comparable while the validation set is used to assess the generalization capability of the network.

Owing to its phenomenal nonlinear function approximation capability, MLPNNs have been employed exhaustively in all science and engineering/technology branches as also finance and economics. A representative list of the most recent applications of MLPNN in chemical and biochemical areas is given in **Table 2.2.**

Table 2.2: Modern chemical engineering/technology applications of multi-layered perceptron neural network

SN	MLP Application	Study	Reference
1	Steady-state and dynamic process modeling/prediction	• Prediction of solubility of N-alkanes in supercritical CO_2 using RBF-ANN and MLP-ANN	Abdi-Khanghaha et al. (2018)

		• Use neural networks for problem solving	Chitra (1993)
		• Forecasting of ozone level in time series using MLP model with a novel hybrid training algorithm	Wang and Lu (2006)
		• Annual electricity consumption forecasting by neural network in high energy consuming industrial sectors	Azadeh (2008)
		• Application of steady-state and dynamic modeling for the prediction of the BOD of an aerated lagoon at a pulp and paper mill: Part II. Nonlinear approaches	Patricia et al. (2004)
		• Prediction of heat capacities of ionic liquids using chemical structure-based networks	Harooni et al. (2017)
		• Comparative post fatigue residual property predictions of reinforced and unreinforced poly (ethylene terephthalate) fibers using artificial neural networks	Averett et al. (2010)
		• Seed yield prediction of sesame using artificial neural network	Emamgholizadeh et al. (2015)
		• Neural network modeling for predicting brewing fermentations	Syu et al. (1994)

2	Process Monitoring	• The use of a multilayer perceptron (MLP) for modeling the phenol removal by emulsion liquid membrane	Messikha N. et al. (2017)
		• Neural modeling of chemical plant using MLP and B-spline networks	Lightbody et al. (1997)
		• A neural network approach to analyzing multi-component mixtures	Broten and Wood (1993)
		• The use of artificial intelligence combiners for modeling steel pitting risk and corrosion rate	Chou et al. (2017)
		• Estimation of gross calorific value of coals using artificial neural networks	Patel et al. (2007)
		• Modeling and monitoring of batch processes using PCA assisted GRNN	Kulkarni et al (2004)
3	Model based nonlinear process control	• Neural for modeling and control of reactive distillation, IFAC proceedings volumes	Engell and Dadhe (2001)
		• Implementation of neural network predictive control to a multivariable chemical reactor	Yu and Gomm (2003)
		• Robust nonlinear control with neural networks	Ramasamy et al. (1995)
		• Hydroxylation of phenol to	Tendulkar et al.

		dihydroxybenzenes: Development of artificial neural-network-based process identification and model predictive control strategies for a pilot plant scale reactor	(1998)
		• Modeling of coke fractionators	Blaesi J. and Jensen B. (1992).
		• Prediction of all India Summer monsoon rainfall using error-back-propagation neural networks	Venkatesan et al (1997)
4	Process Identification	• Fischer–Tropsch synthesis with $Co/SiO_2–Al_2O_3$ catalyst and steady-state modeling using artificial neural networks	Sharma et al. (1998)
		• Neural networks for the identification of MSF desalination plants	Selvaraj et al. (1995)
		• Fluid property predictions with the aid of neural networks	Lee and Chen (1993)
		• A numerical correlation development study for the determination of Nusselt numbers during boiling and condensation of R134a inside smooth and corrugated tubes	Balcilar et al. (2013)
		• Data processing by neural networks in quantitative chemical analysis	Bos et al. (1993)

		• Identification of gases with classical pattern-recognition methods and artificial neural networks	Niebling (1994)
5	Process optimization	• Artificial chemical reaction optimization of neural networks for efficient prediction of stock market indices	Nayak et al. (2017)
		• Correction of mass spectral drift using artificial neural networks	Goodacre et al. (1996)
		• Hybrid modeling of fermentation processes: A Study on the use of modular neural networks for modeling cells reaction kinetics	Perest et al. (2004)
		• Utilization of apricot seed in (CO) combustion of lignite coal blends	Buyukada and Aydogmus (2018)
6	Fault detection and diagnosis (FDD)	• Optimum parameters for fault detection and diagnosis system of batch reaction using multiple neural networks	Tan et al. (2012)
		• Counter-propagation neural networks for fault detection and diagnosis	Vora et al (1997)
		• A novel fault diagnosis technique for photovoltaic systems-based on ANNs	Chine et al. (2016)

		• Fault detection and isolation for PEM fuel cell stack with independent RBF model	Kamal et al. (2014)
		• Model selection and fault detection approach based on Bayes decision theory: Application to changes detection problem in a distillation column	Chetouani (2014)
7	Quantitative structure-activity/ property relationships (QSAR/ QSPR)	• Determination of binary diffusion coefficients of hydrocarbon mixtures using MLP and ANFIS networks based on QSPR method	Abbasi and Eslamloueyan (2014)
		• Chemometrics tools in QSAR/QSPR studies: A historical perspective	Yousefinejad and Hemmateenejad (2015)
		• Assessment for multi-endpoint values of carbon nanotubes: Quantitative nanostructure-property relationship modeling with norm indexes	Wang et al. (2017)
		• QSPR prediction of the solubility of CO_2 and N_2 in common polymers	Golzar et al. (2013)
8	Soft-sensor development	• Application of soft computing techniques to multiphase flow measurement: A review	Yan et al. (2018)
		• Soft-sensing estimation of	Canete et al.

		plant effluent concentrations in a biological wastewater treatment plant using an optimal neural network	(2016)

2.3.2. Support Vector Regression (SVR)

Support vector regression is derived from the *Support vector machine* (SVM) formalism, which is a statistical learning-based methodology to perform supervised nonlinear classification (Vapnik, 1995). Towards performing the stated classification, SVM first maps the coordinates of the objects (example data) into a high dimensional feature space. This is done by employing nonlinear functions termed *kernels* or *features*. Mapping into a high dimensional space has an advantage that in this space it is easier to separate two classes with the help of a linear classifier as done in common practice. The SVR formalism uses same concepts but for conducting nonlinear regression (function approximation). In essence, it tries to secure a multiple input – single output (MISO) function possessing following characteristics.

(i) Its location is such that it deviates maximally from all training data.

(ii) The shape of the function is as flat as possible. A function with a small weight (parameter) vector assures its flatness.

Using an example data set $G = (\boldsymbol{x}_r, y_r)_{r=1}^{R}$ that consists of R input-output patterns (r = 1, 2,…, R); $\boldsymbol{x}$ (= $[x_1\ , x_2,\ldots, x_M]^{\mathrm{T}}$) refers to an M-dimensional vector of independent predictors/ variables, and y_r denotes the scalar output, the SVR formalism aims at approximating a function $f(\mathbf{x})$. Here, the aim is to map the input data, $(\boldsymbol{x}_r)_{r=1}^{R}$, nonlinearly into a high dimensional feature space ($\boldsymbol{\Phi}$) and perform a linear regression in the $\boldsymbol{\Phi}$-space as given by:

$$y_r = f(\boldsymbol{x}_r) = \boldsymbol{w} \cdot \Phi(\boldsymbol{x}_r) + b \qquad (2.18)$$

where $\boldsymbol{w}$ is the vector of function coefficients, b is a real-valued constant representing the threshold value and $\Phi(\boldsymbol{x}_r)$ refers to a set of nonlinear transformations. A loss function (LF) is defined and used for estimating the quality of the regression performed by the SVR. The commonly used LF is known as "ε-insensitive" function defined as:

$$|y_r - f(\boldsymbol{x}_r)|_\varepsilon \;=\; \begin{cases} \frac{0}{|y_r - f(\boldsymbol{x}_r)| - \varepsilon} & \begin{array}{c} if\,|y_r - f(\boldsymbol{x}_r)| \le \varepsilon \\ \text{otherwise} \end{array} \end{cases} \qquad (2.19)$$

Consider Figure 2.4 showing the principle employed by the SVR algorithm wherein it tries to position a tube of radius ε around the nonlinear regression function. The tube encloses a region is termed an "ε-insensitive zone"; here ε essentially denotes the tolerance to the deviation from the fitted function. In this formulation, errors are defined as the deviations larger than ε. The coefficient vector ($\boldsymbol{w}$) and constant b appearing in equation 2.18 are estimated from the training data. It is done by minimizing an empirical risk function (ERF). In simple terms, the empirical risk signifies the prediction error pertaining to the training data. It is termed "empirical" since it is evaluated using a part of the data surrogate for the true distribution. The ERF is defined as:

$$R_{emp}(\boldsymbol{w}) = \frac{1}{2}\,||\boldsymbol{w}||^2 + \frac{C}{N}\;\sum_{r=1}^{R}\;|y_r - f(\boldsymbol{x}_r)\,|_\varepsilon \qquad (2.20)$$

where, C denotes the regularization constant, which determines the trade-off between the training data set error and complexity of the model. A detailed account of the SVR and its implementation can be found in e.g., Vapnik, 1995, and Ivanciuc, 2007. Here, we present the final form of the SVR-based regression function given as:

$$f(\boldsymbol{x},\boldsymbol{w}) = \; f(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\lambda}^*) = \sum_{r=1}^{R}\; f(\boldsymbol{\lambda}_{\boldsymbol{r}}^* - \boldsymbol{\lambda}_r)K\,(\boldsymbol{x}_{\boldsymbol{r}},\boldsymbol{x}\,)\; + \; b \qquad (2.21)$$

where

- R refers to the number of training data points

- $K(\boldsymbol{x}_r, \boldsymbol{x})$ refers to the kernel function representing the dot product in the feature space (**Φ**);
- λ_r, λ_r* (>0) are the coefficients (Lagrange multipliers) satisfying the condition $\lambda_r \lambda_r$* = 0 (r = 1, 2,…, R).

The vector $\boldsymbol{w}$ is represented in terms of the Lagrange multipliers λ_r and λ_r*.

In equation (2.20), only a few of the coefficients, (λ_r* − λ_r), possess non-zero magnitudes, and the respective input vectors, $\boldsymbol{x_r}$, are known as "support vectors (SVs)." These are the important data points in the training set that signify the most enlightening observations squeezing the information content of the training set. It may be noted that the observations lying close to the prediction by the SVR model and located within the limit defined by the ε-tube are ignored and thus the ultimate model is determined on the basis of only the support vectors.

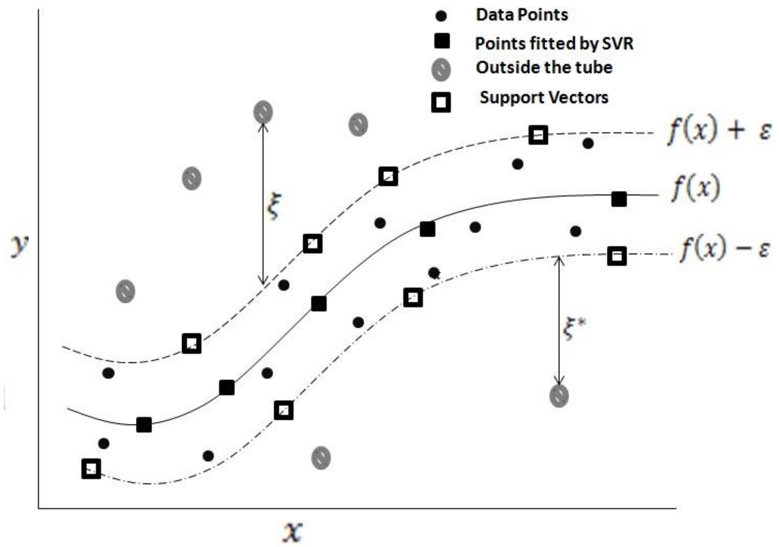


Figure 2.4: Schematic presentation of support vector regression using ε-insensitive loss function.

The salient features which make the SVR an attractive method for developing exclusively data-driven MISO models are specified below (Sharma and Tambe, 2014).

- The method minimizes a quadratic function possessing a single minimum. This feature avoids difficulties allied with finding a solution in the presence of multiple local minima.
- It ensures good generalization ability and sparseness of the regression model, robustness of the solution, as also an automatic control of the solution complexity.
- Provides an explicit knowledge of the *support vectors* that are crucial in defining the regression function; the knowledge of SVs helps in the understanding of the regression model in terms of the training data.

Owing to their several attractive features the SVM/SVR methodologies have been used in various process engineering tasks. A representative listing of these studies is presented in **Table 2.3.**

Table 2.3: Selective chemical engineering/technology applications of support vector regression

SN	**SVR Application**	**Study**	**Reference**
1	Steady-state and dynamic process modeling/prediction	• Daily natural gas consumption forecasting based on a structure-calibrated support vector regression approach	Bai and Li (2016)
		• Prediction of ^{0}API values of crude oils by use of saturates/aromatics/resins/asphaltenes analysis: Computational-intelligence based models	Goel et al. (2016)
		• Rotary drying process modeling and online compensation	Wang et al. (2015)
		• SVR-based prediction of point gas	Gandhi et al

		hold-up for bubble column reactor through recurrence quantification analysis of LDA time-series	(2008)
		• Real-time product quality control for batch processes based on stacked least-squares support vector regression models	Zhang et al. (2012)
2	Process Monitoring	• Simulation of subcooled flow boiling with an SVR based interphase mass transfer model	Dong et al. (2017)
		• Just-In-Time statistical process control: Adaptive monitoring of vinyl acetate monomer process	Kano et al. (2011)
		• Online sensor for monitoring a microalgal bioreactor system using support vector regression	Nadadoor et al. (2012)
		• Evaluation using online support-vector- machines and fuzzy reasoning. Application to condition monitoring of speeds rolling process	Bouhouche et al. (2010)
3	Model based nonlinear process control	• Adaptive nonlinear model predictive control using an on-line support vector regression updating Strategy	Wang et al. (2014)
		• High ash char gasification in thermo-gravimetric analyzer and prediction of gasification performance parameters using computational intelligence	Patil-Shinde et al (2016)

		formalisms	
		• Nonlinear multivariable modeling of locomotive proton exchange membrane fuel cell system	Li et al. (2014)
		• Multi-parametric Metamodels for model predictive control of chemical processes	Shokry et al. (2011)
		• A novel unified correlation model using ensemble support vector regression for prediction of flooding velocity in randomly packed towers	Liu et al (2014)
4	Process Identification	• Black-box identification of a pilot-scale dryer model: A support vector regression and an imperialist competitive algorithm approach	Sałat et al. (2017)
		• Using machine learning algorithms to predict the pressure drop during evaporation of R407C	Khosravi et al. (2018)
		• Computational intelligence-based models for prediction of elemental composition of solid biomass fuels from proximate analysis	Ghugare et al. (2017)
		• Development of support vector regression (SVR)-based correlation for prediction of	Gandhi et al. (2007)

		overall gas hold-up in bubble column reactors for various gas–liquid systems	
5	Process optimization	• Support vector regression modelling and optimization of energy consumption in carbon fiber production-line	Golkarnarenji et al. (2018)
		• Hybrid process modeling and optimization strategies integrating neural networks/SVR and genetic algorithms: study of benzene isopropylation on H-beta catalyst	Nandi et al. (2004)
		• Optimization techniques for improving the performance of information retrieval system	Badhe et al (2014)
		• Forecasting of coal seam gas content by using support vector regression based on particle swarm optimization	Meng et al (2014)
		• A novel modeling approach to optimize oxygen-steam ratios in coal gasification process	Arabloo et al (2015)
		• Simulation and optimization of a full-scale Carrousel oxidation ditch plant for municipal wastewater treatment	Xie et al. (2011)
		• Prediction of silicon content in hot metal using support vector regression based on chaos	Tang et al. (2009)

		particle swarm optimization	
6	Fault detection and diagnosis (FDD)	• Bond graphs for the diagnosis of chemical processes	Bouamama et al. (2012)
		• DICA enhanced SVM classification approach to fault diagnosis for chemical processes	Monroy et al. (2009)
		• Dynamic fault prognosis for multivariate degradation process	Wang et al. (2018)
7	Quantitative structure-activity/ property relationships (QSAR/QSPR)	• Nonlinear QSAR models with high-dimensional descriptor selection and SVR improve toxicity prediction and evaluation of phenols on Photobacterium phosphoreum	Zhou et al. (2015)
		• Support vector regression based QSPR for the prediction of some physicochemical properties of alkyl benzenes	Yang (2005)
		• Application of QSPR for prediction of percent conversion of esterification reactions in supercritical carbon dioxide using least squares support vector regression	Esteki et al., (2005)
		• Boosting support vector regression in QSAR studies of bioactivities of chemical compounds	Zhou et al. (2006).
8	Soft-sensor development	• A deep learning-based data driven soft sensor for	Gopakumar et al. (2018)

		bioprocesses	
		• A nonlinear soft sensor based on modified SVR for quality estimation in polymerization	Lee et al. (2003)
		• Robust soft sensors based on integration of genetic programming, analytical neural networks, and support vector machines	Kordon et al. (2002)
		• A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses	Yu J. (2012)
		• DoE framework for catalyst development based on soft computing techniques	Valero et al. (2009)
		• Soft-sensor development for fed-batch bioreactors using support vector regression	Desai et al. (2006)

2.3.3. Genetic programming (GP)

Genetic Programming is a member of the stochastic optimization strategies termed “evolutionary algorithms” that follow the “survival of the fittest” principle of the Darwin’s theory of evolution along with the genetic propagation of characteristics. In its original form, GP was proposed by Koza (1992) for automatically generating computer codes that would do pre-specified tasks. Genetic programming’s second application, termed “symbolic regression (SR),” has been exploited in the studies presented in this thesis. The novel feature of the GP-based SR (GPSR) is that given an example data set containing inputs (independent/predictor variables) and the corresponding outputs (dependent/response variables), it possesses a striking capability of searching as also optimizing an appropriate structure (form) of a linear or nonlinear data-fitting function and all the parameters related with it. More importantly, the GP approach performs the affirmed search and optimization without making any assumptions about the structure of the linear or nonlinear data-fitting function. The details of the exclusively data-driven modeling problem solved by the GPSR are provided below.

Consider a multiple input - single output (MISO) dataset, $G = (\boldsymbol{x}_r, y_r)_{r=1}^{R}$ that consists of R input-output patterns (r = 1, 2,…, R); $\boldsymbol{x}$ (= [x_1 , x_2,…, x_M]$^{\mathrm{T}}$) refers to an M-dimensional vector of independent variables/predictors, and y_r is the corresponding scalar output. The task of the GPSR is to secure a suitable linear or nonlinear function (*f*) as given below that best fits the dataset, *G*:

$$y = f(x_1, x_2 \dots, x_M; \beta) \tag{2.22}$$

where β denotes an L-dimensional vector of function parameters,

$\beta = [\beta_1, \beta_2, \dots, \beta_L]^{\mathrm{T}}$.

To search and optimize the form of the data-fitting function, f, and the related parameters (β), GPSR starts to mimic a natural evolution step rather in a abstract manner. It stochastically generates an initial population of N_{pop} number of candidate (probable) solutions (mathematical expressions/ models) to the symbolic regression problem defined in Eq. (2.22). Commonly, tree structures are used for articulating these candidate solutions. Each one of these trees is constructed randomly using *function* and *terminal* nodes. While the first type of nodes represents mathematical operators, the ones belonging to the latter type define predictors (input/independent variables, $\{x_r\}$) and parameters, β, of a candidate expression. The operator set comprises *addition, subtraction,* *multiplication*, *division*, *logarithm*, *exponentiation*, and *trigonometric* operators. In GPSR, it is necessary that each candidate solution is evaluated for its ability (fitness) in fulfilling the given data-fitting task. This is done using a fitness function that computes a score (termed *fitness score/ value*) measuring how well a candidate solution is fitting the example set data. In the succeeding step, candidate solutions breed among themselves in frequency, which is directly proportional to their fitness. This is known as "crossover" operation that produces two new offspring (i.e., new candidate solutions). As in Natural evolution, the new offspring may be subjected to *mutation*, whereby their genetic material (contents of the tree) is altered randomly, albeit to a small extent. The above stated process of crossover and mutation is iterative, adaptive and open ended (McConaghy, 2010). Over time, the best candidate solutions (i.e., possessing high fitness) will survive. A typical step wise GPSR implementation procedure is given below. The corresponding flow-chart is depicted in as follows Figure 2.5.

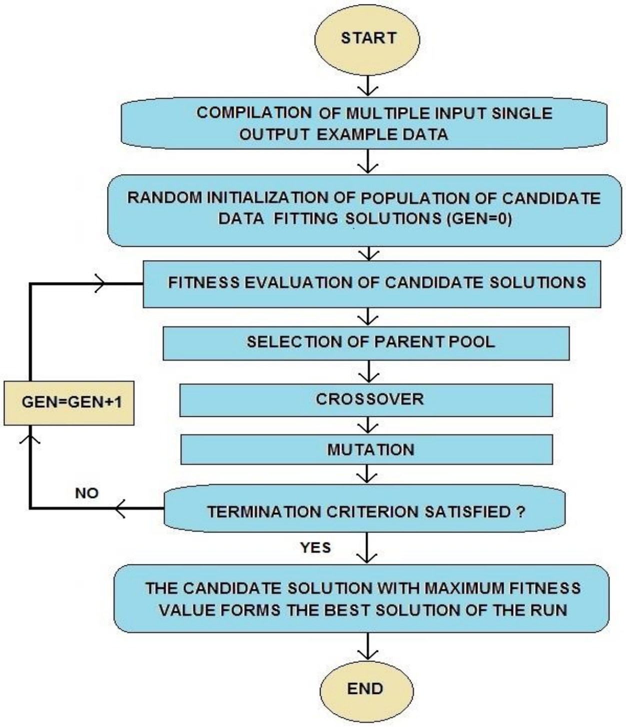

START
COMPILATION OF MULTIPLE INPUT SINGLE OUTPUT EXAMPLE DATA
RANDOM INITIALIZATION OF POPULATION OF CANDIDATE DATA FITTING SOLUTIONS (GEN=0)
FITNESS EVALUATION OF CANDIDATE SOLUTIONS
SELECTION OF PARENT POOL
CROSSOVER
MUTATION
GEN=GEN+1
NO
TERMINATION CRITERION SATISFIED ?
YES
THE CANDIDATE SOLUTION WITH MAXIMUM FITNESS VALUE FORMS THE BEST SOLUTION OF THE RUN
END

Figure 2.5: A brief algorithm of genetic programming

(i) Initialization: This step randomly generates an initial population of N_{pop} number of mathematical expressions (candidate solutions) of different structures and lengths. The different approaches used for this random generation are, *half* (ramped half-and-half) and *full* initialization (Khandelwal et al., 2017; Langdon et al., 2008).

(ii) *Fitness evaluation*: Fitness score of each candidate solution is determined to measure how accurately it predicts the target outputs specified in the example (training) dataset. The fitness of a solution can be evaluated using the following mathematical formula:

$$S_r = \frac{1}{1+RMSE_r} \text{ where } r = 1,2, \ldots., N_{\text{pop}} \qquad (2.23)$$

Here, S_r is the fitness score of the r^{th} candidate (instantaneous or present moment) solution.

(iii) *Selection:* In this step, a pool of candidate solutions (termed *parent* pool) is formed to undergo crossover operation. Here, from the current population of candidate solutions, parents are selected based on their high fitness scores relative to other candidates. There exist several methods of selection such as *Roulette-wheel selection*, *tournament selection*, and *lexicography parsimony pressure selection* (Karakus, 2011). Upon forming the parent pool, its constituents are ranked according to their fitness scores.

(iv) *Crossover*: It is the most vital genetic operator since it produces offspring candidate solutions. It is performed as described below.

(a) Choose randomly two candidate solutions (parents) from the ranked parent population.

(b) Splice each parent tree from the chosen pair at a randomly selected node into two segments.

(c) Mutually exchange the spliced segments between the parents and join them to produce two offspring trees (expressions). This crossover procedure is iterated until a specified number of offspring (= N_{pop}) are generated.

(v) *Mutation*: This genetic operator introduces random changes in the structure of the offspring candidate solution trees formed in the previous step. In mutation, a function (terminal) node is interchanged by another one of the similar types. Mutation helps in maintaining the genetic diversity in the offspring population and thereby broadens the search for good data-fitting models as well as prevents the premature convergence of the GPSR run. The mutated offspring population represents a new generation of candidate solutions and thus the generation index increases by unity.

(vi) *Termination*: Perform Steps (ii) to (v) iteratively until either of the following two termination criteria (stopping condition) is satisfied:

(a) GPSR has evolved over a pre-specified number of maximum generations.

(b) The best candidate solution (expression) in the population possessing highest fitness score does not change appreciably over several generations.

A schematic diagram of tree structures and the genetic operations is given in Figure 2.6.

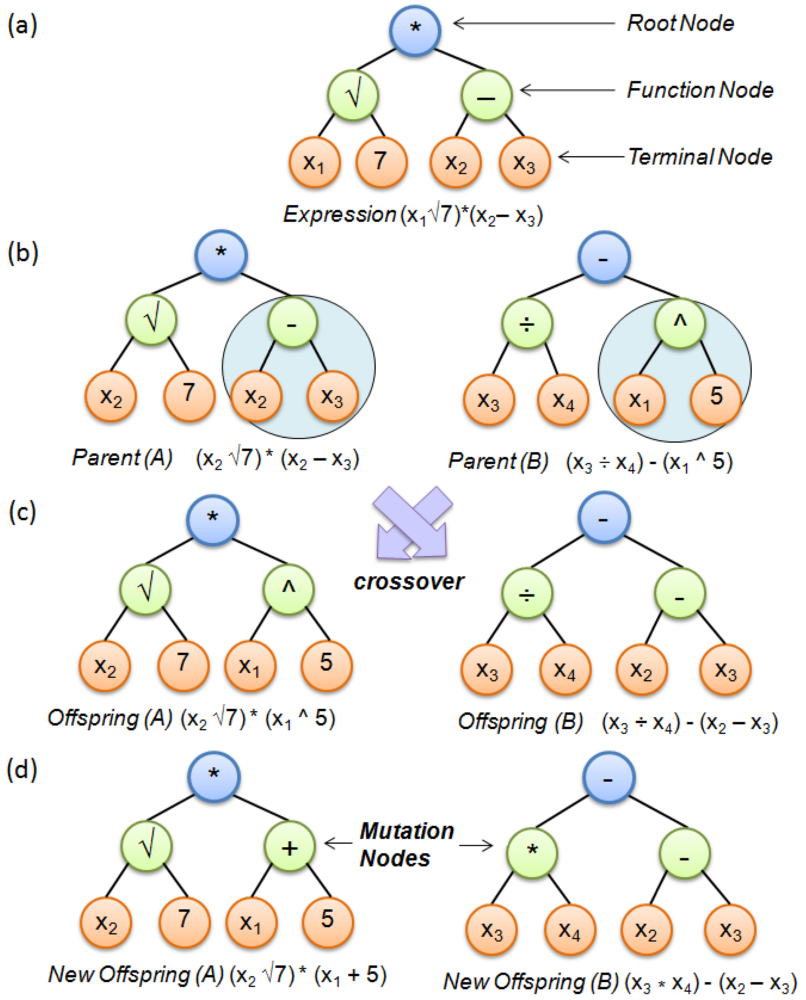

(a)
Root Node
Function Node
Terminal Node
Expression (x1√7)*(x2– x3)
(b)
Parent (A) (x2 √7) * (x2 – x3)
Parent (B) (x3 ÷ x4) - (x1 ^ 5)
(c)
crossover
Offspring (A) (x2 √7) * (x1 ^ 5)
Offspring (B) (x3 ÷ x4) - (x2 – x3)
(d)
Mutation Nodes
New Offspring (A) (x2 √7) * (x1 + 5)
New Offspring (B) (x3 * x4) - (x2 – x3)

Figure 2.6: (a) Illustration of a tree structure, (b) two parent trees undergoing crossover operation, (c) two offspring trees post-crossover, (d) mutation operation on two offspring trees.

In this figure, panel (a) shows a typical tree structure representing an expression “($x_1\sqrt{7}$) *(x_1 - x_3)”. The crossover and mutation are illustrated in panels (b) and (c), respectively. Greater details of the GPSR method and its implementation procedure can be found in Poli et al. (2008), Ghugare et al. (2014), Vyas et al. (2015), and Goel et al. (2015). It may be noted here that compared to ANNs and SVR, the GP formalism, despite its several attractive and novel features, has been employed infrequently for performing data-driven modeling. A representative list of GPSR applications in chemical engineering/technology area is provided in **Table 2.4**.

Table 2.4: Selected Chemical engineering/technology applications of genetic programming

SN	GP application	Study	Reference
1	Steady-state and dynamic process modeling/prediction	• Artificial intelligence-based modeling of high ash coal gasification in a pilot plant scale fluidized bed gasifier	Patil-Shinde et al. (2014)
		• Modeling of vaporization enthalpies of petroleum fractions and pure hydrocarbons using genetic programming	Parhizgar et al. (2013)
		• Toward genetic programming (GP) approach for estimation of hydrocarbon/water interfacial tension	Rostami et al. (2017)
		• Prediction of HHV of coals of different ranks and from diverse geographies	Ghugare and Tambe (2017)
		• Prediction of reactivity ratios in free radical copolymerization from monomer resonance–polarity (Q–e) parameters: genetic programming-based models.	Shrinivas et al. (2016)

2	Process Monitoring	- A novel approach for estimation of solvent activity in polymer solutions using genetic programming	Tashvigh et al. (2015)
		- Real-time monitoring of thin film microstructure in chemical vapor deposition using a modified moving horizon estimation	Xiong and Gallivan (2008)
3	Model based nonlinear process control	- Multi-gene genetic programming based predictive models for municipal solid waste gasification in a fluidized bed gasifier	Pandey et al. (2015)
		- PID based nonlinear processes control model uncertainty improvement by using Gaussian process model	Chan et al. (2016)
		- Non-linear principal components analysis using genetic programming	Hiden et al(1998)
		- Modeling and temperature control of rapid thermal processing	Dassau et al. (2006)
		- Automated nonlinear model predictive control using genetic programming	Grosman and Lewin et al. (2002)

4	Process Identification	• Consider genetic programming for process identification	Kulkarni et al. (1999)
		• Process identification using genetic programming: a case study involving fluidized catalytic cracking (FCC) unit	Nandi et al. (2000)
		• Identification of algebraic and state space models using genetic programming	Tun and Lakshminarayan (2004)
		• Genetic programming-based models for prediction of vapor-liquid equilibrium	Patil-Shinde et al (2018)
		• Use genetic programming for selecting predictor variables and modeling in process identification	Verma et al. (2016)
		• The detection of caffeine in a variety of beverages using Curiepoint pyrolysis mass spectrometry and genetic programming	Goodacre and Gilbert et al. (1999)
5	Process optimization	• Process structure optimization using a hybrid disjunctive-genetic programming approach	Yuan et al. (2009)
		• Genetic programming assisted stochastic optimization strategies for	Cheema et al (2002)

		optimization of glucose to gluconic acid fermentation	
		• A novel approach for modeling and optimization of surfactant/polymer flooding based on Genetic Programming evolutionary algorithm	Bahrami et al. (2016)
		• Soft-computing models for soot-blowing optimization in coal-fired utility boilers	Pena et al. (2011)
6	Fault detection and diagnosis (FDD)	• Feature generation using genetic programming with application to fault classification.	Guo et al. (2005)
		• Fault detection using genetic programming	Zhang et al (2005)
		• Simulation-based fault propagation analysis: Application on hydrogen production plant	Gabbar et al. (2014)
7	Quantitative structure-activity/ property relationships (QSAR/QSPR)	• A genetic programming-based QSPR model for predicting solubility parameters of polymers	Koç D. and Koç M. (2015)
		• Genetic programming based quantitative structure–retention relationships for the prediction of Kovats retention indices	Goel et al. (2015)

8	Soft-sensor development	• Biomass inferential sensor based on ensemble of models generated by genetic programming	Kordon et al. (2004)
		• Softsensor Model for Styrene Polymerization Process and its Application in Model based Control	Ghugare et al (2016)
		• Soft-sensor development for biochemical systems using genetic programming	Sharma and Tambe et al. (2014)

2.4 GUIDELINES FOR IMPLEMENTATION OF CI-BASED MODELING METHODS

In what follows, guidelines for developing exclusively data-driven CI-based models are given. These are useful in developing models possessing excellent prediction and generalization performance.

1. Before implementation of a CI based modeling method, it is necessary to ensure that the quantum of the data available is sufficient for training, and testing the model. It may be noted that bigger the size of these sets, higher is the probability of obtaining a model with good prediction accuracy and generalization capability. It is absolutely necessary that the available data are statistically well-distributed and cover the ranges of practical interest. Here, it is important to note that data-driven nonlinear models are incapable of accurate extrapolation beyond the ranges of the data used in their training. Thus, additional data in the regions of interest need to be collected and fresh model should be built rather than attempting the stated extrapolation.
2. It needs to be examined before employing CI-based methods whether the modeling task on hand can be achieved using conventional methods. For

example, it is simple and easy to employ algebraic methods to develop linear models. If the form of a nonlinear model to be developed is known a priori then it is advisable to use nonlinear regression methods such as the Levenberg-Marquardt's nonlinear regression method (Levenberg, 1944; Marquardt, 1963) to estimate its parameters than to develop a CI-based model.

3. Non-analyzed and non-preprocessed data can lead to suboptimal models possessing poor prediction accuracy and generalization ability and therefore analysis and processing of raw data are necessary (Freeman, 1999). Such a preprocessing includes normalization and/or denoising of data, and outlier removal. It is also necessary to analyze the data forming the input space of the models to examine if the inputs are correlated. Typically, linear and nonlinear principal component analyses are performed to identify linearly and nonlinearly correlated inputs. Such analyses assist in (a) reducing the dimensionality of the input space if correlated inputs are indeed present, and (b) reducing the computational load in developing the model. Additionally, proper data representation methods must be employed. For example, continuous versus indicator variables, representation of time-windowed data, de-trending, encoding inputs, filtering etc.
4. The data pre-processing and analysis performed by the model developer and the related methods must be understood fully by the end user. This is required for the optimal usage of the developed models and knowing their strong and weak attributes. The developer must avoid an over-fitted model since such a model is of no practical utility. It is therefore important for the developer to be well-versed with the methods for the detection and prevention of the over-training and over-parameterization that causes the over-fitting of a CI-based model.
5. In the case of an MLPNN, although it is capable of mapping multiple input – multiple output (MIMO) functions, it is not advisable to do so. It

may be noted that in MIMO function approximation, the weights (function parameters) between the input and hidden layer nodes remain same for all outputs. Only the weights between the hidden and output layer change for each output. This essentially indicates that to perform MIMO mapping, the MLPNN has to learn to map more than one nonlinear functional relationship while using the same set of input-to-hidden layer weights. This constraint affects the ability of the MLPNN in constructing a model with good prediction and generalization performance. Thus, it is advisable to use as many multiple input – single output (MISO) models as the number of outputs since it allows not only the weights between hidden and output layers to be different but also those between input and the hidden layers of the MLPNN. This of course increases the modeling effort but the improved prediction and generalization ability of the resultant models justify the efforts.

6. As a basic principle, the developer must strive to build parsimonious models. Such models are less complex since they have less number of terms and parameters in their structure. All other things being equal, a parsimonious model is known to fare well at the generalization when compared with its more complex counterpart. In the case of an MLPNN, complexity of the model can be minimized by using as few hidden layers and units in each one of them as possible. While developing an SVR model, a model with reduced complexity can be constructed by restricting as much as possible the number of support vectors which are the most informative data points. In GPSR, a parsimonious model can be developed by limiting the depth of the trees representing the candidate solutions and using fewer operators.
7. It has been witnessed that no single type of CI-based models (MLPNN, SVR, GP and their variants) are found to be superiorly performing in all application. Therefore, it is advisable that separate optimal models are

built using all three methods and the best performing one among them is chosen for the use.

2.5 GENETIC ALGORITHMS

Genetic algorithms (GA) is a stochastic optimization paradigm that belongs to the class of evolutionary algorithms. In this thesis, GA has been utilized in obtaining optimized conditions for the photocatalytic degradation of pharmaceutical pollutants (see Chapter 3). A detailed description of GA is presented in the ensuing paragraphs**.**

Genetic algorithms (Goldberg, 1989; Holland, 1975) are the most widely used stochastic optimization algorithms. Similar to the genetic programming, GAs are also based on the Darwinian principles of the “survival of the fittest” and the genetic propagation of characteristics over successive generations, followed in the biological evolution of species. The difference between the GP and GA is that while the former performs symbolic regression and used in data-driven modeling, GA is mainly used for function minimization/ maximization.

GAs have been a great success in solving problems involving very large search spaces. GA broadly works in the framework, wherein it considers the environment as an objective function to be maximized/minimized and the individuals in the environment as *candidate solutions* in the search space (Goldberg, 1989). In the context of the study presented in this thesis, a candidate solution represents decision variables appearing in the objective function, which are to be optimized by the GA in a manner such that the objective function is minimized/maximized. In function maximization tasks, the objective function may represent reaction conversion, yield, profit etc while in the function minimization task, the function may refer to the financial loss in process operation, selectivity of the undesired products, etc.

Having developed an optimal CI-based model (f^*), in the next phase the model's input space representing process operating variables ($\boldsymbol{x}$) can be optimized using the GA formalism. Consider for example, an optimization task involving of maximization of an objective function as defined below.

$$Maximize\ y_d = G[f^*(\boldsymbol{x}, \sigma)] \text{ subject to constraints; } x_n^{lower} < x_n < x_n^{upper} \qquad (2.24)$$

where y_d denotes a process performance variable, G denotes an objective function expressed in terms of a CI-based model. It is assumed that the form of the nonlinearf^*, magnitudes of the elements of the parameter vector (σ), and x^{lower} and x^{upper} are known.

The step-wise procedure of GA towards fulfilling the above-stated optimization task is described below.

Step 1 (*Initialization*): Form an initial population of size N_{pop} of candidate (probable) solution strings (chromosomes) whose elements (binary digits or real numbers) are chosen randomly. Set the generation counter to zero. Each chromosome in the population is of the same length and has as many segments (formed by bits or real values) as the decision variables to be optimized by the GA.

Step 2 Evaluate fitness of each chromosome in the population using a pre-specified fitness function. Fitness function assigns a score to the candidate solution, which is directly proportional to the ability of the solution to fulfill the task of objective function maximization/ minimization.

Step 3 (*Selection*): A pool of parent chromosome (of size N_{pop}) strings is formed in this step. The selection of these strings is done using algorithms such as the weighted Roulette-Wheel algorithm (Bäck and Schwefel, 1993; De Jong, 1975). This step in an oversimplified manner mimics the "survival of the fittest" principle of Darwinian evolution. In essential feature of the selection step is to form a pool of relatively fitter chromosomes (parents) from the current population of chromosomes. The probability of a string getting selected in the parent pool is directly proportional to its fitness value. It is possible that multiple copies of fitter chromosomes are included in the parent pool. The formation of parent pool is to select fitter chromosomes for offspring production.

Step 3 (*Crossover*): Crossover is the most important step of GA. It is responsible for generating offspring with genetic material that is different than the parents. It is performed as follows: (a) choose randomly a chromosome pair from the parent pool, (b) randomly select a crossover point along the length of each parent chromosome and slice the strings at this point to generate two substrings per parent, (c) exchange the substrings mutually between the parent strings and join them to obtain two offspring.

Step 4: Repeat step 3 until the total number of offspring generated equals N_{pop} following which the offspring population is merged with the parent population. This gives a post-merger population with $2 \times N_{\text{pop}}$ number of chromosomes.

Step 5 (*Mutation*): Nature imparts mutation (introducing small changes in the genetic material) while creating offspring. GA mimics this operation by mutating elements of each of the $2 \times N_{\text{pop}}$ number of offspring strings randomly where the probability of mutation (P_{mut}) is kept small. During mutation, the top-ranking string in the parent population is excluded so as not to lose it. Next, each of the $2N$ chromosomes are evaluated using the fitness function and these are ranked beginning with the chromosome with the highest fitness score following which the lower half of the 2N-sized population is discarded. The resulting population of size N_{pop} forms the new generation (population) of candidate solutions and thus the generation index is increased by unity.

Step 6: Repeat steps 3-5 until convergence is achieved. Convergence is reached when one of the following two conditions are met: (a) GA has evolved over the pre-specified number of generations, and (b) the fitness score of the best solution in the population no longer increases over successive generations. The solution possessing highest fitness score forms the best solution of the particular GA run. Multiple runs with systematic variation in the GA parameters such as crossover and mutation probabilities, and length of the chromosomes, need to be conducted for obtaining an overall optimal solution. Since the converged solution also depends upon the initial randomly generated population of candidate solutions, this set also needs to be initialized differently by varying the random number generator seed multiple times.

A flowchart of the genetic algorithm operation is depicted in **Figure 2.7**. It may be noted that being stochastic procedures a number of variants to the basic GA presented here have been proposed. These variants yield better performance in specific cases.

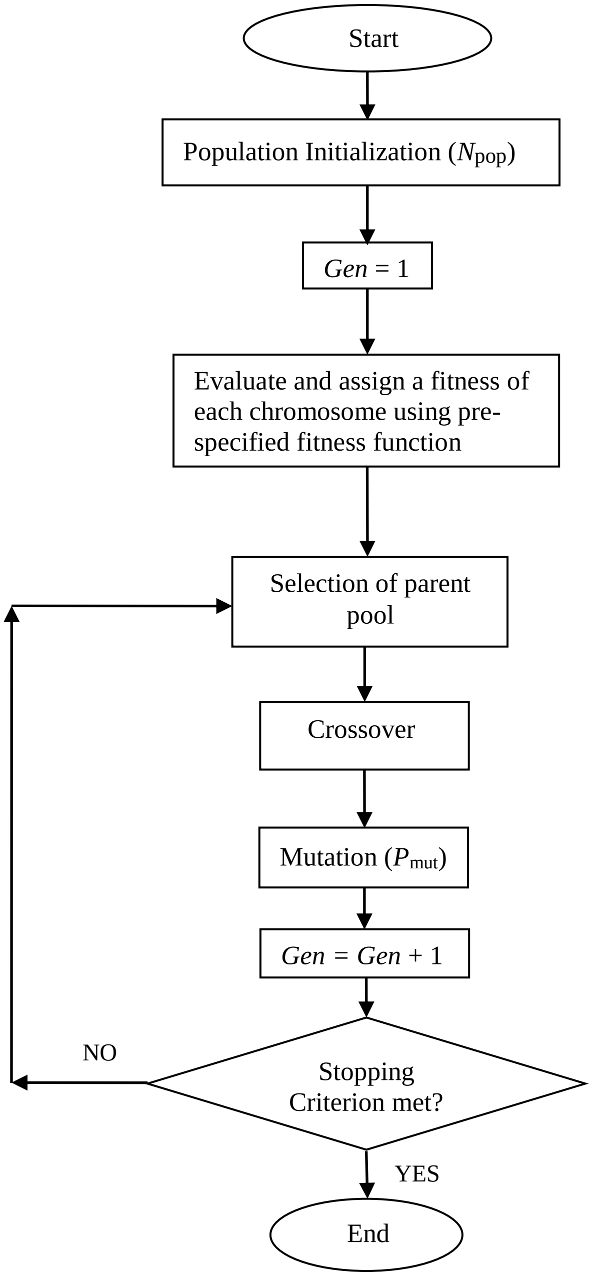

Start
Population Initialization (N_{pop})
Gen = 1
Evaluate and assign a fitness of each chromosome using pre-specified fitness function
Selection of parent pool
Crossover
Mutation (P_{mut})
Gen = Gen + 1
Stopping Criterion met?
NO
YES
End

Figure 2.7: Flow diagram representing genetic algorithm operations

Table 2.5 presents some representative applications of the GA method in chemical/ biochemical engineering.

Table 2.5: Selected representative studies of GA applications in chemical/ biochemical engineering

SN	GA Application	Study	Reference
1	Steady-state and dynamic process modeling/prediction	• Mathematical modeling of continuous ethanol fermentation in a membrane bioreactor by pervaporation compared to conventional system: Genetic algorithm	Esfahanian et al. (2016)
		• Modeling and optimization using artificial intelligence strategies	Hamid et al (2014)
		• Adaptive genetic programming for steady-state process modeling	Grosman and Lewin (2004).
2	Process Monitoring	• Automatic generation of interlock designs using genetic algorithms	Lepar et al. (2017)
		• Genetic algorithms for feature selection of image analysis-based quality monitoring model: An application to an iron mine	Chatterjee and Bhattacherjee (2011)
3	Model based nonlinear process control	• Filtered predictive control design using multiobjective optimization based on	Araujo and Coelho (2017)

		genetic algorithm for handling offset in chemical processes	
		- A genetic algorithm-based approach to intelligent modeling and control of pH in reactors	Mwembeshi et al. (2004)
		- Solving a nonlinear non-convex trim loss problem with a genetic hybrid algorithm	Ostermark (1999)
		- Application of nonlinear multivariable model predictive control to transient operation of a gas turbine and NO_X emissions reduction	Pires et al. (2018)
4	Process Identification	- Genetic algorithm for fuel spill identification	Lavine et al. (2001)
		- Discrete-time parameter estimation with genetic algorithms	Das and Goldberg (1988)
		- Genetic algorithms combined with discriminant analysis for key variable identification	Chiang and Pell (2004)

5	Process optimization	• Experimental investigation, modeling and optimization of membrane separation using artificial neural network and multi-objective optimization using genetic algorithm	Soleimani et al. (2013)
		• A hybrid neural-genetic multi-model parameter estimation algorithm	Petridis (1998)
		• A comparison of three diversity indices based on their components of richness and evenness	DeJong (1975)
		• The removal of arsenite [As (III)] and arsenate [As(V)] ions from wastewater using TFA and TAFA resins: Computational intelligence-based reaction modeling and optimization	Patil-Shinde et al. (2016)
		• Simultaneous optimization of chemical flow-shop sequencing and topology using genetic algorithms	Cartwright and Long (1993)
		• Optimization of	Desai et al. (2006)

		fermentation media for exopolysaccharide production from *Lactobacillus plantarum* using artificial intelligence-based techniques	
6	Fault detection and diagnosis (FDD)	• Fault prediction based on dynamic model and grey time series model in chemical processes	Tian et al . (2014)
		• Optimal features selection for designing a fault diagnosis system	Ardakani et al. (2016)
7	Quantitative structure-activity/ property relationships (QSAR/QSPR)	• Application of quantitative structure-property relationship analysis to estimate the vapor pressure of pesticides	Goodarzi et al. (2016)
		• Modeling physico-chemical properties of (benzo) triazoles, and screening for environmental partitioning	Bhhatarai and Gramatica (2011)
8	Soft-sensor development	• Improving the efficiency of dissolved oxygen control using an on-line control system based on a genetic algorithm evolving FWNN	Ruan et al. (2017)

		software sensor	
		- A comparative study of three evolutionary algorithms for surface acoustic wave sensor wavelength selection	Li and Heinemann (2007)

2.6 REDUCTION OF DIMENSIONALITY OF THE PREDICTOR SPACE

In a typical chemical process operation, very large amount of data comprising values of the operating and performance variables, and raw material and product quality attributes/analyses are routinely generated and archived. These data inherently contain instrumental and/or measurement noise. In many instances process data are also correlated linearly or nonlinearly, which leads to redundancy. Thus, it becomes necessary to preprocess the data to reduce the noise (if not eliminate it completely) and perform the analysis of correlations so that the correlated variables could be ignored leading to the reduction in the dimensionality of the data. The advantage of dimensionality reduction is that (a) it helps in constructing parsimonious (less complex) models possessing a good generalization capability, and (b) it reduces the computational effort and time involved in developing the model. A widely used linear dimensionality reduction technique is known as *Principal Component Analysis* (PCA). The techniques used for performing nonlinear dimensionality are, for example, auto-associative and Sammon's neural networks (Lerner et al., 1999). In the studies presented in this thesis, linear PCA has been used for reducing the dimensionality of the predictor (input) space of the CI-based models. In what follows, the PCA method is described in brief. An in-depth description of the PCA is provided for instance by Geladi and Kowalski (1986), Iyer et al. (2003), Abdi and Williams (2010), and Sharma et al. (2014).

2.6.1 Principal Component Analysis

Principal Component Analysis (PCA) was introduced in the year 1901, by a statistician, Karl Pearson, while proposing a formulation (Pearson, 1901) for determining the closest fitting lines and planes for a system of points in a specified space. It was subsequently established that PCA is more appropriate for the analysis of variance for the modeling of the response data (Fisher and MacKenzie, 1923). The concept of "percentage of variance captured" in the PCA was popularized by Hotelling (1933). PCA (Geladi and Kowalski, 1986) is the most widely used linear dimensionality reduction technique that extracts linear relationships existing among the variables of a dataset. It is a multivariate, statistical technique that can be used to examine data variability and frequently applied to the datasets that are large, difficult to interpret, and where complex inter-relationships between variables are difficult to identify and visualize (Kozub and MacGregor, 1992).

In PCA, the new variables derived, called *Principal Components* (PCs) or *factors*, are computed from the linear combinations of the original variables. The first principal component accounts for the highest variability in the data; each succeeding PC accounts for as much of the still outstanding variability as possible (see Figure 2.8). Thus, in theory, it is possible to have an infinite number of PCs with each accounting for less data variability than the previous one (Dong and McAvoy, 1996). Choosing, the first few PCs, which are lesser in number than the number of original data variables and accounting for a large portion of the data variability (say 95% or more) to represent the original data set allows compression of the information content and consequently dimensionality reduction.

Procedurally, PCA begins by finding the eigenvalues and eigenvectors of the sample covariance or correlation matrix. The eigen value essentially represents the standard sample variance of the projected data points. The eigenvectors corresponding to the largest eigenvalues (PCs) are used to reconstruct a large fraction of the variance of the original data. Thus, PCA generates PCs as a set of “pseudo-variables” or “latent variables,” which are linearly independent (uncorrelated) orthogonal variables (Nomikos and MacGregor, 1994). PCA determines the linear mapping between the variables in the original dataset in a low dimensional space in a manner such that the variance of the data in the low dimensional representation is maximized. This is done by decomposing the original dataset comprising the linearly correlated variables into a PCA transformed variable set defining the eigenvectors of the covariance of the data.

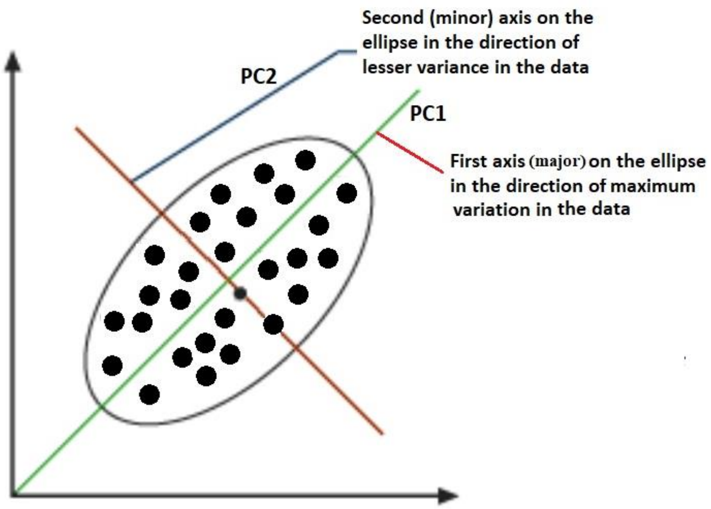

Second (minor) axis on the ellipse in the direction of lesser variance in the data
PC2
PC1
First axis (major) on the ellipse in the direction of maximum variation in the data

Figure 2.8: An illustration of principal component analysis. Given a data bounded in an ellipse, PCA derives two principal components, one in the direction of maximum variation in the data and the second in the direction of lesser variation.

To illustrate the principles of the PCA method, consider a two-dimensional matrix, $S(I, J)$, consisting of I measurements of J variables. The PCA decomposes this matrix, into matrices of latent variables and the corresponding *parameters* or *loadings* as given by,

$$S = TP' + E \qquad (2.25)$$

where, S is a mean-centered (mean = 0) matrix and the variance is scaled to unity. This indicates that the standard deviation of each element of every column is unity; $T(I,J)$ refers to the matrix of J principal component *scores*; i.e. each of the column matrix of T represents a PC; P'denotes the transpose of the *loading* matrix, $P(J,J)$, and E refers to the residuals. The first M principal component scores account for a large amount of variance in the dataset, and, therefore, the original matrix S can also be written as

$$S = \sum_{k=1}^{M} t_k \, (P_k)' + E' \qquad (2.26)$$

where, t_k denotes an I-dimensional k^{th} score vector, P_k refers to the transpose of the k^{th} J-dimensional loading vector, P_k, and E' is the residual matrix. It is thus witnessed that the original S (I, J) data matrix is now represented in terms of a smaller matrix. The sum of squares of elements of the score vector t_k is related to the Eigen value of that vector (known as 'trace'), which is a measure of the variance captured by the k^{th} PC. It also shows that a larger Eigen value is indicative of more significant respective PC.

Principal component analysis has proven to be an excellent method for feature extraction and data reduction in large datasets and has been used in a variety of chemical/ biochemical process engineering tasks as listed in **Table 2.6**.

Table 2.6: Recent chemical/ biochemical engineering applications of principal component analysis (PCA)

SN	PCA Application	Study	Reference
1	Process Monitoring	• Local and global principal component analysis for process monitoring	Yu (2012)
		• Modeling and monitoring of batch processes using principal component analysis	Kulkarni et al. (2004)

		(PCA) assisted generalized regression neural networks (GRNN)	
		• Statistical analysis of the physico–chemical data on the coastal waters of Cochin	Iyer et al (2003)
2	Process Identification	• Experimental determination by principal component analysis of a reaction pathway of biohydrogen production by anaerobic fermentation	Lara et al. (2008)
		• Descriptive and multivariable analysis of the physico-chemical and biological parameters of Sfax wastewater treatment plant	Ouali et al. (2009)
		• Analyzing adsorption data of erythrosine dye using principal component analysis	Degs et al. (2012)
3	Fault detection and diagnosis (FDD)	• Fault detection of batch processes using multiway kernel principal component analysis	Lee et al. (2004)
		• Dynamic model-based fault diagnosis for (bio)chemical batch processes	Kerkhof et al. (2012)
4	Quantitative structure-activity/ property	• Highly correlating distance/ connectivity-based topological indices: 5.	Shamsipur et al. (2008).

	relationships (QSAR/QSPR)	Accurate prediction of liquid density of organic molecules using PCR and PC-ANN	
		• Development of QSPR model relating solvent structure to crystal morphology	Haser et al. (2014).
5	Soft-sensor development	• Double locally weighted principal component regression for soft sensor with sample selection under supervised latent structure	Yuan et al. (2016).
		• Developing a soft sensor for fineness in a cement ball mill	Kavitha et al. (2014)
		• Data-driven Soft Sensors in the process industry	Kadlec et al. (2009)

2.7 STATISTICAL METHOD FOR MODEL DISCRIMINATION

In the studies presented in this thesis, several reaction/process models were developed using CI-based strategies, namely ANN, SVR and GP. Thus, it became essential to compare their prediction accuracy and generalization performance and choose the best model. At primary level, such a comparison was done using statistical measures, namely coefficient of correlation (*CC*), root mean square error (*RMSE*) and mean absolute percentage error (*MAPE*). In addition, an advanced test termed Steiger's z-test (Steiger, 1980) was performed for comparing rigorously the prediction accuracy and generalization performance of the multiple competing CI-based models. The task of the recognition of the best model in terms of a rigorous statistical comparison between the correlation coefficients of a pair of models is facilitated by this test.

2.7.1 Steiger's z - test

This test examines whether the two *CC*s corresponding to the predictions of a pair of models are statistically significantly different or equivalent. It tests the null hypothesis (H_0) that two *CC* magnitudes are not statistically different, that is, $CC_{AB} = CC_{BC}$, where subscripts *A*, *B*, and *C*, respectively designate the desired (target) output of the model ({y} values) and those predicted by the models *J* and *K*. To afford the above-stated comparison, initially the z-scores are computed using Fisher's r-to-z transformation (Fisher, 1921) as follows:

$$z_{ij} = \frac{1}{2} \ln \left[\frac{1+ CC_{ij}}{1- CC_{ij}}\right] \tag{2.27}$$

where CC_{ij} denotes the correlation coefficient between the readings 'i' and 'j'. The covariance between two predictions y_B and y_C , denoted by h is expressed as

$$h = cov\ (CC_{AB}, CC_{AC}) \tag{2.28}$$

If the Fisher's z-transformations of both correlation coefficients (CC_{AB}, CC_{AC}) are expressed as z_{AB}, and z_{AC} respectively, then Fisher's z-transformation value is expressed as

$$z = \frac{(z_{AB} - z_{AC})\sqrt{N_R - 3}}{\sqrt{2(1-h\ CC_{BC})}} \tag{2.29}$$

Here, N_R is the total number of readings and CC_{BC} is the correlation coefficient between the model predicted values of models *B* and *C*. If the normal distribution value of 'y' is defined as $'normdist'$ and its absolute value be $abs[z]$ then the population correlation matrix correlating with Fisher's z-transform is expressed as

$$p = 2\big(1 - normdist(abs[z])\big) \tag{2.30}$$

The significance of population correlation matrix value or p-value indicates acceptance or rejection of the null hypothesis (H_0). If the p magnitude is less than 0.05, then the null hypothesis is rejected (at 95 % confidence level), meaning that the models being compared are considerably different. On the other hand, $p >$ 0.05 indicates acceptance of the null hypothesis thus meaning that the models under investigation are comparable and distinct. Consequently, it is possible to choose a model possessing higher prediction accuracy and generalization capability.

2.8 CONCLUSION

Conventionally, modeling of chemical, biochemical, polymer and other reactions/processes is performed using the *phenomenological* approach, which possesses certain drawbacks. Its alternative, namely *empirical* modeling implemented using a variety of linear and nonlinear regression methods also suffers from deficiencies. In the last three decades computational intelligence (CI) has provided multiple modeling approaches possessing certain advantages over the phenomenological and empirical modeling approaches. This thesis presents modeling studies wherein three CI-based methods namely multilayer perceptron neural networks (MLPNN), support vector regression (SVR) and genetic programming (GP) have been extensively used in the modeling of chemical and biochemical systems. Accordingly, this Chapter presents an overview of the stated three CI-based formalisms. Specifically, the Chapter covers their origin, operating principles, implementation procedure, and various applications in chemical engineering and technology. The major applications of MLPNN, SVR and GP include modeling, control, monitoring, identification, soft-sensor development, quantitative structure –activity/property relationships and fault detection and diagnosis.

Similar to the conventional modeling, the standard methods of process optimization such as the deterministic gradient based approaches suffer from certain serious deficiencies. A number of computational/artificial intelligence based stochastic optimization techniques have been proposed in the last three decades that overcome the deficiencies of the deterministic gradient based optimization methods. In the present thesis, a widely used stochastic optimization method namely genetic algorithms (GA) that belongs to the artificial intelligence based evolutionary algorithms, has been employed for optimizing reaction conditions of a photocatalytic degradation of pharmaceutical pollutants (see Chapter 3). Accordingly, an overview of GA method encompassing its origin, salient features, implementation procedure and chemical and biochemical applications, has been presented in this chapter.

Invariably, process data contain variables that are correlated. Their presence results in enhanced computational load during modeling and redundancy. A widely utilized method for identifying linearly correlated variables in the data is known as principal component analysis (PCA). This method defines new variables based on the original data that are not linearly correlated. This ultimately helps in reducing the dimensionality of the dataset. In the present chapter, PCA has been presented for reducing the dimensionality of the input (predictor) space of the CI-based models (see Chapter 3, 5 and 6). Accordingly, a detailed description of PCA along with its applications has been provided in this chapter.

In the various modeling studies presented in the thesis, it was necessary to compare the prediction accuracy and generalization capability possessed by the MLPNN, SVR and GP based models and choose the best one. In addition to the commonly used measures such as correlation coefficient (*CC*), and root mean square error (*RMSE*), an advanced statistical test known as Steiger's z-test has been utilized for comparing the performances of the CI-based models. Accordingly, the stated test has been discussed elaborately in this thesis.

To summarize, this chapter lays down a strong theoretical and practical foundation by presenting a comprehensive overview of the CI-based modeling and optimization as also model selection methods employed in the research studies presented in the subsequent chapters.

2.9 REFERENCES

Abbasi A., Eslamloueyan R. (2014). Determination of binary diffusion coefficients of hydrocarbon mixtures using MLP and ANFIS networks based on QSPR method, *Chemometrics and Intelligent Laboratory Systems*, **132**, pp. 39-51.

Abdi H., Williams L.J. (2010). Principal component analysis, *Wiley Interdisciplinary Rev. Comput. Stat.*, **2(4)**, pp. 433–459.

Abdi K. M., Bemani A., Naserzadeh Z., Zhang Z. (2018). Prediction of solubility of N-alkanes in supercritical CO_2 using RBF-ANN and MLP-ANN, *Journal of CO_2 Utilization*, **25**, pp. 108-119.

Aluffi-Pentini F., Parisi V., Zirilli F. (1985). Global Optimization and Stochastic Differential Equations, *Journal of Optimization Theory and Applications*, **47 (1)**, pp.1–16.

Arabloo M., Bahadori A., Ghiasi M., Lee M.A., Zendehboudi S. (2015). A novel modeling approach to optimize oxygen–steam ratios in coal gasification process, *Fuel*, **153**, pp.1-5.

Araujo R.B., Coelho A.R. (2017). Filtered predictive control design using multi-objective optimization based on genetic algorithm for handling offset in chemical processes, *Chemical Engineering Research and Design*, **117**, pp. 265-273.

Ardakani M., Askarian M., Shokry A., Escudero G., Graells M., Espuña A. (2016). Optimal features selection for designing a fault diagnosis system, *Computer Aided Chemical Engineering*, **38**, pp. 1111-1116.

Averett R.D., Realff M.L. Jacob K.L. (2010). Comparative post fatigue residual property predictions of reinforced and unreinforced poly (ethylene terephthalate) fibers using artificial neural networks, *Composites Part A: Applied Science and Manufacturing*, **41(3)**, pp.331-344.

Azadeh A., Ghaderi S. F., Sohrabkhani S. (2008). Annual electricity consumption forecasting by neural network in high energy consuming industrial sectors, *Energy Conversion and Management*, **49(8)**, pp. 2272-2278.

Bäck T., Schwefel H. P. (1993). An overview of evolutionary algorithms for parameter optimization, *Evolutionary computation*, **1(1)**, pp.1-23.

Badhe Y., Bhat R., Borate P. (2014). Optimization Techniques for Improving the

Performance of Information Retrieval System, *International Journal of Research in Advent Technology*, **2(2)**, pp. 263-267. E-ISSN: 2321-9637.

Bai Y., Li C. (2016). Daily natural gas consumption forecasting based on a structure-calibrated support vector regression approach, *Energy and Buildings*, **127**, pp. 571-579.

Balcilar M., Aroonrat K., Dalkilic A.S., Wongwises S. (2013). A numerical correlation development study for the determination of Nusselt numbers during boiling and condensation of R134a inside smooth and corrugated tubes, *International Communications in Heat and Mass Transfer*, **48**, pp. 141-148.

Bahrami P., Kazemi P., Mahdavi S., Ghobadi H. (2016). A novel approach for modeling and optimization of surfactant/ polymer flooding based on Genetic Programming evolutionary algorithm, *Fuel*, **179**, pp.289-298.

Bhhatarai B., Gramatica P. (2011). Modelling physico-chemical properties of (benzo) triazoles, and screening for environmental partitioning, *Water Research*, **45(3)**, pp.1463-1471.

Biegler L.T. (2010). *Introduction to Process Optimization in Nonlinear Programming*, MOS-SIAM Series on Optimization, pages 1-16, Society for Industrial and Applied Mathematics.

Bishop C.M. (1995). *Neural networks for pattern recognition,* Oxford University Press, Oxford.

Blaesi J., Jensen B. (1992). Can neural network compete with process calculations? *INTECH*, **39(12)**, pp. 34-37.

Bouamama B.O., Harabi R.E., Abdelkrim M.N., Gayed M.K. (2012). Bond graphs for the diagnosis of chemical processes, *Computers & Chemical Engineering*, **36**, pp.301-324.

Bouhouche S., Yazid L.L., Hocine S., Bast J. (2010). Evaluation using online support-vector-machines and fuzzy reasoning; Application to condition monitoring of speeds rolling process, *Control Engineering Practice*, **18(9)**, pp.1060-1068.

Bos M., Bos A., Van der Linden W. (1993). Data processing by neural networks in quantitative chemical analysis, *Analyst*, **118**, pp. 323-328.

Broten G. S., Wood H. C. (1993). A neural network approach to analyzing multi-component mixtures, *Measurement Science and Technology*, **4(10)**, page 957.

Buchacz A., Zolkiewski S. (2007). Dynamic analysis of the mechanical systems vibrating transversally in transportation, *Journal of Achievements in Materials and Manufacturing Engineering*, **20**, pp. 331-334.

Buyukada M., Aydogmus E.(2018). Utilization of apricot seed in (co-) combustion of lignite coal blends: Numeric optimization, empirical modeling and uncertainty estimation, *Fuel*, **216**, pp.190-198.

Canete J.F., Saz O.P.D., Baratti R., Mulas M., Ruano A., Garcia-Cerezo A. (2016). Soft-sensing estimation of plant effluent concentrations in a biological wastewater treatment plant using an optimal neural network, *Expert Systems with Applications*, **63**, pp. 8-19.

Carberry J.J. (2001). *Chemical and Catalytic Reaction Engineering*, Dover Publications Inc., pages 623.

Cartwright H., Long R. (1993). Simultaneous optimization of chemical flow-shop sequencing and topology using genetic algorithms, *Ind. Eng. Chem. Res.*, **32 (11)**, pp 2706–2713.

Chan L.T., Chen T., Chen J. (2016). PID based nonlinear processes control model uncertainty improvement by using Gaussian process model, *Journal of Process Control*, **42**, pp. 77-89.

Chatterjee S., Bhattacherjee A. (2011). Genetic algorithms for feature selection of image analysis-based quality monitoring model: An application to an iron mine, *Engineering Applications of Artificial Intelligence*, **24(5)**, pp. 786-795.

Chetouani Y. (2014). Model selection and fault detection approach based on Bayes decision theory: Application to changes detection problem in a distillation column, *Process Safety and Environmental Protection*, **92(3)**, pp. 215-223.

Cheema J., Sankpal N., Tambe S.S., Kulkarni B.D. (2002). Genetic programming assisted stochastic optimization strategies for optimization of glucose to gluconic acid fermentation, *Biotechnology Progress*, **18**, pp. 356−1365.

Chiang L.H., Pell R.J. (2004). Genetic algorithms combined with discriminant analysis for key variable identification, *Journal of Process Control*, **14(2)**, pp.143-155.

Chine W., Mellit A., Lughi V., Malek A., Sulligoi G., Pavan A.M. (2016). A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks, *Renewable Energy,* **90**, pp. 501-512.

Chitra S. (1993). Use neural networks for problem solving, *Chemical Engineering Process*, **89**, pp. 44-52.

Chou J.S., Ngo N.T., Chong W.K. (2017). The use of artificial intelligence combiners for modeling steel pitting risk and corrosion rate, *Engineering Applications of Artificial Intelligence,* **65**, pp. 471-483.

Das R., Goldberg D. (1988). Discrete-time parameter estimation with genetic algorithms, *19th Annual Pittsburgh Conference on Modeling and Simulation*, **19**, pp. 2391-2395.

Dassau E., Grosman B., Lewin D.R. (2006). Modeling and temperature control of rapid thermal processing, *Computers & Chemical Engineering*, **30(4)**, pp.686-697.

De Castro L. N., Von Zuben F. J. (2002). Learning and optimization using the clonal selection principle, *IEEE transactions on evolutionary computation*, **6(3)**, pp.239-251.

Degs Y.S., Halawa R.A., Alrubb S.S. (2012). Analyzing adsorption data of erythrosine dye using principal component analysis, *Chemical Engineering Journal*, **191**, pp.185-194.

Desai K.M., Akolkar S.K., Badhe Y.P., Tambe S.S., Lele S.S. (2006). Optimization of fermentation media for exopolysaccharide production from Lactobacillus plantarum using artificial intelligence-based techniques, *Process Biochemistry*, **41**, pp. 1842–1848.

Desai K.M., Badhe Y.P., Tambe S.S., Kulkarni B.D. (2006). Soft-sensor development for fed-batch bioreactors using support vector regression, *Biochemical Engineering Journal*, **27**, pp.225–239.

DeJong T. (1975). A comparison of three diversity indices based on their components of richness and evenness, *OIKOS*, **26**, pp. 222-227.

De Jong K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems (Doctoral dissertation, University of Michigan), *Dissertation Abstracts International*, **36(10)**, 5140B. (University Microfilms No. 769381)

Dong D., McAvoy T. J. (1996). Nonlinear Principal Component Analysis Based on Principal Curves and Neural Networks, *Comp. & Chem. Engg*, **20**, pp. 65 – 78.

Dong F., Hou L., Jiang L., Cao T., Nia J. (2017). Simulation of sub-cooled flow boiling with an SVR based interphase mass transfer model, *Applied Thermal Engineering*, **116**, pp. 840-849.

Dutta B.K. (2017). *Mathematical Methods in Chemical and Biological Engineering*, CRC Press, Taylor & Francis Group, pages 685.

Edgar T. F., Himmelblau D. M., Lasdon L. S. (2001). Optimization of chemical processes, McGraw-Hill, New York.

Emamgholizadeh S., Parsaeian M., Baradaran M. (2015). Seed yield prediction of sesame using artificial neural network, *European Journal of Agronomy*, **68**, pp.89-96.

Engell S., Dadhe K. (2001). Neural Networks for Modeling and Control of Reactive Distillation, *IFAC Proceedings Volumes*, **34(22)**, pp. 354-359.

Esfahanian M., Rad A.S., Khoshhal S.,Najafpour G., Asghari B. (2016). Mathematical modeling of continuous ethanol fermentation in a membrane bioreactor by pervaporation compared to conventional system: Genetic algorithm, *Bioresource Technology*, **212**, pp. 62-71.

Esteki M., Rezayat M., Ghaziaskar H.S., Khayamian T. (2005). Application of QSPR for prediction of percent conversion of esterification reactions in supercritical carbon dioxide using least squares support vector regression, *The Journal of Supercritical Fluids*, **54(2)**, pp. 222-230.

Fisher R. A. (1921). On the probable error of a coefficient of correlation deduced from a small sample, *Metron*, **1**, pp. 1-32.

Fisher R. A., Mackenzie W. A. (1923). Studies in crop variation. II. The manurial response of different potato varieties, *The Journal of Agricultural Science*, **13(3)**, pp.311-320.

Freeman J.A (1999). Neural Network development and deployment rules-of-thumb, *Hydrocarbon Processing*, pp.101-107.

Gabbar H.A., Hussain S., Hosseini A.H. (2014). Simulation-based fault propagation analysis—Application on hydrogen production plant, *Process Safety and Environmental Protection*, **92(6)**, pp.723-731.

Gandhi A.B., Joshi J.B., Jayaraman V.K., Kulkarni B.D. (2007). Development of support vector regression (SVR)-based correlation for prediction of overall gas hold-up in bubble column reactors for various gas–liquid systems, *Chemical Engineering Science*, **62(24)**, pp.7078-7089.

Gandhi A.B., Joshi J.B., Kulkarni A., Jayaraman V., Kulkarni B.D. (2008). SVR-based prediction of point gas hold-up for bubble column reactor through recurrence quantification analysis of LDA time-series, *International Journal of Multiphase Flow*, **34(12)**, pp. 1099-1107.

Geladi P., Kowalski B.R. (1986). Partial least-squares regression: a tutorial, *Analytica Chimica Acta*, **185**, pp.1–17.

Ghugare S. B., Tiwary S., Elangovan V., Tambe S. S (2014). Prediction of Higher Heating Value of Solid Biomass Fuels using Artificial Intelligence

Formalisms, *Bioenergy Research*, **7**, pp.681–692. DOI: 10.1007/s12155-013-9393-5.

Ghugare S.B., Tambe S.S. (2016). Development of genetic programming based soft-sensor model for styrene polymerization process and its application in model-based control, *Indian Control Conference (ICC)*, pp. 238-244, IIT Hyderabad, India.

Ghugare S.B., Tambe S.S. (2017). Genetic programming based high performing correlations for prediction of higher heating value of coals of different ranks and from diverse geographies, *Journal of the Energy Institute,* **90(3)**, pp. 476-484.

Ghugare S.B., Tiwary S., Tambe S.S. (2017). Computational intelligence based models for prediction of elemental composition of solid biomass fuels from proximate analysis, *International Journal of System Assurance Engineering and Management*, 8**(4)**, pp. 2083–2096.

Goel P., Bapat S., Vyas R., Tambe A., Tambe S.S. (2015). Genetic programming based quantitative structure–retention relationships for the prediction of Kovats retention indices, *Journal of ChromatographyA*, 1420, pp.98-109. DOI: 10.1016/j.chroma.2015. 09.086.

Goldberg D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*, Reading: Addison-Wesley, pp. 343-349.

Golkarnarenji G., Naebe M., Badii K., Milani A.S., Jazar R.N., Khayyam H. (2018). Support vector regression modelling and optimization of energy consumption in carbon fiber production-line, *Computers & Chemical Engineering*, **109**, pp. 276-288.

Golzar K., Amjad-Iranagh S., Modarress H.(2013). QSPR prediction of the solubility of CO_2 and N_2 in common polymers, *Measurement*, **46(10)**, pp.4206-4225.

Goodacre R., Douglas B. Kell D.B. (1996). Correction of mass spectral drift using artificial neural networks, *Analytical Chemistry*, **68(2)**, pp.271-280.

Goodacre R., Gilbert R.J. (1999). The detection of caffeine in a variety of beverages using Curiepoint pyrolysis mass spectrometry and genetic programming, *The Analyst*, **124**, pp.1069-1074.

Goodarzi M., Coelho L.S., Honarparvar B., Ortiz E.V., Duchowicz P.R. (2016). Application of quantitative structure-property relationship analysis to estimate the vapor pressure of pesticides, *Ecotoxicology and Environmental Safety*, **128**, pp.52-60.

Gopakumar V., Tiwari S., Rahman I. (2018). A deep learning based data driven soft sensor for bioprocesses, *Biochemical Engineering Journal*, **136**, pp.28-39.

Grosman B., Lewin D.R. (2002). Automated nonlinear model predictive control using genetic programming, *Computers & Chemical Engineering*, **26(4-5)**, pp. 631-640.

Grosman B., Lewin D.R. (2004). Adaptive genetic programming for steady-state process modeling, *Computers & Chemical Engineering*, **28(12)**, pp. 2779-2790.

Guo H., Jack L.B., Nandi A.K. (2005). Feature generation using genetic programming with application to fault classification, *IEEE on systems*, **35(1)**, pp.89-99.

Hamid A., Deshpande A.S., Badhe Y.P., Barve P.P., Tambe S.S., Kulkarni B.D. (2014). Biodegradable iron chelates for H_2S abatement: Modeling and optimization using artificial intelligence strategies, *Chemical Engineering Research and Design*, **92**, pp. 1119–1132.

Harooni A.B., Marghmaleki A.N., Mohammadi A.H. (2017). Prediction of heat capacities of ionic liquids using chemical structure based networks, *Journal of Molecular Liquids*, **227**, pp. 324-332.

Haser J.C., Herring R.H., Datta S., Eden M.R. (2014). Development of QSPR model relating solvent structure to crystal morphology, *Computer Aided Chemical Engineering*, **34**, pp.321-326.

Hiden H., Willis M., Tham M., Turner P., Montague G. (1998). Non-linear principal components analysis using genetic programming, *IEE Conf. Publication*, **446**, pp.302-307.

Holland J. H. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.

Hotelling H. (1933). Analysis of a complex of statistical variables into principal components, *Journal of educational psychology*, **24(6)**, pp .417-441.

Ivanciuc O. (2007). Applications of support vector machines in chemistry, *Rev. Comput. Chem.*, **23**, pp. 291–400. DOI: 10.1002/9780470116449.ch6.

Iyer C.S., Sindhu M., Kulkarni S.G., Tambe S.S., Kulkarni B.D. (2003). Statistical analysis of the physico–chemical data on the coastal waters of Cochin, *Journal of Environmental Monitoring*, **5**, pp.324-327.

Jerne N. K. (1974). Towards a network theory of the immune system. Annual Review of Immunology, **125**, pp.373-389.

Kadlec P., Gabrys B., Strandt S. (2009). Data-driven Soft Sensors in the process industry, *Computers & Chemical Engineering*, **33(4)**, pp.795-814.

Kamal M.M., Yu D.W., Yu D.L. (2014). Fault detection and isolation for PEM fuel cell stack with independent RBF model, *Engineering Applications of Artificial Intelligence*, **28**, pp.52-63.

Kano M., Sakata T., Hasebe S. (2011). Just-In-Time Statistical Process Control: Adaptive Monitoring of Vinyl Acetate Monomer Process, *IFAC Proceedings Volumes*, **44(1)**, pp. 13157-13162.

Karakus M. (2011). Function identification for the intrinsic strength and elastic properties of granite rocks via genetic programming (GP), *Computer & Geosciences*, **37**, pp.1318-23.

Kavitha S.P., Guruprasath M., Tangirala A.K. (2014). Developing a soft sensor for fineness in a cement ball mill, *IFAC Proceedings Volumes*, **47(1)**, pp.1019-1025.

Kennedy J., Eberhart R. (1995). Particle Swarm Optimization, *Proceedings of ICNN'95 - International Conference on Neural Networks*, pp. 1942 -1948. DOI: 10.1109/ICNN.1995.488968

Kerkhof P.V., Gins G., Jan J.V., Vanlaer J., Impe J.V. (2012). Dynamic model-based fault diagnosis for (bio) chemical batch processes, *Computers & Chemical Engineering*, **450**, pp.12-21.

Khandelwal M, Armaghani D.J., Majid M.Z.A., Yagiz S.(2017). Function development for appraising brittleness of intact rocks using genetic programming and nonlinear multiple regression models, *Engineering with Computers*, **33(1)**, pp.13-21.

Khosravi A., Pabon J.J. G., Koury R.N.N., Machado L. (2018). Using machine learning algorithms to predict the pressure drop during evaporation of R407C, *Applied Thermal Engineering*, **133**, pp.361-370.

Koç D.I., Koç M.L. (2015). A genetic programming-based QSPR model for predicting solubility parameters of polymers, *Chemometrics and Intelligent Laboratory Systems*, **144**, pp.122-127.

Kordon A., Smits G., Jordaan E., Rightor E.D. (2002). Robust soft sensors based on integration of genetic programming, analytical neural networks, and support vector machines, *Proceedings of 2002 Congress on Evolutionary Computation*, **2**, pp. 896-901.

Kordon A., Jordaan E., Chew L., Smits G., Bruck T., Haney K., Jenings A. (2004). Biomass inferential sensor based on ensemble of models generated by genetic programming, *Genetic and Evolutionary Computation conference*, **2**, pp. 1078-1089.

Koza J.R. (1992). *Genetic Programming: On the programming of computers by means of natural selection*, A Bradford Book, The MIT Press, Cambridge, Massachusetts.

Kozub D. J., MacGregor J. F. (1992). State estimation for semi-batch polymerization reactors, *Chem. Engg. Sci.*, **47**, pp.1047 – 1062.

Kulkarni B.D., Tambe S.S., Dahule R.K., Yadavalli V.K. (1999). Consider genetic programming for process identification, *Hydrocarbon Processing*, **78(7)**, pp.89-97.

Kulkarni S.G., Chaudhary A.K., Nandi S., Tambe S.S., Kulkarni B.D. (2004). Modeling and monitoring of batch processes using principal component analysis (PCA) assisted generalized regression neural networks (GRNN), *Biochemical Engineering Journal*, **18(3)**, pp.193-210.

Langdon W.B., Poli R, McPhee N.F., Koza J.R. (2008). Genetic programming: An introduction and tutorial, with a survey of techniques and applications, *Computational Intelligence: A compendium*, Springer Berlin Heidelberg, **115**, pp.927-1028.

Lara C.A., Latrille E., Buffiere P., Bernet N., Steyer J.P.(2008). Experimental determination by principal component analysis of a reaction pathway of biohydrogen production by anaerobic fermentation, *Chemical Engineering and Processing: Process Intensification* , **47(11)**, pp. 1968-1975.

Lavine B.K., Brzozowski D., Moores A. J., Davidson C.E., Mayfield H.T. (2001). Genetic algorithm for fuel spill identification, *Analytica Chimica Acta*, **437(2)**, pp. 233-246.

Lee D.E., Song S.O., Yoon E. S. (2003). A Nonlinear Soft Sensor Based on Modified SVR for Quality Estimation in Polymerization, *IFAC Proceedings Volumes*, **36 (5)**, pp. 879-883.

Lee J.M., Yoo C.K., Lee I.B. (2004). Fault detection of batch processes using multiway kernel principal component analysis, *Computers & Chemical Engineering*, **28(9)**, pp.1837-1847.

Lee M., Chen J. (1993). Fluid Property Predictions with the Aid of Neural Networks, *Ind. Eng. Chem. Res.*, **32 (5)**, pp.995-997.

Lepar Y.Y., Wang Y.C., Chang C.T. (2017). Automatic generation of interlock designs using genetic algorithms, *Computers & Chemical Engineering*, **101**, pp.167-192.

Lerner B., Guterman H., Aladjem M. (1999). A comparative study of neural network-based feature extraction paradigms, *Pattern Recognition Letters*, **20(1)**, pp.7-14.

Levenberg K. (1944). A Method for the Solution of Certain Non-Linear Problems in Least Squares, *Quarterly of Applied Mathematics*, **2**, pp. 164–168.

Li C., Heinemann P.H.(2007). A comparative study of three evolutionary algorithms for surface acoustic wave sensor wavelength selection, *Sensors and Actuators B: Chemical*, **125(1)**, pp. 311-320.

Li Q., Chen W., Liu Z., Gio A., Huang J. (2014). Nonlinear multivariable modeling of locomotive proton exchange membrane fuel cell system, *International Journal of Hydrogen Energy*, **39(25)**, pp.13777-13786.

Liu Y., Li C., Gao Z. (2014). A novel unified correlation model using ensemble support vector regression for prediction of flooding velocity in randomly packed towers, *Journal of Industrial and Engineering Chemistry*, **20(3)**, pp.1109-1118.

Lightbody G., Reilly P.O., Irwin G.W., Kelly, McCormick J. (1997). Neural modeling of chemical plant using MLP and B-spline networks, *Control Engineering Practice*, **5(11)**, pp.1501-1515.

Marquardt, D. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters, *SIAM Journal on Applied Mathematics*, **11 (2)**, pp. 431–441.

McConaghy T., Vladislavleva E., Riolo R. (2010). *Genetic programming theory and practice 2010: An introduction*, Genetic Programming Theory and Practice VIII 8.

Meng Q., Xiaoping M., Zhou Y. (2014). Forecasting of coal seam gas content by using support vector regression based on particle swarm optimization, *Journal of Natural Gas Science and Engineering*, **21**, pp. 71-78.

Messikha N., Bousbaab S., Bougdah N. (2017). The use of a multilayer perceptron (MLP) for modelling the phenol removal by emulsion liquid membrane, *Journal of Environmental Chemical Engineering*, **5(4)**, pp. 3483-3489.

Moller M.F. (1993). A scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks*, **6(4)**, pp. 525-533.

Monroy I., Benitez R., Escudero G., Graells M. (2009). DICA enhanced SVM classification approach to fault diagnosis for chemical processes, *Computer Aided Chemical Engineering*, **26**, pp. 267-272.

Moscato P., Norman M.G. (1992). A memetic approach for the travelling salesman problem implementation of a computational ecology for the combinatorial optimization on message passing systems, *Parallel Computing and Transputer Applications*, **1**, pp. 177-186.

Mwembeshi M.M., Kent C.A., Salhi S. (2004). A genetic algorithm based approach to intelligent modelling and control of pH in reactors, *Computers & Chemical Engineering*, **28(9)**, pp.1743-1757.

Nadadoor V.R., Siegler H.D., Shah S.L., McCaffrey W.C., Benzvi A. (2012). Online sensor for monitoring a microalgal bioreactor system using support vector regression, *Chemometrics and Intelligent Laboratory Systems*, **110 (1)**, pp.38-48.

Naddeo A., Annarumma M., Pappalardo M., Cappetti N. (2008). State of the art of the passive pedestrian safety simulation, *Journal of Achievements in Materials and Manufacturing Engineering*, **30(1)**, pp.51-58.

Nandi S., Badhe Y.P., Lonari J., Sridevi U., Rao B.S., Tambe S.S., Kulkarni B.D. (2004). Hybrid process modeling and optimization strategies integrating neural networks/support vector regression and genetic algorithms: study of benzene isopropylation on H-beta catalyst, *Chemical Engineering Journal*, **97**, pp. 115–129.

Nandi S., Ghosh S., Tambe S.S., Kulkarni B.D. (2001). Artificial neural-network-assisted stochastic process optimization strategies, *AIChE Journal*, **47(1)**, pp. 126-141.

Nandi S., Rahman I., Tambe S.S., Sonolikar R.L., Kulkarni B.D. (2000). *Process identification using genetic programming: a case study involving fluidized catalytic cracking (FCC) unit*, Petroleum Refining and Petrochemical Based Industries In Eastern India, pp.195-201.

Nayak S. C., Misra B.B., Behera H.S. (2017). Artificial chemical reaction optimization of neural networks for efficient prediction of stock market indices, *Ain Shams Engineering Journal*, **8(3)**, pp. 371-390.

Niebling G. (1994). Identification of gases with classical pattern-recognition methods and artificial neural networks, *Sensors and Actuators*, **18-19**, pp. 259-263.

Nomikos P., MacGregor J.F. (1994). Monitoring batch processes using multiway principal component analysis, *AIChE Jour.*, **40**, pp.1361 – 1375.

Ostermark R. (1999). Solving a nonlinear non-convex trim loss problem with a genetic hybrid algorithm, *Computers & Operations Research*, **26(6)**, pp.623-635.

Ouali A., Azri C., Medhioub K., Ghrabi A. (2009). Descriptive and multivariable analysis of the physico-chemical and biological parameters of Sfax wastewater treatment plant, *Desalination*, **246(1-3)**, pp. 496-505.

Pandey D.S., Pan I., Das S., Leahy J.J., Kwapinski W. (2015). Multi-gene genetic programming based predictive models for municipal solid waste gasification in a fluidized bed gasifier, *Bioresource Technology*, **179**, pp. 524-533.

Parhizgar H., Dehghani M.R., Eftekhari A. (2013). Modeling of vaporization enthalpies of petroleum fractions and pure hydrocarbons using genetic programming, *Journal of Petroleum Science and Engineering*, **112**, pp. 97-104.

Patel S.U., Jeevankumar B., Badhe Y.P., Sharma B.K., Saha S., Biswas S., Chaudhury A., Tambe S.S., Kulkarni B.D. (2007). Estimation of gross calorific value of coals using artificial neural networks, *Fuel*, **86(3)**, pp. 334-344.

Patil-Shinde V., Mulani K.B., Donde K., Chavan N.N., Ponrathnam S., Tambe S.S. (2016). The removal of arsenite [As (III)] and arsenate [As (V)] ions from wastewater using TFA and TAFA resins: Computational intelligence based reaction modeling and optimization, *Journal of Environmental Chemical Engineering*, **4(4) Part-A**, pp.4275-4286.

Patil-Shinde V., Kulkarni T., Kulkarni R., Chavan P.D., Sharma T., Sharma B.K., Tambe S.S., Kulkarni B.D. (2014). Artificial intelligence-based modeling of high ash coal gasification in a pilot plant scale fluidized bed gasifier, *Ind. Eng. Chem. Res.*, **53 (49)**, pp.18678–18689.

Patil-Shinde V., Saha S., Sharma B., Tambe S.S., Kulkarni B.D. (2016). High ash char gasification in thermo-gravimetric analyzer and prediction of gasification performance parameters using computational intelligence formalisms, *Chemical Engineering Communications*, **203(8)**, pp.1029-1044. DOI: https://doi.org/10.1080/00986445.2015.1135795

Patil-Shinde V., Tambe S.S. (2018). Genetic programming-based models for prediction of vapor-liquid equilibrium, *Calphad*, **60**, pp. 68–80.

Patricia K., Esquerre O., Dale E., Seborg D.E., Mori M., Bruns R.E. (2004). Application of steady-state and dynamic modeling for the prediction of the BOD of an aerated lagoon at a pulp and paper mill: Part II. Nonlinear approaches, *Chemical Engineering Journal*, **105(1-2)**, pp. 61-69.

Petridis V. (1998). A hybrid neural-genetic multi-model parameter estimation algorithm, *IEEE Transactions on Neural Networks*, **9(5)**, pp.862-876.

Pearson K. (1901). *LIII. On lines and planes of closest fit to systems of points in space*, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2**(11)**, pp.559-572.

Pena B., Teruel E., Diez L.I. (2011). Soft-computing models for soot-blowing optimization in coal-fired utility boilers, *Applied Soft Computing* , **11(2)**, pp.1657-1668.

Perest J., Oliveiraz R., Feyo de Azevedo S. (2004). Hybrid Modeling of Fermentation Processes: A Study on the use of Modular Neural Networks for Modeling Cells Reaction Kinetics, *Computer Applications in Biotechnology*, IFAC Proceedings Elsevier, pp. 293-297.

Pires T.S., Cruz M.E., Colaço M.J., Alves M.A. (2018). Application of nonlinear multivariable model predictive control to transient operation of a gas turbine and NO_X emissions reduction, *Energy*, **149**, pp. 341-353.

Poli R., Langdon W., Mcphee N. (2008). *A Field Guide to Genetic Programming*. http://www0.cs.ucl.ac.uk/staff/wlangdon/ftp/papers/poli08_fieldguide.pdf. (Accessed on January 24, 2018).

Ramasamy S., Deshpande P.B., Tambe S. S., Kulkarni B.D. (1995). Robust nonlinear control with neural networks, *Proceedings of the Royal Society A (Mathematical, Physical and Engineering Sciences)*, **449 (1937)**, pp. 665-668.

Rechenberg I. (1989). Evolution strategy: nature's way of optimization, *Optimization: Methods and Applications, Possibilities and Limitations*, **47**, pp. 106- 126.

Rostami A., Ebadi H., Arabloo M., Meybodi M.K., Bahadori A. (2017). Toward genetic programming (GP) approach for estimation of hydrocarbon/water interfacial tension, *Journal of Molecular Liquids*, **230**, pp. 175-189.

Ruan J., Zhang C., Li Y., Li P., Yang Z., Chen X., Huang M., Zhang T. (2017). Improving the efficiency of dissolved oxygen control using an on-line control system based on a genetic algorithm evolving FWNN software sensor, *Journal of Environmental Management*, **187**, pp. 550-559.

Rumelhart D., Hinton G., Williams R. (1986). Learning representations by back propagating error, *Nature*, **323**, pp. 533–536.

Sałat R., Awtoniuk M., Korpysz K. (2017). Black-box identification of a pilot-scale dryer model: A support vector regression and an imperialist competitive algorithm approach, *IFAC-Papers Online*, **50(1)**, pp. 1559-1564.

Shamsipur M., Ghavami R., Sharghi H., Hemmateenejad B. (2008). Highly correlating distance/ connectivity-based topological indices: 5. Accurate prediction of liquid density of organic molecules using PCR and PC-ANN, *Journal of Molecular Graphics and Modelling*, **27(4)**, pp.506-511.

Sharma A., Saikia A., Khare P., Dutta D.K., Baruah B.P. (2014). The chemical composition of tertiary Indian coal ash and its combustion behaviour – a statistical approach: Part 2, *J. Earth Syst. Sci.*, **123(6)**, pp. 1439–1449.

Sharma B.K., Sharma M.P., Roy S.K., Kumar S., Tendulkar S.B., Tambe S.S., Kulkarni B.D. (1998). Fischer–Tropsch synthesis with Co/SiO_2–Al_2O_3 catalyst and steady-state modeling using artificial neural networks, *Fuel*, **77(15)**, pp. 1763–1768.

Sharma S., Tambe S.S. (2014). Soft-sensor development for biochemical systems using genetic programming, *Biochemical Engineering Journal*, **85**, pp. 89–100.

Shokry A., Dombayci C., Espuna A. (2016). Multiparametric metamodels for model predictive control of chemical processes, *Computer Aided Chemical Engineering*, **38**, pp. 937-942.

Shrinivas K., Kulkarni R.P., Shaikh S., Ghorpade R.V., Vyas R., Tambe S.S., Ponrathnam S., Kulkarni B.D. (2016). Prediction of reactivity ratios in free radical copolymerization from monomer resonance–polarity (Q–e) parameters: genetic programming-based models, *International Journal of Chemical Reactor Engineering*, **14(1)**, pp. 361-372.

Selvaraj R., Deshpande P.B., Tambe S.S., Kulkarni B.D. (1995). Neural networks for the identification of MSF desalination plants, *Desalination*, **101(2)**, pp. 185-193.

Soleimani R., Shoushtari N.A., Mirza B., Salahi A. (2013). Experimental investigation, modeling and optimization of membrane separation using artificial neural network and multi-objective optimization using genetic algorithm, *Chemical Engineering Research and Design*, **91(5)**, pp. 883-903.

Spall J. C. (2004). *Stochastic Optimization in 2004- Handbook of Computational Statistics* (J. Gentle, W. Härdle, and Y. Mori, eds.), Chapter 6, pages 170-194, Springer Heidelberg.

Steiger J. H. (1980). Tests for comparing elements of a correlation matrix, *Psychological Bulletin*, **87(2)**, pp.245-251.

Syu M., Tsao G., Austin G., Celotto G., D'Amore T. (1994). Neural network modeling for predicting brewing fermentations, *Journal of the American Society of Brewing Chemist*, **52**, pp. 15-19.

Tambe S. S., Kulkarni B. D., Deshpande P. B. (1996). *Elements of artificial neural networks with selected applications in chemical engineering, and chemical & biological sciences*. Louisville: Simulation & Advanced Controls Inc. USA.

Tan W.L., Nor N.M., Bakar M.Z., Ahmad Z., Sata S.A. (2012). Optimum parameters for fault detection and diagnosis system of batch reaction using multiple neural networks, *Journal of Loss Prevention in the Process Industries*, **25 (1)**, pp. 138-141.

Tang X., Zhuang L., Jiang C. (2009). Prediction of silicon content in hot metal using support vector regression based on chaos particle swarm optimization, *Expert Systems with Applications*, **36(9)**, pp.11853-11857.

Tashvigh A.A., Ashtiani F.Z., Karimi M., Okhovat A. (2015). A novel approach for estimation of solvent activity in polymer solutions using genetic programming, *Calphad*, **51**, pp. 35-41.

Tendulkar S.B., Tambe S.S., Chandra I., Rao P.V., Naik R.V., Kulkarni B.D. (1998). Hydroxylation of Phenol to Dihydroxybenzenes: Development of Artificial Neural-Network-Based Process Identification and Model Predictive Control Strategies for a Pilot Plant Scale Reactor, *Ind. Eng. Chem. Res.*, **37 (6)**, pp 2081–2085.

Tian W., Hu M., Li C. (2014). Fault prediction based on dynamic model and grey time series model in chemical processes, *Chinese Journal of Chemical Engineering*, **22(6)**, pp.643-650.

Tun K., Lakshminarayan S. (2004). Identification of algebraic and state space models using genetic programming, *IFAC Proceedings Volumes*, **37(9)**, pp. 311-316.

Valero S., Argente E., Botti V., Serra J., Serna P., Moliner M., Corma A. (2009). DoE framework for catalyst development based on soft computing techniques, *Computers & Chemical Engineering*, **33(1)**, pp.225-238.

Vapnik V. *The Nature of Statisitcal Learning Theory*, Springer Verlag, New York, 1995.

Verma D., Goel P., Patil-Shinde V., Tambe S.S (2016). Use genetic programming for selecting predictor variables and modeling in process identification, *IEEE Control Conference (ICC)*, pp.230-237.

Venkatesan C., Raskar S., Tambe S.S., Kulkarni B.D., Keshavamurty R. (1997). Prediction of all India summer monsoon rainfall using error-back-propagation neural networks, *Meteorology and Atmospheric Physics*, **62**, pp. 225-240.

Vora N., Tambe S.S., Kulkarni B.D. (1997). Counter-propagation neural networks for fault detection and diagnosis, *Computers and Chemical Engineering*, **21(2)**, pp.177-185.

Vyas R., Goel P., Tambe S.S. (2015). Genetic programming applications in chemical sciences and engineering, 2015, In: "*Handbook of Genetic Programming Applications*," Amir H. Gandomi, Amir H. Alavi and Conor Ryan (Eds.), Chapter 5, pp. 99-140, Springer International Publishing, Switzerland, DOI 10.1007/978-3-319-20883-1.

Wang D., Lu W.Z. (2006). Forecasting of ozone level in time series using MLP model with a novel hybrid training algorithm, *Atmospheric Environment*, **40(5)**, pp. 913-924.

Wang P., Yang C., Tian X., Huang D. (2014). Adaptive nonlinear model predictive control using an on-line support vector regression updating strategy, *Chinese Journal of Chemical Engineering*, **22(7)**, pp. 774-781.

Wang X., Jiang B., Lu N., Zhang C. (2018). Dynamic fault prognosis for multivariate degradation process, *Neurocomputing*, **275**, pp. 1112-1120.

Wang X., Qin B., Xu H., Zhu W. (2015). Rotary drying process modeling and online compensation, *Control Engineering Practice*, **41**, pp. 38-46.

Wang Y., Yan F., Jia Q., Wang Q. (2017). Assessment for multi-endpoint values of carbon nanotubes: Quantitative nanostructure-property relationship modeling with norm indexes, *Journal of Molecular Liquids*, **248**, pp.399-405.

Xie W.M., Zhang R., Li W.W., Ni B.J., Fang F., Sheng G.P., Yu H.Q., Song J., Le D.Z., Bi X. J., Liu C.Q., Yang M. (2011). Simulation and optimization of a full-scale Carrousel oxidation ditch plant for municipal wastewater treatment, *Biochemical Engineering Journal*, **56(1-2)**, pp.9-16.

Xiong R., Gallivan M.G. (2008). Real-time monitoring of thin film microstructure in chemical vapor deposition using a modified moving horizon estimation, *IFAC Proceedings Volumes*, **41(2)**, pp.10480-10485.

Yan Y., Wang L., Wang T., Wang X. , Hu Y. ,Duan Q. (2018). Application of soft computing techniques to multiphase flow measurement: A review, *Flow Measurement and Instrumentation*, **60**, pp.30-43.

Yang S., Lu W., Chen N., Hu Q. (2005). Support vector regression based QSPR for the prediction of some physicochemical properties of alkyl benzenes, *Journal of Molecular Structure: THEOCHEM*, **719(1-3)**, pp. 119-127.

Yousefinejad S., Hemmateenejad B. (2015). Chemometrics tools in QSAR/QSPR studies: A historical perspective, *Chemometrics and Intelligent Laboratory Systems*, **149**, pp. 177-204.

Yu D.L., Gomm J.B. (2003). Implementation of neural network predictive control to a multivariable chemical reactor, *Control Engineering Practice*, **11(11)**, pp.1315-1323.

Yu J. (2012). A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses, *Computers & Chemical Engineering*, **41**, PP. 134-144.

Yu J. (2012). Local and global principal component analysis for process monitoring, *Journal of Process Control*, **22(7)**, pp. 1358-1373.

Yuan X., Huang B., Ge Z., Song Z. (2016). Double locally weighted principal component regression for soft sensor with sample selection under supervised latent structure, *Chemometrics and Intelligent Laboratory Systems*, **153**, pp. 116-125.

Yuan W., Odjo A., Sammons N.E., Caballero J., Eden M.R. (2009). Process structure optimization using a hybrid disjunctive-genetic programming approach, *Computer Aided Chemical Engineering*, **27**, pp. 669-674.

Zhang G., Patuwo B. E., Hu M. Y. (1998). Forecasting with artificial neural networks: The state of the art, *International journal of forecasting*, **14(1)**, pp. 35-62.

Zhang L., Jack L., Nandi A. (2005). Fault detection using genetic programming, *Mechanical Systems and Signal Processing*, **19(2)**, pp. 271-289.

Zhang S., Wang F., He D., Jia R. (2012). Real-time product quality control for batch processes based on stacked least-squares support vector regression models, *Computers & Chemical Engineering*, **36**, pp. 217-226.

Zhou Y.P., Jiang J.H., Lin W.Q., Zou H.Y., Wu H. L., Shen G.L., Yu R.Q. (2006). Boosting support vector regression in QSAR studies of bioactivities of chemical compounds, *European Journal of Pharmaceutical Sciences*, **28(4)**, pp. 344-353.

Zhou W., Wu S., Dai Z., Yuan C., Xiang Y., Chen J., Sun C., Zhou Q., Yuan Z. (2015). Nonlinear QSAR models with high-dimensional descriptor selection and SVR improve toxicity prediction and evaluation of phenols on Photobacterium phosphoreum, *Chemometrics and Intelligent Laboratory Systems*, **145**, pp. Pages 30-38.

Zurada J.M. (1992). *Introduction to artificial neural systems*, (Vol. 8). St. Paul: West.

CHAPTER 3

PREDICTION OF RATE CONSTANT OF PHOTOCATALYTIC DEGRADATION OF PHARMACEUTICAL POLLUTANTS BY MACHINE LEARNING BASED FORMALISMS

Abstract

Personal and healthcare products affect the quality of the ambient water when these materials come into the environment in the form of metabolites excreted by the animal and human populations or as effluents from hospitals, pharmacies, and chemical manufacturing facilities. These compounds are difficult to capture or confine and therefore pose a serious threat to the aquatic ecosystem, and human health. Photocatalytic degradation (PCD) is seen to be an attractive and inexpensive method, when compared to the ones such as photo-Fenton oxidation, sonolysis, ozonation, and photolysis for the elimination of pharmaceutical compounds such as *Ciprofloxacin*, *Naproxen,* and *Paracetamol* from the wastewater. Owing to the underlying complex nonlinear physicochemical phenomena, the design and construction of a "first principles" model for predicting the rate constant of the PCD is a time-consuming, tedious and costly task. To subjugate the said difficulty, in this study, mathematical models predicting the rate constant of the PCD of the above-stated three pharmaceuticals have been developed using two computational intelligence (CI) based data-driven modeling methods, namely, *genetic programming* (GP) and *artificial neural networks* (ANNs). These models use process conditions and molecular structural parameters of the three pharmaceutical molecules as inputs to predict the magnitude of the PCD rate constant. Among the ANN and GP-based models, the latter was found to possess better prediction accuracy and generalization capability (CC > 0.973 and $RMSE$ < 0.64). Accordingly, the GP-based model was subsequently used in optimizing the process conditions of the PCD of the three pharmaceutical molecules (Ciprofloxacin, Naproxen, and Paracetamol). The said optimization was performed using a stochastic nonlinear optimization method namely *genetic algorithms* (GA). Next, the GA-provided optimal process conditions were subjected to experimental validation. Here, all the optimal conditions were validated successfully. The CI-based PCD modeling and optimization strategies explored in this chapter can be fruitfully used for similar purposes with other pharmaceutical degradation reactions.

3.1 INTRODUCTION

A huge increase in the human population and industrial activity in the last five to seven decades has resulted in a severe contamination of fresh water resources in some regions of the world. Thus, globally it has become necessary to protect the scarce sources of the fresh non-contaminated potable water to fulfill the ever-increasing demand from the widening human population. The presence of toxins and harmful chemicals discharged from the healthcare, and personal care products through metabolites excreted by the human beings, and disposals from hospitals, pharmacies, and chemical manufacturing industries, have made a large number of fresh water resources unfit for drinking, and domestic usage. Some of them are ciprofloxacin (CFX), Naproxen (NPX) and paracetamol (PARA).

Therefore, the biodegradation of CFX have been studied using activated sludge process by Li and Zhang (2010). Recently, the studies on biodegradation of NPX have been reported by Ding et al. (2017) using enzymatic catalytic reactions. Similarly, biodegradation of paracetamol has been studied by Zhang et al. (2013). In most of these studies shows the long-time duration (in days) is necessary for biodegradation of pharmaceutical pollutants. Additionally, the non-biodegradable pharmaceutical pollutants present in the wastewater have posed a serious threat to the ecosystem, and human and animal health. However, such pollutants are difficult to remove only by the conventional waste water purification processes as they are time-consuming, costly and need a large amount of chemicals, which are difficult to separate post-purification. This problem is well answered by the photo-catalytic degradation process. It is an advanced oxidation process with several advantages; it is also cost-effective.

3.2 WHY PHOTOCATALYTIC DEGRADATION?

The wastewater is characterized by a high organic load and a low biodegradability index (BOD_5 /COD); this means that it is difficult to degrade it by the traditional treatments, such as the biological treatment. The wastewater containing pharmaceutical pollutants typically has turbid color and oily touch; advanced oxidation processes (AOPs) are needed to degrade these complex molecules (Tchobanoglous et al., 2002). The AOPs are sub-categorized as Ozonation, Photo-Fenton Oxidation, Sonolysis, Photolysis, Sono Hybrid, Hybrid AOP, and Photo- catalytic degradation (PCD). The implementation, advantages, and disadvantages of each AOP are briefly stated below.

3.2.1 Ozonation

Ozone is unstable and degrades over a time frame ranging from a few seconds to 30 minutes. The rate of its degradation is a function of the water chemistry, pH, and water temperature (Dodd et al., 2006). An ozonation system includes passing dry, clean air through a high voltage electric discharge, which creates an ozone concentration of approximately 1% or 10,000 mg/L. The raw water is then passed through a venturi throat so as to create vacuum and pulling the ozone gas into the water. The air is then bubbled through the water being treated.

- **Advantages**
  - ❖ Ozone affects over a wide pH range and rapidly reacts with the bacteria, viruses, and protozoans and possesses stronger germicidal properties than the chlorination.
  - ❖ Ozone has a very strong oxidizing efficiency over a short time interval.
  - ❖ The treatment does not add chemicals to the water.
  - ❖ Ozone can eliminate a wide variety of inorganic, organic and microbiological entities and accordingly taste, and odor issues. The microbiological agents include bacteria, viruses, and protozoans (such as Giardia and Cryptosporidium).
- **Disadvantages:**

- Ozone oxidizes iron, manganese, and sulfur in the water to form insoluble metal oxides or elemental sulfur. Removal of these insoluble particles requires post-filtration.
- Involves additional equipment and operational costs; also, not easy to find a professional well-versed in the ozone treatment and system maintenance.
- Ozonation provides no germicidal or disinfection residual to inhibit or prevent regrowth.
- Ozonation by-products are still being evaluated and it is possible that some by-products could be carcinogenic. These may include brominated by-products, aldehydes, ketones, and carboxylic acids. This is one reason why the post-filtration system must include an activated carbon filter.
- The system may require pretreatment of the waste-water for reducing its hardness or addition of polyphosphate to prevent the formation of carbonate scale.
- Ozone is less soluble in water, compared to chlorine, and, therefore, special mixing techniques are needed.
- Potential fire hazards and toxicity issues associated with ozone generation.

3.2.2 Photo-Fenton Oxidation

The process that uses mainly iron as a catalyst is known by the reagent discovered by Fenton (Fenton, 1894). The ferrous iron (Fe^{++}) initiates and catalyzes the decomposition of H_2O_2, resulting in the generation of the hydroxyl radicals. These radicals via photo-reduction of the metal ions initiate the degradation process (Sun et al., 2009). The generation of the radicals involves a complex reaction sequence in an aqueous solution. Generally, Fenton’s oxidation process is characterized by pH adjustment, oxidation reaction, and neutralization and coagulation for precipitation.

- **Advantages**
  - ❖ H_2O_2 can act both as an OH scavenger and an initiator.
  - ❖ The organic substances can be removed at two stages that is oxidation and coagulation.
- **Disadvantages**
  - ❖ Formation of a high concentration of anions in the treated wastewater.
  - ❖ Generation of large amounts of ferrous iron sludge.

3.2.3 Sonolysis

The disadvantage of the above-stated processes namely the generation of harmful by-products makes it necessary to explore alternative disinfectants and new treatment technologies. To this end, in recent years, ultrasound has emerged as a promising alternative disinfection technique (Bel et al., 2009). The science of ultrasound involves the study of the formation, impact, and applications of sonorous waves occurring at frequencies higher than 20 kHz; this frequency is the upper audibility threshold for the human ear. Ultrasound has been studied with reference to physical, chemical, medical, industrial, and environmental engineering applications.

- **Advantages**
  - ❖ Simple, flexible design with low capital costs
  - ❖ Easy upgrading of the conventional treatment unit
  - ❖ The high efficiency of several bacteria inactivation
  - ❖ Oxidation of natural organic matter and degradation of chemicals pollutants
  - ❖ No production of conventional disinfection by-products
  - ❖ High synergy/improved efficiency in combination with conventional disinfection treatments (O_3; Cl_2; UV).
- **Disadvantages**

- ❖ Design criteria still developing
- ❖ The increase of water turbidity
- ❖ Maintenance/replacement of ultrasound probe
- ❖ Lack of remaining disinfection capacity

3.2.4 Photolysis

Addition of a photocatalyst accelerates the decrease of total organic content/loading (TOC) by photolytic and photocatalytic reactions. For example, the treatment of the swimming pool water by UV- LED radiation without the photo-catalyst showed no decrease of the TOC since no photolytic reaction took place. A slight decrease of the TOC can be observed when TiO_2 was added, which means that UV-LEDs are capable to excite the photocatalyst TiO_2 (Van Doorslaer et al., 2011).

- **Advantage**
  - ❖ Efficient and effective due to the specific wavelength emitted by the radiation source.
- **Disadvantage**
  - ❖ Produces unbalanced lethal hydroxyl radicals (OH*) which may impart erosion to vessel content.

3.2.5 Sono Hybrid

In brief, this process is a combination of Sono-catalysis, Sono-Fenton, and Sono-photo-catalysis. The experimental details of this method are provided by Chakma and Moholkar (2015).

- **Advantage**
  - ❖ Highly effective in commercial wastewater treatment plants
- **Disadvantage**
  - ❖ High operational and maintenance cost.

3.2.6 Hybrid AOP

This process is a combination of the sonolysis, Fenton and, UV degradation processes. The process details of the hybrid-AOP are well explained by Tchobanoglous et al. (2002).

- **Advantage**
    - ❖ Highly effective in commercial waste-water treatment plants; effects savings in the catalyst costs thereby operational cost is lowered.
- **Disadvantage**
    - ❖ High operational cost and useful in commercial processes for dilute concentrations only.

3.2.7 Photocatalytic Degradation

Photocatalytic degradation (PCD) of organic pollutants is a promising technology since it directly degrades pollutants instead of transforming them. Moreover, this is done at ambient conditions. The principle of this process is an acceleration of the photoreaction in the presence of a catalyst (Belden et al., 2007).

- **Advantages**
    - ❖ Capable of removing a wide range of organic pollutants (such as pesticides, and herbicides) and micropollutants (such as endocrine disrupting compounds).
    - ❖ The use of solar radiation can be used to improve the design of the photo-reactor for reducing the cost of treatment.
    - ❖ Easier to scale up.
- **Disadvantage**
    - ❖ Wide band gap (3.2eV), and need for an efficient and cost-effective catalyst for high photon-efficiency to utilize wider solar spectra.

Currently, removal of the pharmaceutically-active micropollutants such as *ciprofloxacin* (CFX), *naproxen* (NPX), and *paracetamol* (PARA), from the wastewater up to a permissible limit is an issue of concern since commonly used wastewater treatment plants can not remove these micropollutants completely. Among the various above-described waste-water treatment methods the photocatalytic degradation **(**PCD)—since it has several attractive features, and no major disadvantages—is found to be an efficient and inexpensive pollutant-removal approach when compared with the other methods (Gebhardt and Schröder, 2007). Remediation of the CFX, NPX and PARA compounds has been studied by the photocatalytic degradation (Bhatkhande et al., 2001; Kamble et al.,2006). In several studies, TiO_2 was used as an effective catalyst for the removal of pollutants from the wastewater successfully. The performance of *N*-doped TiO_2 for the PCD of pharmaceutical pollutants, namely, *Ciprofloxacin* (CFX), *Naproxen* (NPX), and *Paracetamol* (PARA), was rigorously studied by Shetty et al. (2016) and Shetty et al. (2017). This chapter presents the results of a study wherein (i) state-of-the art computational intelligence based mathematical models have been developed for the prediction of the PCD rate constant governed by the pseudo-first-order kinetics using the data collected by Shetty et al. (2016; 2017), (ii) the best performing CI-based model has been employed for optimizing the PCD process performance using a stochastic optimization method, namely genetic algorithms (GA), and (iii) the optimal process conditions provide by the GA have been validated successfully experimentally.

3.3 EXPERIMENTAL METHODOLOGY

In here, an overview of the experimental procedure that was used to conduct the photocatalytic degradation of pharmaceutical pollutants is provided. A detailed discussion of the PCD experimentation can be found in Shetty et al. (2017).

An annular borosilicate glass reactor was used as the photocatalytic reactor (see Figure 3.1) for performing the PCD of individual pharmaceutical pollutants using artificial and solar radiation. The source of artificial radiation was a visible tungsten lamp with the power output of 400 W. The de-ionized water was used to prepare an aqueous solution of each pharmaceutical pollutant and fed to the solution inlet as shown in Figure 3.1. The temperature of the reaction mixture was kept constant by supplying the cold/chilled water through the annular space between the lamp and bottle, and it was measured at regular intervals by the temperature indicator. The reactor assembly shown in Figure 3.1 can be surrounded by parabolic reflectors if the experiments need to be conducted in the presence of solar radiation.

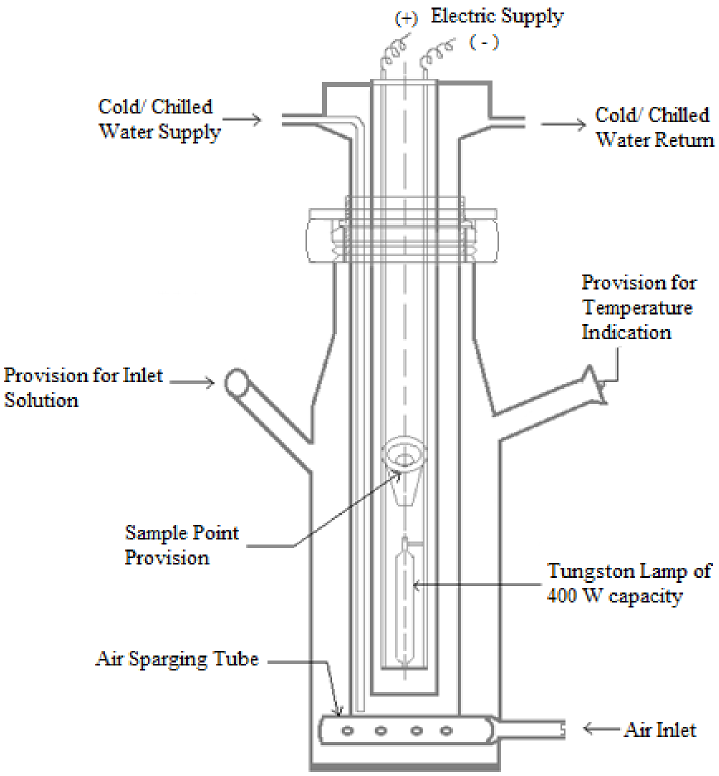

(+) Electric Supply
(-)
Cold/ Chilled Water Supply
Cold/ Chilled Water Return
Provision for Temperature Indication
Provision for Inlet Solution
Sample Point Provision
Tungsten Lamp of 400 W capacity
Air Sparging Tube
Air Inlet

Figure 3.1: Schematic diagram of the experimental assembly of PCD of pharmaceutical pollutants

In the photocatalytic experiments, N-doped TiO_2 or Aeroxide® P-25 TiO_2, was used as the effective photocatalyst (Lin et al., 2012; Shetty et al., 2016). In each experimental run, 500 ml of the aqueous contaminated solution prepared using a single pharmaceutical pollutant was charged in the reactor (Shetty et al., 2017). The air was sparged at the bottom of the reactor to provide the necessary oxidation and to keep the N-doped TiO_2 catalyst in suspension via a ring sparger. Before exposing the reactor assembly to the artificial radiation, the solution was kept in a stirring condition in equilibrium for 15 min in the dark. The artificial radiation was observed to be more efficient to affect the degradation for Ciprofloxacin and Naproxen, while solar radiation was more effective for Paracetamol degradation (Desale et al., 2013; Shetty et al., 2016).

The solution pH was adjusted using 0.1 N $HClO_4$ and 0.1 N NaOH aqueous solutions. Batch adsorption experiments were carried out in a 100 C.C. stoppered conical flask. A 50 ml aqueous solution of 100 mg/L concentration of pollutant sample was used in the conical flask covered by the silver foil to avoid contact with the light. A known amount of the N-doped TiO_2 was added to the flask and then the flask was kept on a rotary shaker at a speed of 250 rpm at 30°C (ambient temperature) for 24 h to arrive at the equilibrium. The residual concentration of the pollutant sample was analyzed by using the HPLC method. The technical details of the analytical procedure are given by Shetty et al. (2017). A total of 36 PCD experiments were conducted to study the effects of variation in the operating conditions on the magnitude of the rate constant pertaining to the PCD of each of the three pharmaceutical molecules. In these experiments, the initial concentrations of Ciprofloxacin, Naproxen and Paracetamol were 97.4 mg/L, 98.0 mg/L and 108.0 mg/L, respectively.

3.4 MODELING OF PHOTOCATALYTIC DEGRADATION PROCESS

For designing a photocatalytic degradation process, an accurate knowledge of or a prediction method for the degradation rate constant is essential. Since performing experiments is costly, tedious, and time-consuming it is desirable to develop suitable mathematical models for the prediction of the stated rate constants. Among the two methods that are available for such a modeling, namely, *phenomenological* (mechanistic) and *empirical*, the latter is less rigorous and inexpensive since unlike the first method, it does not require complete knowledge of the physicochemical phenomena underlying the reaction. Accordingly, we have employed computational intelligence (CI) based exclusively data-driven state-of-the-art methods namely genetic programming (GP) and artificial neural networks (ANNs) for modeling and optimization of the photocatalytic degradation process for the removal of *ciprofloxacin* (CFX), *naproxen* (NPX), and *paracetamol* (PARA) from the waste-water. Specifically, the CI-based models predict the magnitude of the rate constant (K_c) for the PCD of ciprofloxacin, naproxen and paracetamol. The PCD experimental data used in the modeling and optimization are given in Shetty et al. (2016). In the next step, the CI-based model possessing high prediction accuracy and generalization performance was used to obtain the optimal conditions for the PCD of *ciprofloxacin* (CFX), *naproxen* (NPX), and *paracetamol* (PARA).

In this study, four experimental parameters, namely, *pH*, *time of exposure of pharmaceutical pollutant to the solar radiation* (min), the *degraded concentration of pharmaceutical pollutant* (mg/L), and *the ratio of instantaneous concentration to initial concentration*, have been considered as the predictor (input) space of the model. The other influential operational parameters that were kept constant and, therefore, not considered as model inputs are *initial concentration of the pollutant* (mg/L), *solar intensity* (W/m^2), and *catalyst loading* (g/L), and the magnitudes of these parameters have been: 100 mg/L, 815±10 W/m^{2,} and 1 g/L, respectively. Since, a single comprehensive model was developed for predicting the rate constant for all the three pollutants, it was necessary that the input space of the model includes attributes that distinguish between the pollutants. This was achieved using the structural parameters of the pharmaceutical molecules (CFX, NPX, and PARA) provided by *Screening Assistant 2* (Le Guillouxet al., 2012) software package. These parameters were considered as the additional (to the four experimental parameters) ones and are as follows: (i) *number of aromatic atoms* (b_1), (ii) *number of atoms* (b_2), (iii) *number* *of bonds* (b_3), (iv) *molecular weight* (b_4), (v) *number of heavy atoms* (b_5), (vi) *unit stripped logarithm of the solubility* (b_6), (vii) *sum of atomic polarizabilities* (b_7), (viii) *number of hydrophobic atoms* (b_8), (ix) *topological surface area* (b_9), (x) *logarithm of the octanol/water partition coefficient obtained from linear atom type model* (b_{10}), (xi) *Weiner path* (b_{11}), (xii) *number of rings* (b_{12}), (xiii) *logarithm of the octanol/water partition coefficient of the given structure* (b_{13}), (xiv) *molecular refractivity* (b_{14}) and (xv) *number of moles* (b_{15}).

3.4.1. Sensitivity Analysis

For ascertaining the extent to which the model output (rate constant, K_c expressed in unit as, 1/ min) is influenced by the individual independent factors molecular structural parameters (MSP), sensitivity analysis (SA) was performed. Such an analysis helps in understanding the behavior of the model and the coherence between a model and its output (Saltelli et al., 1999). Specifically, it ranks the inputs in the order of their influence on the model output (i.e., the rate constant, K_c). For performing SA, IBM SPSS package (2011), was used on the set of fifteen MSPs of the pharmaceutical pollutants, which have a distinct effect on the PCD. Only the first fourteen structural parameters (b_1—b_{14}) were considered in the SA since the number of moles (b_{15}) was kept constant for all the three pharmaceutical molecules. The results of the SA shown in Figure 3.2 indicate that there exist two distinct groups of structural parameters with importance magnitudes of less than and greater than 20%, respectively. Thus, only those ten SPs with relatively high (>20%) importance values, namely, *molecular refractivity* (b_{14}), *Weiner path* (b_{11}), *number of rings* (b_{12}), *sum of atomic polarizabilities* (b_7), *number of hydrophobic atoms* (b_8), *unit stripped logarithm of the solubility* (b_6), *number of bonds* (b_3), *logarithm of the octanol/water partition coefficient obtained from linear atom type model* (b_{10}), *logarithm of the octanol/water partition coefficient of the given structure* (b_{13}), and *topological surface area* (b_9),were considered along with the four experimental parameters in the input space of the CI-based rate constant predicting models.

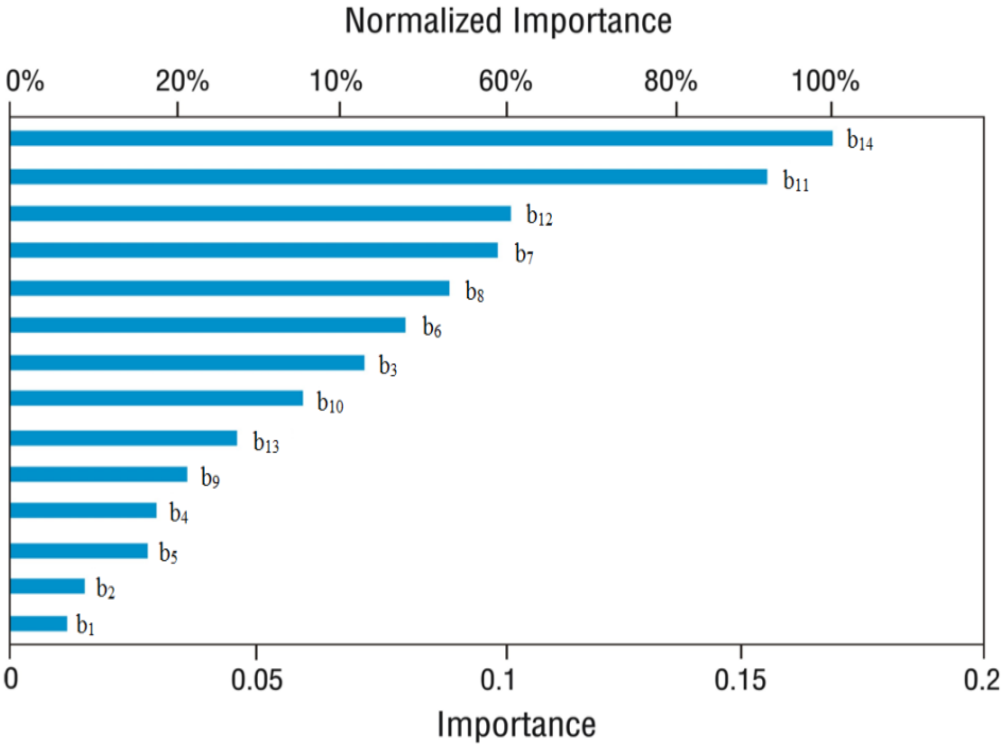


Figure 3.2: Sensitivity analysis of the structural parameters of the pharmaceutical molecules

3.4.2. Principal Component Analysis (PCA)

The principal component analysis (Geladi and Kowalski, 1986) was performed on the data set representing the model inputs. This was done for removing any linearly correlated inputs and thereby reducing the dimensionality of the input space of the CI-based models. The PCA described in detail in Chapter 2 (section 2.6.1) is a multivariate statistical technique that employs an orthogonal transformation to convert a set of linearly correlated data into a set of linearly uncorrelated variables termed *principal components* (PCs). The first few PCs (which are lesser in number than the original input variables) that capture a major portion of the data variance can be selected as model inputs thereby reducing the dimensionality of the input space. The dimensionality-reduced dataset cuts down the computational load pertaining to the construction of data-driven models as also data redundancy.

In this study, *RapidMiner* software package (Mierswa et al., 2006) was used to perform PCA on the fourteen-dimensional predictor input space consisting of (a) the four experimental parameters namely, *pH* (x_1), *time* (x_2), *degraded concentration of pharmaceutical pollutant* (x_3), and *ratio of instantaneous concentration to initial concentration* (x_4), and (b) the ten influential structural parameters identified by the sensitivity analysis. The dataset consisting of fourteen inputs and the corresponding rate constant magnitudes is listed in **Table 3.1**. Prior to performing PCA, all the listed inputs were normalized using "Z-score" normalization procedure using the following equation.

$$\hat{x}_r^i = \frac{x_r^i - \bar{x}_r}{\sigma_r}; i = 1,2,\dots,N_p; r = 1,2,\dots,R \qquad (3.1)$$

where $\hat{x}_r^i$ denotes the normalized values of r^{th} input variable (r = 1, 2, …,R); x_r^i denotes the i^{th} value of the r^{th} non-normalized input variable, x_r; N_p represents the number of observations/ samples in the data set; R designates the number of inputs to PCA; $\bar{x}_r$ and σ_r represent the mean and standard deviation of x_r. The mean and standard deviation formulas and their values used in the normalization procedure are given from Eq. 3.2 to 3.5.

$$\bar{x}_r = \frac{\sum_{i=1}^{N_p} x_r^i}{N_p};\ i = 1,2,\dots,N_p\ ; r = 1,2,\dots.,R \qquad (3.2)$$

$$\sigma_r = \sqrt{\frac{\sum_{i=1}^{N_p}\left(x_r^i - \bar{x}_r\right)}{N_p}};\ i = 1,2,\dots,N_p \quad r = 1,2,\dots.,R \qquad (3.3)$$

where,

$\bar{x}_1 = 6.21,\ \bar{x}_2 = 45.43,\ \bar{x}_3 = 43.24,\ \bar{x}_4 = 0.408,\ \bar{x}_5 = 7.15,\ \bar{x}_6 = 796.5,\ \bar{x}_7 =$
$2.44, \bar{x}_8 = 39.49, \bar{x}_9 = 12.5, \bar{x}_{10} = -2.90, \bar{x}_{11} = 35.67, \bar{x}_{12} = 1.68, \bar{x}_{13} = 1.81,$
$\bar{x}_{14} = 67.15$ (3.4)

$\sigma_1 = 0.892, \sigma_2 = 38.01, \sigma_3 = 40.15, \sigma_4 = 0.368,\ \sigma_5 = 0.843,\ \sigma_6 =$
$268.83, \sigma_7 = 0.843, \sigma_8 = 4.75, \sigma_9 = 2.32, \sigma_{10} = 0.544, \sigma_{11} = 5.138, \sigma_{12} =$
$1.33, \sigma_{13} = 1.02, \sigma_{14} = 17.26$ (3.5)

Similar to the model's inputs, the rate constant magnitudes (K_c) of the PCD reaction were normalized using the same technique; the corresponding mean and standard deviation values are 1.226 and 3.33, respectively. The magnitudes (%) of the variance in the example dataset captured by the fourteen principal components are as follows: PC_1 - 45.7, PC_2 – 40, PC_3 – 7.6, PC_4 – 6.5, PC_5 – 0.3, and PC_6 – PC_{14} ≈ 0.0. As can be seen, the first four PCs have captured a large percentage (≈ 99.7%) of the variance in the fourteen-dimensional normalized input space of the CI-based models to be developed. Accordingly, the first four *PC*s were considered in the GP and ANN based modeling instead of the original 14 inputs. Consequently, the dimensionality of the input space got reduced to only four and, thereby, decreasing the computational load in modeling substantially. The general formula for computing the scores of the principal components with maximum variance created in a PCA is given below:

$$PC_n = a_{11}(\hat{x}_1) + a_{12}(\hat{x}_2) + \cdots + a_{1r}(\hat{x}_r) \tag{3.6}$$

where,

a_{11} = the regression coefficient ("weight") for the first input variable, as used in creating the principal component 1 (PC_1); other coefficients can be interpreted similarly. Using the weights obtained in PCA, the four PCA-transformed inputs are defined as below:

$$\begin{aligned}PC_1 = &-0.018\hat{x}_1 - 0.11\hat{x}_2 + 0.128\hat{x}_3 + 0.129\hat{x}_4 + 0.3\hat{x}_5 + 0.38\hat{x}_6 + 0.286\hat{x}_7 + 0.315\hat{x}_8 \\ &+ 0.076\hat{x}_9 + 0.315\hat{x}_{10} + 0.329\hat{x}_{11} - 0.338\hat{x}_{12} - 0.332\hat{x}_{13} + 0.319\hat{x}_{14}\end{aligned} \tag{3.7}$$

$$\begin{aligned}PC_2 = &\,0.04\hat{x}_1 + 0.069\hat{x}_2 + 0.391\hat{x}_3 + 0.389\hat{x}_4 - 0.276\hat{x}_5 - 0.115\hat{x}_6 - 0.292\hat{x}_7 - 0.255\hat{x}_8 \\ &- 0.414\hat{x}_9 + 0.252\hat{x}_{10} - 0.235\hat{x}_{11} - 0.217\hat{x}_{12} - 0.227\hat{x}_{13} + 0.246\hat{x}_{14}\end{aligned} \tag{3.8}$$

$$\begin{aligned}PC_3 = &\,0.728\hat{x}_1 + 0.637\hat{x}_2 - 0.141\hat{x}_3 - 0.156\hat{x}_4 + 0.018\hat{x}_5 + 0.042\hat{x}_6 + 0.015\hat{x}_7 + 0.022\hat{x}_8 \\ &- 0.023\hat{x}_9 + 0.063\hat{x}_{10} + 0.026\hat{x}_{11} - 0.063\hat{x}_{12} - 0.063\hat{x}_{13} + 0.063\hat{x}_{14}\end{aligned} \tag{3.9}$$

$$PC_4 = -0.682\hat{x}_1 + 0.71\hat{x}_2 - 0.087\hat{x}_3 - 0.097\hat{x}_4 + 0.007\hat{x}_5 + 0.029\hat{x}_6 + 0.004\hat{x}_7 + 0.01\hat{x}_8$$
$$- 0.028\hat{x}_9 + 0.054\hat{x}_{10} + 0.013\hat{x}_{11} - 0.053\hat{x}_{12} - 0.053\hat{x}_{13} + 0.054\hat{x}_{14} \tag{3.10}$$

where $\hat{x}_r$; r = 1, 2, …, R, denote the normalized scores of the fourteen input variable values (x_r).

Table 3.1: Input Variables consists of Photocatalyst degradation experimental parameters and structural attributes of pharmaceutical molecules.

Pharmaceutical Pollutants	***Photocatalyst Degradation Experimental Parameters* (Shetty et al., 2016)**				***Structural Attributes of Pharmaceutical Molecules* (Le Guilloux et al., 2012)**										**K_c , (1/ min) (Shetty et al., 2016)**	***Initial concentration in mg/L***	***Percent degradation obtained***
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}			
Ciprofloxacin	4.5	0.1	40.56	0.418	8.70	1234	4	48.2	16	-2.74	45	1.04	1.37	72.88	8.719	97.4	95.42
		15	12.83	0.132											0.135		
		30	4.46	0.046											0.103		
Naproxen	5.4	0.1	38.55	0.393	6.66	530	2	36.4	13	-3.55	32	3.29	3.04	46.53	9.33	98.0	95.51
		15	11.15	0.114											0.145		
		30	11.77	0.120											0.071		
		45	8.91	0.091											0.053		
		60	8.61	0.088											0.041		
		90	5.50	0.056											0.032		
		120	4.40	0.045											0.025		
Paracetamol	5.6	0.1	102.7	0.951	6.75	813	2	37.6	10	-2.35	34	0.43	0.83	84.5	0.502	108.0	21.31
		15	99.01	0.917											0.0057		
		30	96.02	0.889											0.0039		
		45	95.36	0.883											0.0028		
		60	95.39	0.883											0.0021		
		90	86.11	0.797											0.0025		
		120	84.98	0.787											0.0019		
Ciprofloxacin	7	0.1	32.41	0.331	8.70	1234	4	48.2	16	-2.74	45	1.04	1.37	72.88	11.045	97.4	99.62
		15	11.14	0.114											0.1448		
		30	2.65	0.027											0.1203		
		45	1.24	0.013											0.097		
		60	0.37	0.004											0.092		
Naproxen	7	0.1	28.15	0.287	6.66	530	2	36.4	13	-3.55	32	3.29	3.04	46.53	12.474	98.0	99.11
		15	24.17	0.247											0.093		
		30	16.36	0.167											0.0597		
		45	14.17	0.145											0.0429		
		60	3.51	0.036											0.055		
		90	1.96	0.02											0.0435		
		120	0.87	0.009											0.0394		
Paracetamol	7	0.1	101.3	0.938	6.75	813	2	37.6	10	-2.35	34	0.43	0.83	84.5	0.6365	108.0	35.07
		15	97.67	0.904											0.0067		
		30	93.53	0.866											0.0048		
		45	89.69	0.830											0.0041		
		60	84.60	0.783											0.004		
		90	76.33	0.707											0.0039		
		120	70.12	0.649											0.0036		

For developing the CI-based models, the set consisting of PCA-transformed data from 36 experiments was randomly split into two data subsets as follows: training set (75% data) and test set (25% data). The first data set was used to develop the model while the second one was used to assess the generalization ability of the model.

3.4.3 Development of GP-based model predicting PCD rate constant

Genetic programming (GP) is a computational intelligence (CI) based exclusively data-driven modeling paradigm proposed by Koza (1990) for automatically generating computer programs that perform pre-defined tasks. The other important application of GP is known as "symbolic regression (SR)". The novelty of GP is that provided with an example set consisting of the independent (predictor/input) and dependent (response/output) variables, the GP searches and optimizes the form of an appropriate linear or a nonlinear data fitting function as also all the parameters associated with it. Additionally, GP can identify the key variables and determine their combinations (Patil-Shinde et al., 2016; Vyas et al., 2015). For performing SR, GP uses the "survival of the fittest" and "genetic propagation of characteristics" principles of the Darwinian evolution. The origin, mechanism and implementation procedure of the GP is described in greater details in Chapter 2, section 2.3.3.

3.4.3.1 Results and Discussion (GP-based modeling)

The GP-based equation predicting the magnitude of the photocatalytic degradation constant (K_c) was developed using *Eureqa Formulize* (EF) software package (2009) (Schmidt and Lipson, 2009). A prominent attribute of the EF package for GP-based symbolic regression is that it has been optimized for generating parsimonious (i.e., with reduced complexity) models possessing good generalization ability. For avoiding premature convergence leading to the locally optimal or over-fitted solutions, the EF along with the standard selection, crossover, and mutation operations employs the age-fitness Pareto optimization technique (Vyas et al., 2015**).** During EF-based GP implementation, only those models possessing following characteristics are selected: (a) models containing all the four principal components, i.e. *PC₁*, *PC₂*, *PC₃*, and *PC₄* as inputs (predictors), (b) models with low complexity (smaller number of terms and parameters in the fitting function), and (c) low and comparable magnitudes of correlation coefficient (*CC*) and root mean square error (*RMSE*) for both the training and test set outputs.

The EF package generates distinct mathematical expressions for different random number generator seeds, training and test set sizes, choice of the mathematical operators, error metrics and input normalization schemes. Accordingly, the effects of these attributes on the converged solutions were examined rigorously by varying them systematically to arrive at an overall optimal solution. The prediction accuracy and the generalization performance of the GP-based model were evaluated by computing the *CC* and *RMSE* values using the experimental (target) and the corresponding model-predicted values of K_c.

The overall best GP-based model was selected on the basis of its high *CC* and low *RMSE* magnitudes pertaining to both training and test datasets. The optimal GP-based model for the prediction of rate constant K_c is given below.

$$K_c = 115.6 \times exp(4.70PC_1 + 7.41PC_3PC_4 - 2.37PC_2^2 - 4.51PC_3PC_4^2) - 0.357 - 0.0044PC_2 \qquad (3.11)$$

While all other EF attributes kept constant, the random number generator seed was changed 20 times before the optimal expression given in Eq. (3.11) could be obtained. For clarity, this optimal model searched by GPSR depicted in the form of the tree structure is given in Figure 3.3.

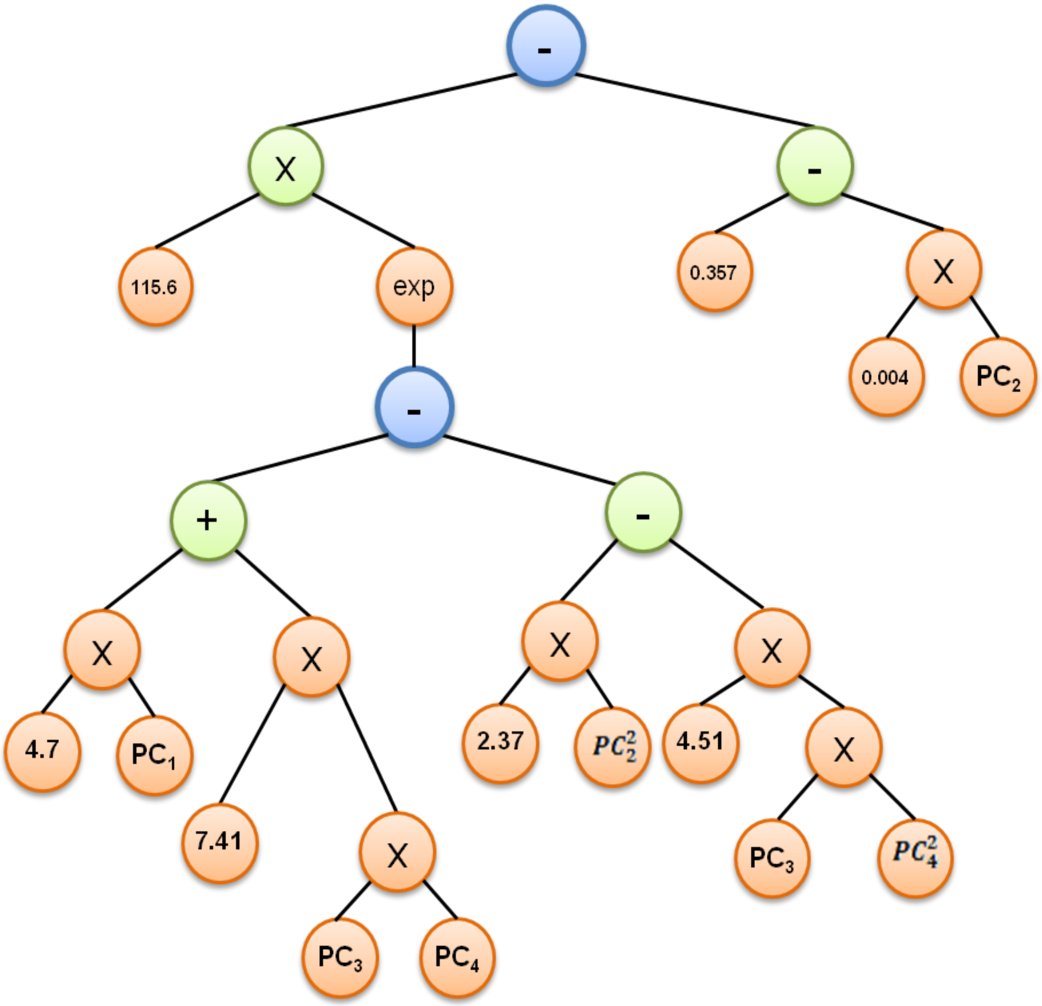


Figure 3.3. Optimal GP-model presented in the form of a tree structure

An important property of the GP-based symbolic regression is that it automatically selects predictors (model inputs) that affect the model output significantly (Patil-Shinde et al., 2016). The optimized form searched by the GP comprises these influential inputs and their combinations. It is thus clear that GPSR automatically (i.e. devoid of any a priori assumptions) makes the choice of the structure, predictors and their combinations, and magnitudes of the parameters of an optimal data-fitting model. In the present case study, the GPSR provided optimal expression (Eq. 3.11) has a nonlinear structure. It also indicates that the K_c magnitude has an exponential dependence on all the four principal components (PC_1, PC_2, PC_3 and PC_4). The choice of all four PCs as inputs, the exponential term, and seven parameters and their magnitudes are made exclusively by the

GPSR without any assistance from the model developer. A drawback of all data-driven modeling strategies such as ANNs (Freeman and Skapura, 1991) and SVR (Vapnik, 1995), which is also shared by the GPSR is that it is difficult to interpret the structure and parameters of the fitted model for unraveling the physical and/or chemical phenomena underlying the system being studied.

Table 3.2: Prediction accuracy and generalization performance of GP- and MLPNN-based models.

Model	*CC*		*RMSE*	
	Training set	Test set	Training set	Test set
GP-based model (Eq. 3.11)	0.998	0.973	0.215	0.008
MLPNN-based Model	0.998	0.982	0.640	0.042

Table 3.2 lists the values of the *CC* and *RMSE* in respect of the K_c predictions made by Eq. 3.11 and their corresponding experimental (desired/target) magnitudes in the training, and test sets. From the high (≈ 0.97) and comparable magnitudes of the *CC* as also low and comparable magnitudes of the *RMSEs* (≈ 0.215) pertaining to the training, and test data, it is clear that the GPSR based model possesses an excellent K_c prediction accuracy and generalization capability. This observation is also supported by the parity plot shown in Figure 3.4 where it is seen that there exists a reasonably good match between the experimental and model predicted K_c values.

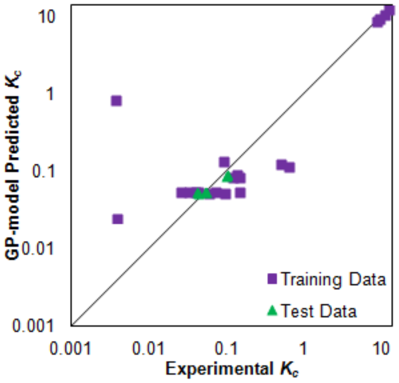


Figure 3.4: Parity Plot of experimental versus GP-model of predicted rate constant (K_c)

The GP strategy/procedure adopted by Eureqa Formulize does not use the “k-fold cross validation” scheme for selecting the optimal model. Specifically, it uses the “single split training-test set” scheme for constructing the model and assessing its generalization ability. Thus, the *k-fold cross-validation* was performed using the optimal model selected by the *Eureqa Formulize*. The mean values of the *CC*s and *RMSE*s of the training and test datasets for $k = 4$ are: Training set: $CC = 0.998$, $RMSE = 0.182$; Test set: $CC = 0.93$, $RMSE = 0.142$. The high *CC* and low *RMSE* magnitudes for both the training and test sets in respect of the four-fold cross-validation clearly support the earlier result that the GPSR model (Eq.3.11) predicting K_{c} possesses an excellent prediction accuracy and generalization performance.

The statistical metrics of the correlation coefficient (*CC*) used in this study to assess the prediction and generalization performance of the GPSR model specifies the strength and direction of the linear correlation between the experimental (target) and model-predicted K_{c} magnitudes. For perfect correlation, as indicated by $CC = 1.0$, the predicted and target values are identical. The consistent high values of *CC* (close to 1.0) in respect of the predictions of K_c by the GPSR model for both training and test sets indicate its high prediction accuracy and generalization capability.

The *RMSE* is the standard deviation of the residuals (prediction errors), which are a measure of how far the data points are from the regression line. Alternatively, it is a measure of how spread out the residuals are or how concentrated the data is around the line of the best fit. The low values of *RMSE*s (≤ 0.215) relative to the experimental K_{c} range of [0.001998 - 12.4742] further supports the *CC*-based observation of high prediction accuracy and generalization ability possessed by the GP-based K_{c} prediction model.

3.4.4 Development of ANN-based model predicting PCD rate constant

The multilayer perceptron neural network (MLPNN) architecture was used to build an ANN-based PCD rate constant (K_c) prediction model. This model was developed using the same training and test sets as used for the development of the

GP-based model. It was trained using the *error-back-propagation* (EBP) algorithm (Rumelhart et al., 1986) available in *RapidMiner* data-mining suite (Rapid Miner, 2007). The input elements in the ANN-based models signify the principal components defined in equations (3.7) to (3.10) as shown in figure 3.5. A detailed discussion of the origin, functioning, and applications of the MLPNNs is provided in Chapter 2, section 2.3.1.1.

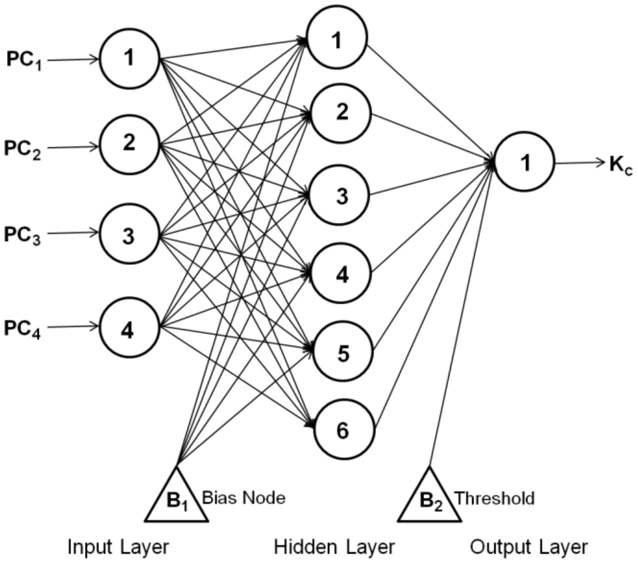


Figure 3.5: The MLPNN used in this study showing inputs and output (K_c)

3.4.4.1 Results and Discussion (ANN-based modeling)

To construct an optimal MLPNN-based model, the effects of network's structural parameters (i.e., the number of hidden layers, number of nodes/neurons in each hidden layer and type of transfer function) and the two EBP algorithm parameters, namely learning rate (η) and momentum coefficient (μ), on the model's prediction accuracy and generalization ability were systematically examined. Also, the effect of random weight initialization was studied to obtain an MLPNN model that corresponds to the global or the deepest local minimum on the model's nonlinear error surface. The prediction accuracy and the generalization performance of the optimal MLPNN-based K_c model have been assessed in terms of *CC* and, *RMSE* magnitudes (refer Table 3.2); these were computed with the target and the corresponding MLPNN-model predicted magnitudes of K_c (listed in Table 3.1). The particulars of the MLPNN architecture, EBP-specific parameter values (η, μ), and the sort of transfer functions used in getting the optimal MLPNN-based K_c prediction model are provided in Table 3.3. Figure 3.6 portrays a parity plot displaying the experimental K_c values and those predicted by the MLPNN-based model. In this plot, there exists a good agreement between the experimental and model-predicted K_c values with a little scatter.

Table 3.3: Structural attributes and EBP algorithm parameters for optimal of MLPNN model

ANN model output	**Number of nodes/neurons in the input layer**	**Number of nodes/neurons in the hidden layer**	**Error Back Propagation Algorithm Parameter**		**Transfer function for hidden nodes/neurons**	**Transfer function in output node/neuron**
			Learning rate (η)	**Momentum (μ)**		
Kc	4	6	0.3	0.2148	Logistic Sigmoid	Identity

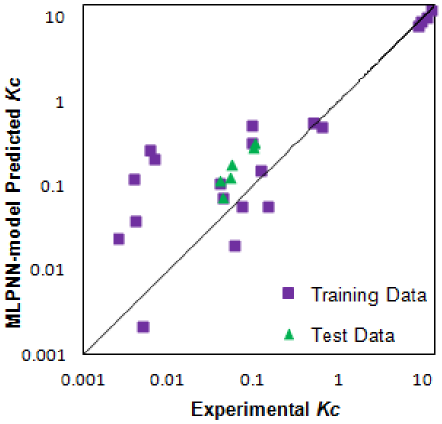


Figure 3.6: Parity Plot of experimental versus an MLPNN-based model of predicted rate constant (K_c)

3.5 OPTIMIZATION OF PCD PROCESS CONDITIONS

In any process, it is important that its operation is optimized to achieve, for example, highest possible conversion/ yield/ selectivity of the desired product/ lowest possible operating temperature/profit or lowest possible energy consumption/ operating cost/selectivity of the undesired product, etc. Accordingly, in the present study, the operation of the PCD process was optimized with a view to maximizing its performance. Specifically, the input space of the GP-based model—which forms the objective function to be maximized —was optimized with a view of maximizing the magnitude of the rate constant of the PCD reaction. The said optimization was conducted using the genetic algorithm stochastic optimization strategy described in-depth in Chapter 2, section 2.5.

Genetic algorithms (GAs) (Goldberg 1989; Holland 1992) is the most widely employed artificial intelligence based stochastic nonlinear search and optimization method based on the natural selection and evolution behavior of biological species. Given an objective function, GA can efficiently and robustly search the decision variable space and obtain optimal values of the decision variables in a manner such that the objective function is maximized/minimized. The strategy is especially well-suited for solving problems involving very large search spaces (Goldberg, 1989; Davis, 1991). The major steps of GA implementation can be briefly described as follows: (a) generation of initial population of candidate solution strings using bits or real values, (b) evaluation of the candidate solutions and assignment of a fitness score to each string in the population, (c) formation of the parent pool of relatively fitter solution strings using a suitable method such as *Roulette wheel selection* (Lipowski and Lipowska, 2012) (d) crossover operation to generate offspring strings, and (e) mutation operation over offspring strings to generate a new population of candidate solution that is usually fitter in terms of fulfilling the optimization objective function maximization/minimization. The GA procedure is iterative wherein the newly formed population of candidate solutions (offspring) is subjected to steps (b) to (e) till convergence is achieved. The above-stated procedure is repeated several times by varying the GA-specific parameter values such as the size of the candidate solution population, and crossover and mutation rates. The solutions obtained thereby were compared, and the one resulting in the highest magnitude of the rate constant (K_c) in respect of the PCD of the specific pharmaceutical pollutant was selected as an overall optimal solution. The salient features, advantages and implementation details of the GA method are described, for example, in Deb and Goldberg (1991) and Nandi et al. (2001).

3.5.1 GA-based optimization of the GP model

The original input space of the GP-based model predicting the rate constant of the PCD of three different pharmaceutical pollutants viz. Ciprofloxacin, Naproxen, and Paracetamol contains four reaction operating variables, viz. pH of the solution (x_1), reaction time (x_2), the degraded concentration of pharmaceutical pollutant (x_3), and the ratio of instantaneous concentration to initial concentration(x_4). The structural attributes of individual pharmaceutical pollutant remain constant as they do not change with the operating conditions. Thus, in reality, only the set of four experimental parameters (x_1 to x_4) termed *decision variables* need to be optimized for maximizing the rate constant of the PCD of a specific pharmaceutical molecule. It must be noted here that while evaluating the objective function (Eq. 3.11) the molecular structural parameters (x_5 to x_{14}) of the specific pharmaceutical molecule undergoing PCD were kept constant.

The GA-based optimization of the input space of the GP-based model was performed using an MS-Excel add-on termed MendelSolve (2018). For each of the three pollutant molecules, optimal PCD conditions were obtained separately. The GA-specific parameters that yielded the three overall best sets of optimized reaction conditions are: *population size* (N_{pop}) = 50, *crossover rate* (R_{cr}) =1, *mutation rate* (R_{mut}) = 0.03125 and the maximum number of generations (N_{gen}) = 100.

Following the above-described GA procedure, five sets of optimized PCD process conditions, which are expected to maximize the photocatalytic degradation rates were obtained. These are listed in Table 3.4. From the tabulated values, the following observations can be made.

- The GA-searched and optimized best set of operating conditions in the case of Ciprofloxacin is capable of improving the extent of PCD from 95.4% (best experimental PCD value listed in Table 3.1) to 99.76 %.

- In the case of Naproxen, the best solution provided by the GA is capable of improving PCD from 95.5% (best experimental PCD value as per Table 3.1) to 99.0 %.
- For Paracetamol, the optimal set of process conditions is capable of increasing the PCD magnitude from 35.07 % (see experimental PCD given in Table 3.1) to 40.0%. The optimized degradation rate in the case of paracetamol is low when compared with that of Ciprofloxacin and Naproxen. The reason for this is that in general the PCD rate for Paracetamol is found to be lower than that for Ciprofloxacin and Naproxen. A detailed account of the degradation of paracetamol is provided by Desale et al. (2013).

It may be noted that the optimized PCD conditions listed in Table 3.4 are simulated by genetic algorithm and these need to be validated experimentally. Accordingly, five PCD experiments were performed using the operating conditions listed in Table 3.4. The table also lists the PCD magnitudes obtained in the validation experiments. From the results of these experiments, it is noted that there exists only a marginal difference between the GA-maximized values (%) of the PCD and those observed in the corresponding validation experiments. The significance of this result is that the GA strategy has been successful in correctly providing the experimental conditions leading to maximization of the PCD of three pharmaceutical pollutants. It also indicates that the GP-based model used in the GA-based optimization has accurately captured the nonlinear relationship existing between the PCD conditions and molecular structural parameters and the PCD rate constant.

Table 3.4: GA-based optimized PCD operating conditions and results of their experimental validation.

Pharmaceutical pollutants	**pH of solution** $\left(x_1^{opt}\right)$	**Reaction Time in Minutes** $\left(x_2^{opt}\right)$	**Degraded concentration of pharmaceuti-cal pollutant** $\left(x_3^{opt}\right)$	**The ratio of instantaneous concentration to the initial concentration** $\left(x_4^{opt}\right)$	**GA Optimized PCD** **(%)**	**Experimentally Validated PCD** **(%)**
Ciprofloxacin	4.5	33.41	1.57	0.007	98.38	98.6
	7.0	58.68	0.227	0.022	99.76	97.52
Naproxen	4.5	98.69	0.98	0.01	99.0	98.61
	7.0	99.45	0.98	0.01	99.0	98.02
Paracetamol	7.0	39.84	64.8	0.6	40.0	41.78

3.6 CONCLUSION

The quality of the ambient water is affected adversely when personal and healthcare products enter the environment in the form of metabolites excreted by the animal and human populations and/or as effluents from hospitals, pharmacies, and pharmaceutical manufacturing facilities. In general, capture or even confinement of these compounds is difficult due to which they pose a serious threat to the aquatic ecosystem and human and animal health. Photocatalytic degradation (PCD) is an effective and relatively inexpensive method, when compared to other approaches for the elimination/reduction of pharmaceutical compounds such as Ciprofloxacin (CFX), Naproxen (NPX) and Paracetamol (PARA) from the wastewater. Earlier, experiments have been conducted involving PCD of the said three pharmaceutical molecules (Shetty et al., 2016). In this study, exclusively data-driven models have been developed using two computational intelligence (CI) based data-driven methods namely Multilayer perceptron neural networks (MLPNN) and genetic programming (GP) for predicting the rate constant (K_c) of the *N*-doped TiO_2 catalyzed photocatalytic degradation (PCD) of CFX, NPX and PARA. The input space of the model consists of the PCD conditions as also the molecular structural parameters of the pollutants. Among the two types of modeling methods, the GP-based model was found to possess superior K_c prediction and generalization performance. In the next step, the GP model was optimized to improve the PCD process. Specifically, a widely used CI-based stochastic optimization method namely genetic algorithm (GA) was used to obtain parameter values of the operating conditions of the PCD process leading to maximization of the rate constant magnitudes and thereby degradation of the three pharmaceutical compounds. Notably, the GA-based optimized conditions were successfully validated by conducting fresh PCD experiments. The remarkable features of the study are as follows: (a) the GP-based model proposed here can be used to assess the efficacy of TiO_2 catalyst in removing pollutants (through the calculation of the rate constant) under different operating conditions, and (b) and the GP/MLPNN based modeling and GA based optimization approach employed in this study can be extended to photocatalytic degradation of other pollutants. The principal advantage of the GP-GA hybrid technique exemplified here is that modeling and optimization can be performed exclusively from the PCD data without invoking the detailed knowledge of physicochemical phenomena underlying the system.

3.7 REFERENCES

Bel E. D., Dewulf J., Witte B. D., Langenhove H. V., Janssen C. (2009). Influence of pH on the sonolysis of ciprofloxacin: biodegradability, eco-toxicity and antibiotic activity of its degradation products, *Chemosphere*, **77(2)**, pp. 291–295.

Belden J.B., Maul J.D., Lydy M.J. (2007). Partitioning and photodegradation of ciprofloxacin in aqueous systems in the presence of organic matter, *Chemosphere*, **66(8)**, pp.1390–1395.

Bhatkhande D.S., Pangarkar V.G., Beenackers A.A. (2001). Photocatalytic degradation for environmental applications, a review, *J. Chemtechnol. Biotechnol.*, **77(1)**, pp. 102–116.

Chakma S., Moholkar V.S. (2015). Intensification of wastewater treatment using sono–hybrid processes: an overview of mechanistic synergism, *Indian Chemical Engineer,* **57(3–4)**, pp.359–381.

Davis L. (1991). *Handbook of genetic algorithms*, 1st Edition, Van Nostrand Reinhold, New York.

Deb K., Goldberg D.E. (1991). *A Comparison of Selection Schemes Used in Genetic Algorithms, Foundations of Genetic Algorithms*, 1st Edition, Morgan Kuafmann Publications, San Mateo, CA, pp. 69–93.

Desale A., Kamble S.P., Deosarkar M.P. (2013). Photocatalytic degradation of paracetamol using TiO_2 photocatalyst, *International Journal of Chemical and Physical Sciences,* **2**, pp.140-148.

Ding T., Lin K., Yang B., Yang M., Li J., Li W., Gan J. (2017). Biodegradation of naproxen by freshwater algae Cymbella sp. and Scenedesmus quadricauda and the comparative toxicity, *Bioresour Technol.*, **238**, pp.164-173. DOI: 10.1016/j.biortech.2017.04.018.

Dodd M.C., Buffle M.O., Von Gunten U. (2006). Oxidation of antibacterial molecules by aqueous ozone: moiety-specific reaction kinetics and application to ozone-based waste water treatment, *Environ Sci Technol*, **40(6)**, pp.1969–77.

Fenton H.J.H. (1894). Oxidation of tartaric acid in presence of iron, *J. Chem. Soc. Transactions,* **65**, pp. 899–911. DOI:10.1039/ct8946500899.

Freeman J.A., Skapura D.M. (1991). *Neural Networks: Algorithms, Applications, and Programming Techniques*, Addison-Wesley: Reading, MA.

Gebhardt W., Schröder H.F. (2007). Liquid chromatography–(Tandem) Mass Spectrometry for the follow-up of the elimination of persistent pharmaceuticals during wastewater treatment applying biological wastewater treatment and advanced oxidation, *J. Chromatography*, **1160(1–2)**, pp. 34–43.

Geladi P., Kowalski B.R. (1986). Partial least-squares regression: A tutorial, *Analytica Chimica Acta*, **185**, pp.1-17.

Goldberg D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning* , 1st Edition, Addison-Wesley, New York.

Holland J.H. (1992). Genetic Algorithms, *Scientific American*, **267(1)**, pp. 66-73.

IBM, SPSS Neural Networks 20 manual, IBM Chicago, 2011.

Kamble S.P., Sawant S.B., Pangarkar V.G. (2006). Photocatalytic degradation of m-dinitrobenzene by illuminated TiO_2 in slurry photoreactor, *J. Chemtechnol. Biotechnol.*, **81(3)**, pp. 365–373.

Koza J.R. (1990). Genetically breeding populations of computer programs to solve problems in artificial intelligence, *Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence*, pp.819–827.

Le Guilloux V., Arrault A., Colliandre L., Bourg S., Vayer P., Morin A. L. (2012). Mining collections of compounds with screening assistant 2., *Journal of Cheminformatics*, **4**, pp.20-29.

Li B., Zhang T. (2010). Biodegradation and Adsorption of Antibiotics in the Activated Sludge Process, *Environmental Biotechnology Laboratory*, **44 (9)**, pp. 3468–3473. DOI: 10.1021/es903490h

Lin Y., Li D., Hu J., Xiao G., Wang J., Li W., Fu X. (2012). Highly efficient photocatalytic degradation of organic pollutants by PANI-modified TiO_2 composite, *The Journal of Physical Chemistry*, **116**, pp. 5764-5772.

Lipowski A., Lipowska D. (2012). Roulette-wheel selection via stochastic acceptance, *Physica A: Statistical Mechanics and its Applications*, **391(6)**, pp. 2193-2196.

MendelSolve, A Genetic Algorithm Solver, URL: http://www.bluestretch.com/mendelsolve/ idex.htm (accessed July 20. 2018).

Mierswa I., Wurst M., Klinkenberg R., Scholz M., Euler T. Y. (2006). Rapid prototyping for complex data mining tasks. *In: Proceeding of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 935–940.

Nandi S., Ghosh S., Tambe S.S., Kulkarni B.D. (2001). Artificial neural-network assisted stochastic process optimization strategies, *AIChE J.* , **47**, pp.126–141.

Patil-Shinde V., Mulani K. B., Donde K., Chavan N. N., Ponrathnam S., Tambe S.S. (2016). The Removal of arsenite [As (III)] and arsenate [As (V)] ions from wastewater using TFA and TAFA resins: Computational intelligence-based reaction modeling and optimization, *Journal of Environmental Chemical Engineering*, **4(4)**, pp. 4275-4286.

Patil-Shinde V., Saha S., Sharma B.K. (2016). High ash char gasification in Thermo-Gravimetric analyzer and prediction of gasification parameters using computational intelligence formalisms, *Chemical Engineering Communications*, **203**, pp.1029-1044.

RapidMiner 7.0 (2007), URL: https://rapidminer.com/products/studio (Accessed December 28, 2017).

Rumelhart D., Hinton G., Williams R. (1986). Learning representations by back propagating error, *Nature*, **323**, pp. 533–536.

Saltelli A., Tarantola S., Chan K.P.S. (1999). A quantitative model-independent method for global sensitivity analysis of model output, *Technometrics*, **41**, pp.39-56.

Schmidt M., Lipson H. (2009). Distilling free-form natural laws from experimental data, *Science*, **324**, pp. 81-85.

Shetty R., Kothari G., Tambe A.S., Kulkarni B.D., Kamble S.P. (2016). Photocatalytic degradation of Ciprofloxacin-HCl using Aeroxide® P-25TiO$_2$ photocatalyst: Comparative evaluation of solar and artificial radiation, *Indian Journal of Chemistry*, **55-A**, pp. 16-22.

Shetty R, Chavan V.B., Kulkarni P.S., Kulkarni B.D., Kamble S.P. (2017). Photocatalytic degradation of pharmaceuticals pollutants using N-doped TiO$_2$ photocatalyst: Identification of CFX degradation intermediates, *Indian Chemical Engineer*, **59(3)**, pp.177-199.

Sun S.P., Guo H.Q., Ke Q. (2009). Degradation of antibiotic ciprofloxacin hydrochloride by photo-Fenton oxidation process, *Environ Eng Sci,* **26(4)**, pp.753–9.

Tchobanoglous G., Burton F., Stensel H.D. (2002). *Wastewater Engineering: Treatment and Reuse*, Metcalf & Eddy Inc., McGraw-Hill Education, 1848 pages.

Van Doorslaer X., Demeestere K., Heynderickx P.M., Van Langenhove H., Dewulf J. (2011). UV-A and UV-C induced photolytic and photocatalytic degradation of aqueous ciprofloxacin and moxifloxacin: reaction kinetics and role of adsorption, *Applied Catalysis B: Environmental*, **101(3–4)**, pp.540–547. DOI: 10.1186/1758-2946-4-20.

Vapnik V. (1995). *The nature of statistical learning theory*, Springer Verlag, New York.

Vyas R., Goel P., Tambe S.S. (2015). *Genetic programming applications in chemical sciences and engineering*, Handbook of Genetic programming applications, Springer International Publishing, pp 99–140. DOI: 10.1007/978-3-319-20883-1.

Zhang L., Hu J., Zhu R., Zhou Q., Chen J. (2013). Degradation of paracetamol by pure bacterial cultures and their microbial consortium, *Appl Microbiol Biotechnol.*, **97(8)**, pp.3687-98. DOI: 10.1007/s00253-012-4170-5.

CHAPTER 4

PREDICTION OF CETANE NUMBER OF BIOFUELS USING ARTIFICIAL INTELLIGENCE BASED MODELS

Abstract

An important parameter that indicates the quality of fuels—especially that of the liquid biofuels—is known as *Cetane Number* (*CN*). It measures the ignition delay time and quality of combustion in a compression-ignition engine. The determination of *CN* experimentally consists of a tedious and time-consuming procedure, which may also yield inaccurate results. To overcome the said difficulty, a number of linear and nonlinear models have been constructed for the prediction of *CN* of biofuels. These models use the *Fatty Acid Methyl Esters* (FAME) composition and fuel properties as predictors (inputs). Towards developing models possessing improved and high prediction accuracy and generalization ability, the present study reports new models that have been developed using two *artificial intelligence* (AI) based state-of-the art data-driven formalisms, namely *genetic programming* (GP), and *artificial neural networks* (ANNs). The notable characteristics of these models are: (a) to our knowledge, which is based on a rigorous literature survey, the biodiesel property based nonlinear *CN* prediction models have been developed for the first time, and (b) the novel GP method, which by itself chooses a linear or a nonlinear data fitting model and its parameters has been employed for the first time for developing the *CN* prediction models. The said property of the GP formalism assists in understanding the inherent dependencies between the predictors and the output (*CN* magnitude) of the model. In the present study, two types of predictors have been used in the model development namely, (i) FAME composition, and (ii) biofuel properties (*density, flash point, higher heating value,* and *kinematic viscosity*). The results of the modeling indicate that (a) the models using the FAME composition in their input space exhibit an excellent *CN* prediction accuracy [*correlation coefficient* (*CC*) > 0.9 and *root mean squared error* (*RMSE*) < 5.00] and generalization performance, and (b) the fuel property-based models exhibit a reasonably good prediction accuracy and generalization performance [*CC* > 0.8 and *RMSE* < 4], albeit inferior than the FAME based models. The AI-based models introduced here due to their excellent prediction and generalization performance have a real potential to replace the existing models for predicting the *Cetane Number* of biofuels.

4.1 INTRODUCTION

A large amount of hazardous gases that are emitted from the exhaust of automobiles has severely intensified the air pollution—causing a significant change in the environment (Bamgboye and Hansen, 2008). Thus, it has become essential to mitigate the amounts of carbon dioxide and other harmful gases in the environment by reducing the usage of petroleum products and shifting the focus to an alternative energy in the form of "*biofuels"*. Biodiesel forms a major proportion of biofuels. It is derived from the vegetable oils or animal fats via the trans-esterification reaction in the presence of an alkali catalyst. There are various other routes to biodiesel production including the biochemical route involving enzymatic transesterification (Kumar et al., 2013). In this process, the fats or the triglycerides are converted into alkyl mono esters of the fatty acid by the reaction with an alcohol (methanol/ ethanol) in the presence of alkali-catalysts or enzymes, producing glycerol as a secondary product. The properties of the biodiesel are strongly dependent upon the specific raw material used in its production. In terms of chemistry, it is much less complex than the conventional petroleum-based diesels (Lopes et al., 2008). Moreover, about eleven percent of the biodiesel (is terms of oxygen by weight) blended with the petro-diesel is shown to result in a complete combustion, due to the higher Cetane number of the blend which makes the combustion smoother; also, the engine is less noisy (Tat and Gerpen, 2003). Thus, the biodiesel is an eco-friendly, alternative diesel fuel that can be manufactured from the domestic renewable resources such as the vegetable oils (edible or non-edible) and animal fats. It is used in running the diesel engines in cars, buses, trucks, construction equipments, boats, and generators, and in-home heating units. The huge popularity of the biodiesel worldwide is due to its much-desired properties such as renewability, high biodegradability, fewer emissions, and higher Cetane number and flash point than the conventional diesel fuel (Knothe et al., 2003). Various types of edible and non-edible oils from, for example, rice bran, coconut, Jatropha curcas, castor, cotton seed, Karanja, and Mahua are available for the commercial biodiesel production. Biodiesel can be used as a fuel in automobiles in the pure form or blended with conventional petro-diesel without any major modifications.

4.2 OBJECTIVE AND MOTIVATION OF THE PRESENT WORK

It is well-known that the fuel properties of the biodiesel, namely, *Cetane number* (*CN*), *density* (*D*), *kinematic viscosity* (*KV*), and *higher heating value* (*HHV*), play a significant role in its utilization in the combustion process (Agarwal et al., 2010; Bajpai and Tyagi, 2006; Creton et al., 2010; Sivaramakrishnan and Ravikumar, 2012; Verduzco et al., 2012). The *CN* is not only the most important indicator of the performance of the internal combustion engine but also represents the dimensionless parameter for the diesel ignition quality. Specifically, it measures the readiness of the fuel to auto-ignite when injected into the engine. The carbon chain length and the degree of unsaturation are the two significant factors of fatty acid methyl esters that affect *CN* of the biodiesel (Tong, 2011). It has also been reported that there exists a relation between the *CN* and other biodiesel properties, namely, *KV*, *D*, *HHV* and *flash point* (*FP*) (Sivaramakrishnan and Ravikumar, 2012).

The standards prescribed for the cetane number determination are: ASTM D613 (in the USA), and *International Organization for Standardization* (ISO) standard ISO 5165, in other countries (Knothe, 2005). Experimental procedure involved in the *CN* determination is complex, tedious and time-consuming; it may also yield inaccurate results necessitating repetition. It is thus necessary that fast and reliable methods are available for monitoring the quality of the biodiesel for its consumption in the internal combustion (IC) engine. In this respect, a mathematical model/correlation capable of predicting accurately the *CN* magnitudes is expected to be of significant utility. Therefore, the main goal of this work is to develop the state-of-the-art artificial intelligence (AI) based mathematical models for the prediction of the cetane number of biodiesels. Here, the *CN* prediction is performed based on two types of inputs (predictors) namely, the composition of fatty acid methyl esters (FAME)(saturated and unsaturated) and physical properties of the biodiesel. The specific inputs used in the development of the models are:

(I) FAME composition – *Capric* (C10:0, x_1), *Lauric* (C12:0, x_2), *Myristic* (C14:0, x_3), *Palmitic* (C16:0, x_4) *Palmitoleic* (C16:01, x_5), *Stearic*

(C18:0, x_6), *Oleic* (C18:01, x_7), *Linoleic* (C18:02, x_8) *Linolenic* (C18:03, x_9), *Arachidic* (C20:0, x_{10}), *Paullinic* (C20:01, x_{11}), *Erucic* (C22:01, x_{12}). Here, the first number in the Carbon ratio for each of the FAME composition shown in bracketed value represents the number of carbon atoms present in the carbon chain and the second number shows the number of unsaturated bonds (usually double bonds), which often introduces a kink in the carbon chain (Kumar et al., 2013). In an unsaturated FAME composition, double bonds play a vital role in the stability of biodiesel as they significantly affect the combustion characteristics of the biodiesel. A higher degree of unsaturation (higher number of double bonds present in the carbon chain) leads to a longer ignition delay, higher soot formation, higher NO_x emissions, and greater heat release rates so as to significantly decrease the *CN* magnitudes of biodiesel (Benjumea et al., 2011). On the contrary, higher *CN* values have been reported in case of high molecular fatty acids (for example, palmitate and stearate fatty acids) with more saturated molecules. Thus, increase in carbon value in chain length with more saturated molecules increases CN value of the biodiesel (Bamgboye and Hansen, 2008).

(II) Properties of biofuels: *density* (kg/l) (*D*), flash *point* (°C) (*FP*), *higher heating value* (MJ/kg) (*HHV*) and *kinematic viscosity* (mm^2/s) (*KV*).

For developing the AI-based models, two datasets consisting of the FAME composition and fuel properties of 171 and 67 biodiesel samples, respectively, and the corresponding *CN*s were compiled from various research articles (Ahmad et al., 2014; Ashwanikumar and Sharma, 2008; Atabani et al., 2013; Deka and Basumatary, 2011; Demirbas, 2009; Ejeh and Aderemi 2014; Eryilmaz et al., 2016; Giakoumis, 2013; Jung et al., 2006; Isigigur et al., 1994; Kaya et al., 2009; Martı´nez et al., 2014; Menga et al., 2008; Miraboutalebi et al., 2016; Onga et al., 2011; Peterson et al., 1997; Saloua et al., 2010; Saydut et al., 2008; Sivaramakrishnan and Ravikumar, 2012; Tong et al., 2011; Winayanuwattikun et al., 2008; Yahya and Stephen, 1994). These datasets are tabulated in Tables 4.A and 4.B in Appendix 4 at the end of this Chapter.

In this work, a novel AI-based exclusively data-driven formalism namely, *genetic programming* (GP) (Koza, 1992) has been used for the first time in the prediction of *CN*. The novelty of this method is it searches and optimizes both the configuration (form) and the related parameters of an appropriate linear or nonlinear data-fitting function without making any assumptions regarding the form and parameters. Additionally, the most widely used ANN architecture namely multi-layer perceptron (MLP) neural network has been used to develop the biodiesel *CN* predicting nonlinear models.

The details of GP and MLP neural networks including their implementation and applications in chemical engineering/technology are presented in sections 2.3.3 and 2.3.1.1, respectively.

4.3 RESULTS AND DISCUSSION

4.3.1 Principal Component analysis

Principal Component Analysis (PCA) is essentially a multivariate technique (Geladi and Kowalski, 1986) that (a) analyzes and extracts the most important information in the form of linearly uncorrelated variables from a given dataset, and (b) simplifies and compresses the size of the dataset (Abdi and Williams, 2010). This extracted information is represented in terms of a set of new orthogonal variables called *principal components* (PCs), which are linear combinations of the original variables. Commonly, fewer PCs than the original dimension of the dataset capture a large amount of variance in the data; this allows usage of the lesser number of PCs as inputs than the number of original input variables thus effectively reducing the dimensionality the input data set. Such a dimensionality reduction decreases the computational load involved in the model development as also avoids data redundancy.

In this study, PCA was performed on the twelve-dimensional FAME composition dataset with a view to reduce the dimensionality of the input space of the GP and MLP-based models. The quantum of the variance addressed by the twelve principal components (PCs) are: *PC₁* (16.6%), *PC₂* (13.4%), *PC₃* (10.5%), *PC₄* (9.3%), *PC₅* (8.5%), *PC₆* (8.2%), *PC₇* (8.1%), *PC₈* (7.6%), *PC₉* (7.3%), *PC₁₀* (5.9%), *PC₁₁* (4.5%) and *PC₁₂* (0.1%). From these values, it is seen that (a) except *PC₁₁* and *PC₁₂*, there is no drastic variation between the values of the variance captured by the successive *PCs* indicating no major redundancy in the data, and (b) the first nine *PCs* have cumulatively captured a relatively large percentage (≈ 89.5%) of the data variance. It is thus possible to use the first nine PCs as inputs for developing the GP and MLP based *CN* predicting models. This serves to reduce the input space of the models from twelve to nine dimensions. The PCA-transformed nine PCs (model inputs) are defined below.

$$PC_1 = 0.394\hat{x}_1 + 0.548\hat{x}_2 + 0.34\hat{x}_3 - 0.237\hat{x}_4 + 0.036\hat{x}_5 - 0.248\hat{x}_6 - 0.51\hat{x}_7 - 0.023\hat{x}_8 + 0.127\hat{x}_9 - 0.119\hat{x}_{10} + 0.089\hat{x}_{11} + 0.106\hat{x}_{12} \quad (4.1)$$

$$PC_2 = -0.245\hat{x}_1 - 0.271\hat{x}_2 - 0.197\hat{x}_3 + 0.037\hat{x}_4 + 0.022\hat{x}_5 - 0.346\hat{x}_6 - 0.324\hat{x}_7 + 0.636\hat{x}_8 + 0.347\hat{x}_9 - 0.264\hat{x}_{10} + 0.002\hat{x}_{11} + 0.078\hat{x}_{12} \quad (4.2)$$

$$PC_3 = -0.129\hat{x}_1 - 0.136\hat{x}_2 - 0.195\hat{x}_3 - 0.553\hat{x}_4 + 0.051\hat{x}_5 + 0.217\hat{x}_6 - 0.076\hat{x}_7 - 0.105\hat{x}_8 + 0.274\hat{x}_9 + 0.368\hat{x}_{10} + 0.439\hat{x}_{11} + 0.388\hat{x}_{12} \quad (4.3)$$

$$PC_4 = -0.261\hat{x}_1 - 0.128\hat{x}_2 + 0.251\hat{x}_3 + 0.393\hat{x}_4 + 0.152\hat{x}_5 - 0.332\hat{x}_6 - 0.041\hat{x}_7 - 0.261\hat{x}_8 - 0.28\hat{x}_9 - 0.124\hat{x}_{10} + 0.461\hat{x}_{11} + 0.435\hat{x}_{12} \quad (4.4)$$

$$PC_5 = 0.068\hat{x}_1 + 0.036\hat{x}_2 + 0.008\hat{x}_3 + 0.089\hat{x}_4 - 0.962\hat{x}_5 - 0.064\hat{x}_6 + 0.055\hat{x}_7 + 0.078\hat{x}_8 - 0.016\hat{x}_9 + 0.054\hat{x}_{10} + 0.158\hat{x}_{11} + 0.142\hat{x}_{12} \quad (4.5)$$

$$PC_6 = -0.338\hat{x}_1 - 0.114\hat{x}_2 + 0.439\hat{x}_3 + 0.042\hat{x}_4 - 0.128\hat{x}_5 + 0.222\hat{x}_6 - 0.047\hat{x}_7 - 0.365\hat{x}_8 + 0.572\hat{x}_9 - 0.244\hat{x}_{10} - 0.291\hat{x}_{11} + 0.07\hat{x}_{12} \quad (4.6)$$

$$PC_7 = 0.284\hat{x}_1 + 0.066\hat{x}_2 - 0.443\hat{x}_3 + 0.128\hat{x}_4 + 0.02\hat{x}_5 + 0.394\hat{x}_6 - 0.044\hat{x}_7 - 0.09\hat{x}_8 - 0.03\hat{x}_9 - 0.565\hat{x}_{10} - 0.071\hat{x}_{11} + 0.459\hat{x}_{12} \quad (4.7)$$

$$PC_8 = -0.028\hat{x}_1 + 0.003\hat{x}_2 + 0.091\hat{x}_3 - 0.121\hat{x}_4 + 0.019\hat{x}_5 - 0.208\hat{x}_6 + 0.092\hat{x}_7 + 0.113\hat{x}_8 - 0.186\hat{x}_9 + 0.302\hat{x}_{10} - 0.647\hat{x}_{11} + 0.605\hat{x}_{12} \quad (4.8)$$

$$PC_9 = -0.288\hat{x}_1 - 0.046\hat{x}_2 + 0.374\hat{x}_3 - 0.415\hat{x}_4 - 0.064\hat{x}_5 + 0.357\hat{x}_6 - 0.068\hat{x}_7 + 0.315\hat{x}_8 - 0.507\hat{x}_9 - 0.328\hat{x}_{10} + 0.047\hat{x}_{11} - 0.018\hat{x}_{12} \quad (4.9)$$

where $\hat{x}_r; r = 1,2,\dots,12,$ denote the normal scores (standardized variables) of the twelve input variable values (x_r). The input variables were normalized using 'Z-score' method.

$$\hat{x}_r^i = \frac{x_r^i - \bar{x}_r}{\sigma_r}; i = 1,2,\dots,N_p \quad (4.10)$$

where x_r^i denotes the ith value of the rth non-normalized input variable, x_r; N_p represents the number of observations/ samples in the data set; $\bar{x}_r$ and σ_r refers to the mean and standard deviation of rth input variable computed using the corresponding values $\{x_r^i\}, i = 1\ ,2,\dots,N_p$ in the original example data set. The mean and standard deviation values used in the normalization procedures are given in Eqn. (4.11) and (4.12) respectively.

$$\bar{x}_1 = 0.042;\ \bar{x}_2 = 5.169;\ \bar{x}_3 = 3.396;\ \bar{x}_4 = 12.89;\ \bar{x}_5 = 1.095;\ \bar{x}_6 = 8.56;$$
$$\bar{x}_7 = 36.21;\ \bar{x}_8 = 23.53;\ \bar{x}_9 = 4.086;\ \bar{x}_{10} = 1.151;\ \bar{x}_{11} = 1.21; \bar{x}_{12} = 1.146 \quad (4.11)$$

$$\sigma_1 = 0.358;\ \sigma_2 = 18.41;\ \sigma_3 = 11.87;\ \sigma_4 = 12.82; \sigma_5 = 8.31;\ \sigma_6 = 12.41;$$
$$\sigma_7 = 20.92; \sigma_8 = 22.2;\ \sigma_9 = 12.3;\ \sigma_{10} = 5.09;\ \sigma_{11} = 8.91;\ \sigma_{12} = 8.60 \quad (4.12)$$

Similar to the FAME composition variables, the *CN* magnitudes were also normalized using the Z-score method and the corresponding mean and standard deviation values are 53.59 and 9.36, respectively. For developing the models possessing good *CN* prediction accuracy and generalization ability, the PCA-transformed nine-dimensional experimental data set was split randomly, wherein 75% data were used as the training set for developing the models while 25% data were used as the test set for assessing the generalization ability of the models.

In this study, the biofuel physical property data consisting of the magnitudes of the *density* (*D*), *flash point* (*FP*), *higher heating value* (*HHV*) and *kinematic viscosity* (*KV*) were not subjected to PCA since the four-dimensional input space formed by the stated properties was sufficiently small (=4) and therefore did not require PCA-based dimensionality reduction.

4.3.2 Development of GP-based models

The two types of GP-based models using (a) the PCs derived from the FAME composition data, and (b) the fuel properties as inputs were developed using *Eureqa Formulize* (EF) software package (Schmidt and Lispon, 2009). A noteworthy attribute of the EF package is that it has been optimized to construct models with lower complexity (parsimonious structure) possessing good generalization capability. For obtaining an overall best model, the GP-algorithm specific parameters were varied systematically and rigorously. The prediction accuracy and the generalization performance of each GP-based model were evaluated in terms of the *coefficient of correlation* (*CC*) and *root mean squared error* (*RMSE*) between the experimental and the corresponding model-predicted *CN* values. The *CC* and *RMSE* values were computed separately for the training and test datasets. The criterion for selecting an overall best model was high (low) and comparable *CC* (*RMSE*) magnitudes in respect of both the training and test datasets. The GP models I (FAME composition based) and II (biofuel property-based) thus obtained are given below (see Equations 4.13 and 4.14). The corresponding magnitudes of the training and the test set *CC*s and *RMSE*s are listed in Table 4.1. The parity plots pertaining to the experimental (desired) vis-à-vis model-predicted *CN* values in respect of the predictions by GP-model I and II are depicted in Figures 4.1 and 4.2, respectively.

Table 4.1: Comparison of *CN* prediction accuracy and generalization performance of the AI-based models

Model inputs	Model	Training set		Test set	
		CC	***RMSE***	***CC***	***RMSE***
• FAME composition	GP- I	0.923	3.59	0.930	3.56
	MLP- I	0.962	2.58	0.936	3.39
• Fuel properties	GP- II	0.810	2.82	0.910	2.27
	MLP- II	0.817	2.82	0.811	3.70

❖ FAME composition based optimal model (GP-model I)

$$y_1 = 0.456PC_4 + 0.375PC_7 + 0.21PC_1 + 0.142PC_6 + 0.049PC_5 - 0.463PC_2 - 0.276PC_2PC_3 \qquad (4.13)$$

❖ Biodiesel fuel properties-based optimal model (GP-model II)

$$y_1 = 0.392KV^2 + 0.427KV \times HHV \times FP + \frac{-0.00738}{FP-0.211} + \frac{A}{(0.392+(A)^2-0.282KV)} - 0.392 \qquad (4.14)$$

where $A = 0.280 + KV \times HHV - D - HHV \times D$ (4.15)

From the structure of the GP-model I, it is seen that it uses only the first seven principal components ignoring PC8 and PC9. The probable reason for the said omission is that the magnitudes of the variance in the FAME data captured by PC8 and PC9 are relatively small (7.6% and 7.3 %, respectively) and therefore, the respective PCs were ignored by the GP algorithm.

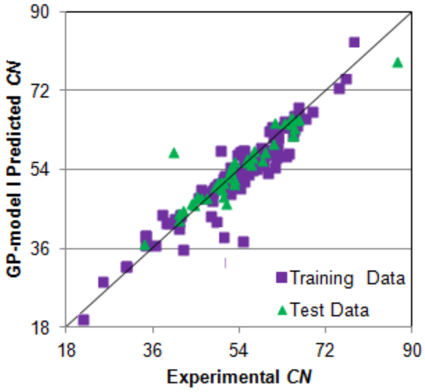


Figure 4.1: Parity plot of the experimental versus GP-Model I predicted values of *CN*.

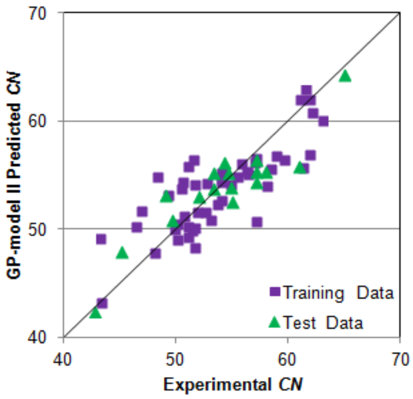


Figure 4.2: Parity plot of the experimental versus GP-Model II predicted values of *CN*.

4.3.3 Development of MLP-based models

Similar to the GP-models I and II, MLP neural network based models were developed using *RapidMiner* data mining package (Mierswa et al., 2006). The same training and test data sets as used in building the two GP models were employed for constructing the MLP-based models I and II. These models were trained by the error-back-propagation algorithm (Rumelhart et al., 1986). Towards securing optimally performing models, the influence of MLP neural network's two structural parameters (the number of hidden layers and number of nodes/neurons in each hidden layer) and two EBP algorithm specific parameters (*learning rate* and *momentum coefficient*) on the model's pertaining to prediction and generalization performance was systematically examined. The details of the MLP's structural and training algorithm specific that led to optimal models I and II are given in **Table 4.2.**

Table 4.2: Structural and training algorithm specific details of MLP-based *CN* prediction models

Model	***No. of hidden layers***	N_I	N_H	***Learning rate, η***	***Momentum coefficient, μ***	**TF*** **(hidden layer)**	**TF*** **(output Layer)**
• MLP- I	1	9	6	0.2416	0.1602	Logistic Sigmoid	Linear
• MLP- II	1	4	5	0.300	0.1005	Logistic Sigmoid	Linear

TF*: Transfer function; N_I, N_H: Number of input and hidden nodes

The *CC* and *RMSE* values pertaining to the predictions of MLP models I and II are listed in Table 4.1. From these values, it is seen that overall the prediction and generalization performance of FAME composition based models (i.e., GP-model I and MLP-model I) is excellent, while that of the biofuel property based models (GP-model II and MLP-model II) is reasonably good. Among the two FAME composition-based models, the performance of MLP-model I is marginally better than that of the GP-model I. Being the best model among the four AI-based ones, the MLP-model I has been selected to prepare the parity plot (see Figure 4.3) of the experimental versus model predicted *CN* magnitudes. In this figure it is seen that the MLP model has fitted the training and test data excellently with very little scatter.

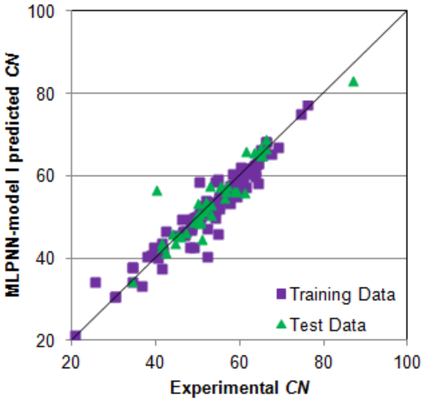


Figure 4.3: Parity plot of the desired versus MLP model I predicted *CN* magnitudes

4.3.4 Comparision of *CN*-based models

A comparison of the AI-based models using FAME composition as input with their currently available counterparts proposed by Bamgboye and Hansen (2008), Borroto et al. (2014), Gopinath et al. (2009), Rodriguez et al. (2013) and Valdes et al. (1998) reveals that the AI-based models presented in this work using data of 171 biodiesel samples have yielded high and comparable prediction and generalization performance (see Table 4.3 below). It may however be noted that the FAME-based models for the prediction of *CN* of biodiesel presented in this study are based upon much higher number (=171) of biodiesel samples compared to the number of samples (51, 57, 38, 48, and 128) considered by the existing linear and nonlinear FAME-based models listed in **Table 4.3.** Since, the FAME-based AI models presented in this study are based on higher number of data, their applicability is wider compared to the existing models.

Table 4.3: A representative comparison of *CN* predicting correlations/models

SN	Authors	Type of Model	Model inputs	Sample size	Statistical analysis# of model predictions
1	Bamgboye and Hansen (2008)	Linear Regression	FAME composition	51	*CC* = 0.921 Std. Error= 4.3
2	Agarwal et al. (2010)	Linear Regression	Biodiesel fuel properties	10	*CC* = 0.99 R^2 = 0.98
3	Gopinath et al. (2009)	Linear Regression	FAME composition	57	*CC = 0.92,* R^2 = 0.85 *Prediction error = 8 %*
4	Borroto et al. (2014)	ANN	FAME composition	38	*CC* = 0.97 *MAE* = 3 %
5	Rodriguez et al. (2013)	Back-Propagati	FAME compositi	48	*CC = 0.94,* R^2 = 0.91

		on Neural Network	on		
6	Valdes et al. (1998)	ANN	FAME composition	128	*CC = 0.92* R^2 = 0.82 *RMSE* = 11.4 %
7	Sivaramakrishnan and Ravikumar (2012)	Linear Regression	Biodiesel fuel properties	26	*CC* = 0.9 R^2 = 0.81 *RMSE* = 7.21 %

[#] *CC*: Correlation coefficient; MSE: mean squared error; *RMSE*: Root mean squared error; R^2: Variance

A notable characteristic of GP-model II and MLP-model II is that these have been developed using a bigger data set (67 biodiesel samples) compared to the property based linear models by Sivaramakrishnan and Ravikumar (2012) (26 biodiesel samples) and Agarwal et al. (2010) (10 biodiesel samples). Thus, the stated GP and MLP models have wider applicability than their linear counterparts. Additionally, the property-based GP and MLP models possess better *CN* prediction accuracy and generalization performance than the existing linear models.

4.4 CONCLUSION

Cetane number is an important parameter for the determination of the ignition quality of the biodiesel. Experimental determination of this number is complex, time consuming and often yields inaccurate results. In this study, two exclusively data-driven artificial intelligence-based formalisms, namely, genetic programming (GP) and multilayer perceptron (MLP) neural networks were employed to develop models predicting *CN* of biodiesels. The GP and MLP methods were individually utilized to develop two types of *CN* prediction models, which respectively used fatty acid methyl esters (FAME) composition and biodiesel fuel properties as their inputs. The results of this study indicate that the FAME composition-based GP and MLP models pertaining to better prediction and generalization performance than the biodiesel property-based models. Among the two FAME composition-based models, the MLP model has yielded marginally superior performance than the GP-based model. Since they are constructed using much larger number of biodiesel data, the FAME based AI models presented in this chapter possess wider applicability than the currently available linear and nonlinear *CN* prediction models for the biodiesel.

4.5 REFERENCES

Abdi H., Williams L.J. (2010). Principal component analysis, *Wiley Interdiscip. Rev. Comput. Stat.*, **2**, pp. 433-459.

Agarwal M., Singh K., Chaurasia S.P. (2010). Prediction of Biodiesel properties form fatty acid composition using linear rregression and ANN techniques, *Indian Chemical Engineer*, **52(4)**, pp. 347-361.

Bamgboye A.I.,Hansen A.C. (2008). Prediction of cetane number of biodiesel fuel from the fatty acid methyl ester (FAME) composition, *Int. Agrophysics*, **22**, pp. 21-29.

Bajpai D., Tyagi V.K. (2006). Biodiesel: Source, Production, Composition, Properties and Its Benefits, *Journal of Oleo Science*, **55(10)**, pp. 487-502.

Benjumea P., Agudelo J.R., Agudelo A.F. (2011). Effect of the Degree of Unsaturation of Biodiesel Fuels on Engine Performance, Combustion Characteristics, and Emissions, *Energy Fuels*, **25**, pp. 77–85.

Borroto Y.Z., Rodriguez R.P., Errasti M., Sierens R., Verhelst S. (2014). Prediction of cetane number and ignition delay of biodiesel using Artificial Neural Networks, *Energy Procedia*, **57**, pp. 877 – 885.

Creton B., Dartiguelongue C., Bruin T., Toulhoat H. (2010). Prediction of the Cetane Number of Diesel Compounds Using the Quantitative Structure Property Relationship, *Energy Fuels*, **24**, pp. 5396–5403. DOI:10.1021/ef1008456

Demirbas A. (2009). Progress and recent trends in biodiesel fuels, *Energ. Convers. Manage*, **50**, pp. 14-34.

Geladi P., Kowalski B.R. (1986). Partial least squares regression:A tutorial, *Analytica Chimica Acta*, **185**, pp. 1-17.

Giakoumis E.G. (2013). A statistical investigation of biodiesel physical and chemical properties, and their correlation with the degree of unsaturation, *Renew. Energ.*,**50**, pp. 858-878.

Gopinath A., Puhan S., Nagarajan G. (2009). Relating the cetane number of biodiesel fuels to their fatty acid composition: a critical study, *Proc I MechE Part D J Autom Engg*, **223**, pp. 565-583.

Jung H., Kittelson D. B., Zachariah M. R. (2006). Characteristics of SME biodiesel-fueled diesel particle emissions and the kinetics of oxidation, *Environ. Sci. Technol.*, **40**, pp. 4949-4955.

Kaya C., Hamamci C., Baysal A., Akba O., Erdogan S. , Saydut A. (2009). Methyl ester of peanut (Arachis hypogea L.) seed oil as a potential feedstock for biodiesel production, *Renew. Energ.*, **34**, pp. 1257-1260.

Knothe G., Matheaus A.C. , Ryan T.W.(2003). Cetane numbers of branched and straight-chain fatty esters determined in an ignition quality tester, *Fuel*, **82**, pp. 971-975.

Knothe G.(2005). Dependence of biodiesel fuel properties on the structure of fatty acid alkly esters, *Fuel Process. Technol.*,**86**, pp. 1059-1070.

Koza J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, A Bradford Book, The MIT Press, Cambridge, Massachusetts.

Kumar G., Srivastava R., Singh R. (2013). Exploring Biodiesel: Chemistry, Biochemistry, and Microalgal Source, *International Journal of Green Energy*, **10(8)**, pp. 775-796.

Lopes J.C.A., Boros L., Krähenbühl M.A., Meirelles A.J.A., Daridon J.L., Pauly J., Marrucho I.M., Coutinho J.A.P. (2008). Prediction of Cloud Points of Biodiesel, *Energy & Fuels*, **22**, pp.747–752.

Martı´nez G., Sa´nchez N.,Encinar J.M., Gonza´lez J.F.(2014). Fuel properties of biodiesel from vegetable oils and oil mixtures- Influence of methyl esters distribution, *Biomass Bioenerg.*,**63**, pp. 22-32.

Mierswa I., Wurst M., Klinkenberg R., Scholz M., Euler T. (2006). YALE: Rapid prototyping for complex data mining tasks, *In: Proceeding of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 935–940.

Miraboutalebi S.M.R., Kazemi P., Bahrami P. (2016). Fatty acid methyl ester (FAME) composition used for estimation of biodiesel cetane number employing random forest and artificial neural networks: A new approach, *Fuel*,**166**, pp. 143-151.

Peterson C.L., Reece D.L., Hammond B.L., Thompson J., Beck S.M. (1997). Processing, characterization, and performance of eight fuels from lipids, *Appl. Eng. Agric.*,**13**, 71-79.

Rumelhart D., Hinton G.,Williams R. (1986). Learning representations by back propagating error, *Nature*, **323**, pp. 533-536.

Rodriguez R.P., Borroto Y.Z., Lapuerta M., Pérez L.G., Verhelst S. (2013). Prediction of the cetane number of biodiesel using artificial neural networks and multiple linear regression, *Energy Conversion and Management*, **65**, pp. 255–261.

Saydut M.Z., Duz C., Kaya A.B., Kafadar, Hamamci C. (2008). Transesterified sesame (Sesamum indicum L.) seed oil as a biodiesel fuel, *Bioresource Technol.*, **99**, pp. 6656-6660.

Schmidt M., Lispon H. (2009). Distilling free-form natural laws from experimental data, *Science*, **324**, pp. 81-85.

Sivaramakrishnan K., Ravikumar P.(2012). Determination of cetane number of biodiesel and it’s influence on physical properties, *J. Eng. Appl. Sci.*,**7**, pp. 205-211.

Tat M.E., Gerpen J.H.V.(2003). Fuel Property Effects on Biodiesel, *The Society for engineering in Agriculture, Food, and Biological Systems*, Paper Number: **036034**, pp. 1-16. DOI: 10.13031/2013.15048

Tong D.,Hu C.,Jiang K., Li Y..(2011). Cetane number predictions of biodiesel from the compoisition of the Fatty Acid Methyl Esters, *J. Am. Oil Chem. Soc.*, **88**, pp. 415-423.

Valdés C.I.R., Verduzco L.F. R., Hernández J.A. (2015). Artificial neural network models to predict density, dynamic viscosity,and cetane number of biodiesel, *Fuel*, **147**, pp. 9–17.

Verduzco L.F.R., Rodriguez J.E.R., Jacob A. R.J. (2012). Peredicting Cetane Number, Kinematic viscosity, Density and higher heating value of biodiesel from its fatty acid methyl ester composition, *Fuel*, **91**, pp. 102-111.

Appendix 4

Table 4.A: Experimental *CN* and FAME composition data set

Types of Biodiesel sources [Carbon chain length: double bonds]	**Capric [10:00]** (x_1)	**Lauric [12:00]** (x_2)	**Myristic [14:00]** (x_3)	**Palmitic [16:00]** (x_4)	**Palmitoleic [16:01]** (x_5)	**Stearic [18:00]** (x_6)	**Oleic [18:01]** (x_7)	**Linoleic [18:02]** (x_8)	**Linolenic [18:03]** (x_9)	**Arachidic [20:00]** (x_{10})	**Paullinic [20:01]** (x_{11})	**Erucic [22:01]** (x_{12})	**Cetane Number** **(*CN*)**	**Reference[§]**
Annona reticulate Linn	0	0	1	17.2	42	7.5	48.4	21.7	0	0	0	0	53.47	Tong et al. (2011)
Mesua ferrea Linn	0	0	0.9	10.8	0	12.4	60	15	0	0.9	0	0	55.1	Tong et al. (2011)
Moringa oleifera Lam	0	0	0	9.1	2.1	2.7	79.4	0.7	0.2	0	0	0	56.66	Miraboutalebi et al. (2016)
Pterygota alata Rbr	0	0	0	14.5	0	8.5	44	32.4	0	0	0	0	51.09	Miraboutalebi et al. (2016)
Vallaris solanacea Kuntze	0	0	0	7.2	0	14.4	35.3	40.4	0	1.8	0	0	50.26	Miraboutalebi et al. (2016)
Mappia foetida Milers	0	0	0	7.1	0	17.7	38.4	36.8	0	0	0	0	50.7	Miraboutalebi et al. (2016)
Terminalia	0	0	0	19.7	0	2.4	37.3	39.8	0	0.6	0	0	49.6	Miraboutalebi

chebula Retz													et al. (2016)
Ziziphus mauritiana Lam	0	0	0	10.4	0	5.5	64.4	12.4	0	1.8	2.6	1.7	55.37	Miraboutalebi et al. (2016)
Joannesia princeps Vell	0	0	2.4	5.4	0	0	45.8	46.4	0	0	0	0	45.2	Miraboutalebi et al. (2016)
Lauric	0	100	0	0	0	0	0	0	0	0	0	0	61.4	Miraboutalebi et al. (2016)
Myristic	0	0	100	0	0	0	0	0	0	0	0	0	66.2	Miraboutalebi et al. (2016)
Palmitic	0	0	0	100	0	0	0	0	0	0	0	0	74.5	Miraboutalebi et al. (2016)
Stearic	0	0	0	0	0	100	0	0	0	0	0	0	86.9	Miraboutalebi et al. (2016)
Oleic	0	0	0	0	0	0	100	0	0	0	0	0	55	Miraboutalebi et al. (2016)
Palmitoleic	0	0	0	0	100	0	0	0	0	0	0	0	51	Miraboutalebi et al. (2016)
Linoleic	0	0	0	0	0	0	0	100	0	0	0	0	42.2	Miraboutalebi et al. (2016)
Erucic	0	0	0	0	0	0	0	0	0	0	0	100	76	Miraboutalebi et al. (2016)
Eicosanoic	0	0	0	0	0	0	0	0	0	0	100	0	64.8	Miraboutalebi et al. (2016)

Linolenic	0	0	0	0	0	0	0	0	100	0	0	0	20.4	Miraboutalebi et al. (2016)
Soybean	0	0	0.1	10.5	0.1	3.7	23.2	48.9	1.2	0	0.3	0.1	47.2	Miraboutalebi et al. (2016)
Inedible tallow	0	0.1	2.1	23.9	2.8	19.5	38.5	6.4	0.3	0	0.5	0.1	61.7	Miraboutalebi et al. (2016)
Moringa oleifera Lam	0	0	0	9.1	2.1	2.7	79.4	0.7	0.2	0	0	0	56.7	Miraboutalebi et al. (2016)
Pongamia pinnata P.	0	0	0	10.6	0	6.8	49.4	19	0	0	2.4	0	55.8	Miraboutalebi et al. (2016)
Vallaris solanacea K.	0	0	0	7.2	0	14.4	35.3	40.4	0	0	0	0	50.3	Miraboutalebi et al. (2016)
Aleurites moluccana	0	0	0	5.5	0	6.7	10.5	48.5	28.5	0	0	0	34.2	Miraboutalebi et al. (2016)
Euphorbia helioscopia L	0	2.8	5.5	9.9	0	1.1	15.8	22.1	42.7	0	0	0	34.2	Miraboutalebi et al. (2016)
Garnicia morella D.	0	0	0	0.7	0	46.4	49.5	0.9	0	0	0	0	63.5	Miraboutalebi et al. (2016)
Actinodaphne angust.	0	87.9	1.9	0.5	0	5.4	0	0	0	0	0	0	63.2	Miraboutalebi et al. (2016)
Melia azadirach Linn	0	0	0.1	8.1	1.5	1.2	20.8	67.7	0	0	0	0	41.4	Miraboutalebi et al. (2016)
Myristica	0	0	39.2	13.3	0	2.4	44.1	1	0	0	0	0	61.8	Miraboutalebi

malabarica L														et al. (2016)
Canola	0	0	0.1	5.2	0.2	2.5	58.1	28.1	0.4	0	1.6	0.4	55	Miraboutalebi et al. (2016)
Lard	0	0.1	1.9	24.5	2.8	14.4	38.3	13.4	0.3	0	0.7	0.1	63.6	Miraboutalebi et al. (2016)
Yellow grease	0	0	1.1	17.3	2.2	9.5	45.3	14.5	1.3	0	1.3	0	52.9	Miraboutalebi et al. (2016)
Linseed	0	0	0	5	0	2	20	18	55	0	0	0	52	Miraboutalebi et al. (2016)
Wild mustard	0	0	0.1	2.6	0.2	0.9	7.8	14.2	13	0	5.4	45.7	61.1	Miraboutalebi et al. (2016)
Waste palm oil	0	0	1	39	0.2	4.3	43.7	10.5	0.2	0	0.2	0	60.4	Miraboutalebi et al. (2016)
Balanites roxburhii	0	0	0	17	4.3	7.8	32.4	31.3	7.2	0	0	0	50.5	Miraboutalebi et al. (2016)
Garnicia echinocarpa	0	0	0	3.7	0	43.7	52.6	0	0	0	0	0	63.1	Miraboutalebi et al. (2016)
Neolitsea umbrosa G.	0	59.1	11.5	0	0	0	21	6.7	0	0	0	0	60.8	Miraboutalebi et al. (2016)
Broussoneti ap. Vent.	0	0	0	4	0	6.1	14.8	71	1	0	0	0	41.2	Miraboutalebi et al. (2016)
Salvadora oleoiles D.	0	35.6	50.7	4.5	0	0	8.3	0.1	0	0	0	0	66.1	Miraboutalebi et al. (2016)
Nephelium	0	0	0	2	0	13.8	45.3	0	0	34.7	4.2	0	64.9	Miraboutalebi

L.														et al. (2016)
Ziziphus maurit. L.	0	0	0	10.4	0	5.5	64.4	12.4	0	0	2.6	1.7	55.4	Miraboutalebi et al. (2016)
Jojoba	0	0	0	1.2	0	0	10.7	0.1	0.4	0	59.5	12.3	69	Miraboutalebi et al. (2016)
Peanut	0	0	0.1	8	0	1.8	53.3	28.4	0.3	0	2.4	0	53	Miraboutalebi et al. (2016)
Grape	0	0	0.1	6.9	0.1	4	19	69.1	0.3	0	0	0	48	Miraboutalebi et al. (2016)
Sunflower	0	0	0	6	0.1	2.9	17	74	0	0	0	0	49	Miraboutalebi et al. (2016)
Sample Mix-X	0	0	0	11.53	0	13.36	60.67	0.62	0	0	0	0	57	Miraboutalebi et al. (2016)
Sample Mix-Y	0	0	0	40	0	5	45	7	0	0	0	0	50	Miraboutalebi et al. (2016)
Sample Mix-Z	0	0	0	23.93	0	19.54	38.54	6.43	0.32	0	0	0	54	Miraboutalebi et al. (2016)
Rapeseed	0	0	1	3.5	0	0.9	64.1	22.5	8	0	0	0	46	Miraboutalebi et al. (2016)
Rubber seed	0	0	0.2	12.5	0	8.3	27.8	37.7	13.4	0	0	0	51	Miraboutalebi et al. (2016)
Cottonseed	0	0.1	1	20.1	0	2.6	19.2	55.2	0.6	0	0	0	52.1	Miraboutalebi et al. (2016)
Jatropha	0	0	0.1	15.6	0	10.5	42.1	30.9	0.2	0	0	0	54	Miraboutalebi et al. (2016)

Karanja	0	0	0.1	9.9	0	7.8	53.2	19.1	0	0	0	0	52	Miraboutalebi et al. (2016)
Jatropha: palm 50:50	0	0.1	0.5	28.1	0	7.7	42.7	20.1	0.2	0	0	0	59	Miraboutalebi et al. (2016)
Neem	0	0.8	0.5	18.2	0	20.1	41.3	16.4	0.3	0	0	0	58.7	Miraboutalebi et al. (2016)
Palm	0	0.2	0.8	39.5	0	5.1	43.1	10.4	0.1	0	0	0	64	Miraboutalebi et al. (2016)
Mahua	0	0	0.2	20.8	0	25.2	36.4	15.8	0.3	0	0	0	61.4	Miraboutalebi et al. (2016)
Sunflower: coconut 50:50	0	20.3	10.5	9.3	0	4.3	19.4	32.6	0	0	0	0	54.6	Miraboutalebi et al. (2016)
Beef tallow	0	0.1	2.5	23.3	0	19.4	42.4	2.9	0.9	0	0	0	58.8	Miraboutalebi et al. (2016)
Jatropha: coconut 50:50	0	20.9	10.4	13.7	0	7.2	26.1	18.2	0.1	0	0	0	58	Miraboutalebi et al. (2016)
Coconut	0	45.6	22.1	10.2	0	3.6	8.2	2.7	0	0	0	0	60	Miraboutalebi et al. (2016)
Cannabis sativa Linn	0	0	0	0	0	0	15	65	15	0	0	0	36.4	Miraboutalebi et al. (2016)
Perilla frutescens Britton	0	0	0	0	0	0	9.8	47.5	36.2	0	0	0	30.1	Miraboutalebi et al. (2016)

Joannesia princeps Vell	0	0	2.4	5.4	0	0	45.8	46.4	0	0	0	0	45.2	Miraboutalebi et al. (2016)
Illicium verum Hook	0	0	4.4	0	0	7.9	63.2	24.4	0	0	0	0	50.7	Miraboutalebi et al. (2016)
Moringa concanensis Nimmo	0	0	0	9.7	0	2.4	83.8	0.8	0	0	0	0	56.3	Miraboutalebi et al. (2016)
Quassia indica Nooleboom	0	0	0	9	0	0	36	48	0	0	0	0	46.7	Miraboutalebi et al. (2016)
Terminalia chebula Retz	0	0	0	19.7	0	2.4	37.3	39.8	0	0	0	0	49.6	Miraboutalebi et al. (2016)
Ziziphus mauritiana Lam	0	0	0	10.4	0	5.5	64.4	12.4	0	0	0	0	55.4	Miraboutalebi et al. (2016)
Pterygota alata Rbr	0	0	0	14.5	0	8.5	44	32.4	0	0	0	0	51.1	Miraboutalebi et al. (2016)
Princepia utilis Royle	0	0	1.8	15.2	0	4.5	32.6	43.6	0	0	0	0	48.9	Miraboutalebi et al. (2016)
Vallaris solanacea Kuntze	0	0	0	7.2	0	14.4	35.3	40.4	0	0	0	0	50.3	Miraboutalebi et al. (2016)
Mappia foetida	0	0	0	7.1	0	17.7	38.4	0	36.8	0	0	0	50.7	Miraboutalebi et al. (2016)

Milers														
Mesua ferrea Linn	0	0	0.9	10.8	0	12.4	60	15	0	0	0	0	55.1	Miraboutalebi et al. (2016)
Aegle marmelos correa Roxb	0	0	0	16.6	0	8.8	30.5	36	8.1	0	0	0	48.3	Miraboutalebi et al. (2016)
Jatropa curcas Linn	0	0	1.4	15.6	0	9.7	40.8	32.1	0	0	0	0	52.3	Miraboutalebi et al. (2016)
Meyna laxiflora Robyns	0	0	0	18.8	0	9	32.5	39.7	0	0	0	0	50.4	Miraboutalebi et al. (2016)
Annonareticulata Linn	0	0	1	17.2	0	7.5	48.4	21.7	0	0	0	0	53.5	Miraboutalebi et al. (2016)
Mimusops hexendra Roxb	0	0	0	19	0	14	63	3	0	0	0	0	59.3	Miraboutalebi et al. (2016)
Terminalia bellirica Roxb	0	0	0	35	0	0	24	31	0	0	0	0	56.2	Miraboutalebi et al. (2016)
Thevetia peruviana Merrill	0	0	0	15.6	0	10.5	60.9	5.2	7.4	0.3	0	0	57.48	Miraboutalebi et al. (2016)
Canarium commune Linn	0	0	0	29	0	9.7	38.3	21.8	1.2	0	0	0	55.58	Miraboutalebi et al. (2016)

Celastrus paniculatus Linn	0	0	0	25.1	0	6.7	46.1	15.4	3	0	0	0	51.9	Miraboutalebi et al. (2016)
Vernonia cinerea Less	0	0	8	23	0	8	32	22	0	3	0	0	57.51	Miraboutalebi et al. (2016)
Putranjiva roxburghii	0	0	0	8	0	15	56	18	0	3	0	0	54.99	Miraboutalebi et al. (2016)
Calophyllum apetalum Wild	0	0	0	8	0	14	48	30	0	0	0	0	51.57	Miraboutalebi et al. (2016)
Calophyllum inophyllum Linn	0	0	0	17.9	2.5	18.5	42.7	13.7	2.1	0	0	0	57.3	Miraboutalebi et al. (2016)
Azadirachta indica	0	0	0	14.9	0	14.4	61.9	7.5	0	1.3	0	0	57.83	Tong et al. (2011)
Pongamia pinnata Pierre	0	0	0	10.6	0	6.8	49.4	19	0	4.1	2.4	0	55.84	Tong et al. (2011)
Sapindus trifoliatus Linn	0	0	0	5.4	0	8.5	55.1	8.2	0	20.7	0	0	59.77	Miraboutalebi et al. (2016)
Mimusops hexendra Robx	0	0	0	19	0	14	63	3	0	1	0	0	59.32	Miraboutalebi et al. (2016)

Holoptelia integrifolia	0	0	3.5	35.1	1.9	4.5	53.3	0	0	1.1	0	0	61.22	Tong et al. (2011)
Balanites roxburghii Planch	0	0	0	17	4.3	7.8	32.4	31.3	7.2	0	0	0	50.46	Miraboutalebi et al. (2016)
Aphanamixis polystachya Park	0	0	0	23.1	0	12.8	21.5	29	13.6	0	0	0	48.52	Miraboutalebi et al. (2016)
Meyna laxiflora Robyns	0	0	0	18.8	0	9	32.5	39.5	0	0	0	0	50.42*	Miraboutalebi et al. (2016)
Aleurites moluccana Wild	0	0	0	5.5	0	6.7	10.5	48.5	28.5	0	0	0	34.18	Miraboutalebi et al. (2016)
Euphorbia helioscopia Linn	0	2.8	5.5	9.9	0	1.1	15.8	22.1	42.7	0	0	0	34.25	Miraboutalebi et al. (2016)
Garcinia echinocarpa Thw	0	0	0	3.7	0	43.7	52.6	0	0	0	0	0	63.1	Miraboutalebi et al. (2016)
Garcinia morella Desr	0	0	0	0.7	0	46.4	49.5	0.9	0	2.5	0	0	63.52	Miraboutalebi et al. (2016)
Saturega hortensis	0	0	0	0.4	0	0.4	12	18	62	0	0	0	25.46	Tong et al. (2011)

Linn														
Actinodaphne angustifolia	4.3	87.9	1.9	0.5	0	5.4	0	0	0	0	0	0	63.2	Miraboutalebi et al. (2016)
Litsea glutinosa Robins	0	96.3	0	0	0	0	2.3	0	0	0	0	0	64.79	Tong et al. (2011)
Neolitsea cassia Linn	0.3	85.9	3.8	0	0	0	4	3.3	0	0	0	0	64.05	Tong et al. (2011)
Neolitsea umbrosa Gamble	1.7	59.1	11.5	0	0	0	21	6.7	0	0	0	0	60.77	Miraboutalebi et al. (2016)
Swietenia mahagoni Jacq	0	0	0	9.5	0	18.4	56	0	16.1	0	0	0	52.26	Miraboutalebi et al. (2016)
Anamirta cocculus Wight & Hrn	0	0	0	6.1	0	47.5	46.4	0	0	0	0	0	64.26	Miraboutalebi et al. (2016)
Broussonetia papyrifera Vent	0	0	0	4	0	6.1	14.8	71	1	3	0	0	41.25*	Miraboutalebi et al. (2016)
Argemone Mexicana	0	0	0.8	14.5	0	3.8	18.5	61.4	0	1	0	0	44.45	Miraboutalebi et al. (2016)
Salvadora	0.8	35.6	50.7	4.5	0	0	8.3	0.1	0	0	0	0	66.13	Miraboutalebi

oleoides Decne														et al. (2016)
Salvadora persica Linn	0.1	19.6	54.5	19.6	0	0	5.4	0	0	0	0	0	67.47*	Miraboutalebi et al. (2016)
Nephelium lappaceum Linn	0	0	0	0.2	0	13.8	45.3	0	0	34.7	4.2	0	64.86	Miraboutalebi et al. (2016)
Madhuca butyracea Mac	0	0	0	66	0	3.5	27.5	3	0	0	0	0	65.27	Miraboutalebi et al. (2016)
Madhuca indica JF Gmel	0	0	0.1	17.8	0	14	46.3	17.9	0	3	0	0	56.61	Miraboutalebi et al. (2016)
Rhus succedanea Linn	0	0	0	25.4	0	0	46.8	27.8	0	0	0	0	52.22	Miraboutalebi et al. (2016)
Ervatamia coronaria Stapf	0	0	0	24.4	0.2	7.2	50.5	15.8	0.6	0.7	0.2	0	56.33	Miraboutalebi et al. (2016)
Basella rubra Linn	0	0	0.4	19.7	0.4	6.5	50.3	21.6	1.1	0	0	0	54	Miraboutalebi et al. (2016)
Corylus avellana	0	0	0	3.2	3.1	0	2.6	88	2.9	0	0	0	54.5	Tong et al. (2011)
Jatropa curcas Linn	0	0	1.4	15.6	0	9.7	40.8	32.1	0	0.4	0	0	52.31	Tong et al. (2011)

Moringa concanensis Nimmo	0	0	0	9.7	0	2.4	83.8	0.8	0	3.3	0	0	56.32*	Miraboutalebi et al. (2016)
Croton tiglium Linn	0	0	11	1.2	0	0.5	56	29	0	2.3	0	0	49.9	Miraboutalebi et al. (2016)
Princeptia utilis Royle	0	0	1.8	15.2	0	4.5	32.6	43.6	0	0	0	0	48.94	Miraboutalebi et al. (2016)
Aegle marmelos correa Roxb	0	0	0	16.6	0	8.8	30.5	36	8.1	0	0	0	48.3*	Miraboutalebi et al. (2016)
APME	0	0	0	18.4	0.3	11.8	18.3	26.7	23.2	0.5	0.2	0.6	44*	Miraboutalebi et al. (2016)
PaME	0	0.1	1	42.8	0	4.5	40.5	10.1	0.2	0	0	0	54.6*	Miraboutalebi et al. (2016)
SME	0	0.1	0.1	10.2	0	3.7	22.8	53.7	8.6	0.3	0	0.1	37.9*	Miraboutalebi et al. (2016)
Manilkara zapotaseed oil	0	0	0	13.27	0	2.8	64.15	17.92	1.86	0	0	0	52*	Miraboutalebi et al. (2016)
SBO	0	0	0	5.93	0.04	30.47	54.26	7.8	0.66	0.51	0	0	58.23*	Miraboutalebi et al. (2016)
SBOB	0	0	0	17.49	0.5	6.72	48.54	22.2	0.77	0.34	0	0	54.88*	Miraboutalebi et al. (2016)
Kenaf methyl	0	0	0	17.5	0.5	1.7	24.5	41.3	0.4	0.4	0	0	54*	Miraboutalebi et al. (2016)

esters														
Nephelium L.	0	0	0	2	0	13.8	45.3	0	0	34.7	4.2	0	64.9*	Miraboutalebi et al. (2016)
Perilla frutescens Britton	0	0	0	0	0	0	9.8	47.5	36.2	0	0	0	30.1*	Miraboutalebi et al. (2016)
Madhuca butyracea M.	0	0	0	66	0	3.5	27.5	3	0	0	0	0	65.3*	Miraboutalebi et al. (2016)
Melia azadirach Linn	0	0	0.1	8.1	1.5	1.2	20.8	67.7	0	0	0	0	41.37	Miraboutalebi et al. (2016)
Myristica malabarica Lam	0	0	39.2	13.3	0	2.4	44.1	1	0	0	0	0	61.85	Miraboutalebi et al. (2016)
Urtica dioica Linn	0	0	0	9	0	0	14.6	73.7	2.7	0	0	0	38.73	Miraboutalebi et al. (2016)
Tectona grandis Linn	0	0	0.2	11	0	10.2	29.5	46.4	0.4	0	2.3	0	48.31	Miraboutalebi et al. (2016)
Garcinia combogia Desr	0	0	0	2.3	0	38.3	57.9	0.8	0.4	0	0.3	0	61.5*	Miraboutalebi et al. (2016)
Garcinia indica Choisy	0	0	0	2.5	0	56.4	39.4	1.7	0	0	0	0	65.16	Miraboutalebi et al. (2016)

Rape	0	0	0	4.9	0	1.6	33	20.4	7.8	0	9.3	23	55*	Miraboutalebi et al. (2016)
Broussonetia papyriferaVent	0	0	0	4	0	6.1	14.8	71	1	0	0	0	41.3	Miraboutalebi et al. (2016)
Myristica malabarica Lam	0	0	39.2	13.3	0	2.4	44.1	1	0	0	0	0	61.8	Miraboutalebi et al. (2016)
Anamirta cocculus Wight & Hrn	0	0	0	6.1	0	47.5	46.4	0	0	0	0	0	64.26	Miraboutalebi et al. (2016)
Peanut	0	0.02	0	11.91	0.08	2.99	39.99	40.69	0	1.28	0	3.19	48.86	Winayanuwattikun et al. (2008)
Coconut	0	64.44	20.45	7.71	0.09	1.73	4.61	0.96	0	0.04	0	0	65.8	Winayanuwattikun et al. (2008)
Rice bran	0	0.01	0.41	20.01	0.09	1.7	41.7	34.26	1.1	0.57	0	0.18	50.09	Winayanuwattikun et al. (2008)
Palm	0	0.59	0.96	38.67	0.11	3.32	45.45	10.87	0.2	0.23	0	0.02	59.11	Winayanuwattikun et al. (2008)

Soybean	0	0.01	0.04	10.7	0.02	3.02	24.02	56.58	5.35	0.11	0	0.19	42.21	Winayanuwattikun et al. (2008)
Sunflower	0	0	0.08	6.59	0.09	3.14	22.43	66.2	0.92	0.15	0	0.43	41.41	Winayanuwattikun et al. (2008)
Corn	0	0	0.02	13.17	0.11	2.46	35.19	47.97	0.61	0.37	0	0.13	46.3	Winayanuwattikun et al. (2008)
Cashew nut	0	0	0.07	10.36	0.19	9.04	63.38	16.17	0.29	0.67	0	0.07	54.03	Winayanuwattikun et al. (2008)
Black sesame	0	0	0	9.28	0.03	4.77	35.49	49.54	0.31	0.47	0	0.14	45.91	Winayanuwattikun et al. (2008)
White Sesame	0	0.02	0.02	9.01	0.07	4.62	41.14	44.38	0.23	0.42	0	0.11	46.92	Winayanuwattikun et al. (2008)
Almond	0	0.4	0.19	6.3	0.4	1.13	68.74	22.63	0.22	0	0	0	50.54	Winayanuwattikun et al. (2008)
Canola	0	0.06	0.07	4.65	0.13	1.64	65.93	21.16	5.16	0.93	0	0.26	52.98	Winayanuwattikun et al. (2008)
Safflower	0	0.02	0.11	6.44	0.06	2.2	14.13	76.57	0.15	0.2	0	0.15	39.32	Winayanuwatti

														kun et al. (2008)
Olive	0	0.03	0.02	11.37	0.63	2.58	80.46	4.17	0.56	0.21	0	0.01	55	Winayanuwatti kun et al. (2008)
Wild almond	0	55.55	38.19	2.55	0.44	0.22	2.63	0.44	0	0	0	0	66.13	Winayanuwatti kun et al. (2008)
Mapok	0	0.14	1.55	23.56	0.14	8.82	43.38	19.55	0.04	0.58	0	2.24	56.35	Winayanuwatti kun et al. (2008)
White Silk Cotton	0	0.06	0.16	24.19	0.21	4.15	26.3	42.03	1.73	0.39	0	0.69	49.52	Winayanuwatti kun et al. (2008)
Rubber	0	0.3	0.42	9.87	0.22	5.28	21.76	45.82	16.12	0.14	0	0.09	40.29	Winayanuwatti kun et al. (2008)
Castor bean	0	0.13	0.21	10.56	1.44	9.54	29.71	41.26	3.3	3.79	0	0.06	48.32	Winayanuwatti kun et al. (2008)
Physic nut	0	0.14	0.17	14.82	0.81	4.15	40.98	38.61	0.27	0.06	0	0	48.91	Winayanuwatti kun et al. (2008)
Tobacco	0	0	0	10.52	2.67	11.01	74.99	0.81	0	0	0	0	40.1	Winayanuwatti kun et al.

														(2008)
Pomelo	0	0.01	0.14	25.24	0.17	4.25	24.53	43.32	2	0.27	0	0.11	49.29	Winayanuwattikun et al. (2008)
Papaya	0	0.26	0.46	17.12	0.45	2.98	72.91	4.83	0.29	0.67	0	0.07	56.27	Winayanuwattikun et al. (2008)
Rambutan	0	0.08	0.11	8.77	0.96	7.25	55.25	3.72	0.26	22.05	0	1.34	61.17	Winayanuwattikun et al. (2008)
Pumpkin	0	0.01	0.18	20.53	0.07	6.51	38.68	32.96	0.22	0.28	0	0.6	51.87	Winayanuwattikun et al. (2008)
Tangerine	0	0.01	0.03	21.67	0.39	4.03	20.99	48	4.45	0.24	0	0.05	46.48	Winayanuwattikun et al. (2008)
Duea-kai	0	0.5	0.42	26.19	0.11	10.23	46.89	14.68	0.23	0.7	0	0.07	57.35	Winayanuwattikun et al. (2008)

where * Test data

Appendix 4

Table 4.B: Experimental *CN* versus the biodiesel fuel properties data set

Types of Biodiesel sources	**Kinematic Viscosity (mm^2/s) (*KV*)**	**Heating value (MJ/Kg) (*HHV*)**	**Flash Point (°C) (*FP*)**	**Density (kg/L) (*D*)**	**Cetane Number (*CN*)**	**Reference[§]**
Waste Cooking	4.75	39.805	161.7	0.8806	56.2	Giakoumis (2013)
Soybean	4.04	39.8	165	0.8857	42.6	Giakoumis (2013)
Cottonseed	4.7	40.48	165.4	0.879	53.3	Giakoumis (2013)
Soybean Methyl Ester	3.891	39.77	188	0.886	54.8[*]	Peterson et al. (1997)
Sesame Seed Oil	4.2	40.4	170	0.8672	50.48	Saydut et al. (2008)
Chicken fat	4.81	39.89	162.2	0.8763	57	Giakoumis (2013)
Olive	5.05	40.28	171	0.8812	58.9	Giakoumis (2013)
Biodiesel from waste cooking oil	5.3	42.65	195.85	0.897	54	Demirbas (2009)
Methyl ester of Peanut seed oil	4.42	40.1	166	0.8485	53.59	Kaya (2009)
Soyabean	4.5	41.28	178	0.885	45[*]	Giakoumis (2013)
Yenice safflower seed methyl ester	4.07	40.06	180	0.8878	49.8	Isigigur et al. (1994)
High Oleic Sunflower	4.74	40.47	167	0.8766	53.2	Martı´nez et al. (2014)
Rapeseed	4.2	41.55	80	0.882	54	Sivaramakrishnan and Ravi kumar (2012)
Safflower	4.1	40.155	169.9	0.8838	51.8	Giakoumis (2013)
Cynara	4.66	39.95	175	0.8857	43.2[*]	Martı´nez et al.

Cardunculus						(2014)
Methyl Tallow ester	4.994	39.961	187.8	0.878	61.8	Yahya and Stephen (1994)
Rapeseed	4.42	40.32	177	0.8796	48.3	Martı´nez et al. (2014)
Waste Cooking oil	4.23	32.9	171	0.89	54.5	Menga et al. (2008)
Calophyllum inophyllum biodiesel	4	41.397	140	0.869	57*	Onga et al. (2011)
Canola Methyl Ester	4.754	39.9	163	0.881	57.9	Peterson et al. (1997)
Hazelnut	4.3	39.8	163.5	0.8779	53.8	Giakoumis (2013)
Neem	4.72	39.96	162.5	0.8762	54.2*	Giakoumis (2013)
Hazelnut oil Methyl ester	4.945	40.198	169	0.88	51.43	Eryilmaz et al. (2016)
Methyl Soyoil Ester	4.18	39.823	190.6	0.887	49.6	Yahya and Stephen (1994)
Beef tallow	4.83	40.04	157.2	0.8743	60.9	Giakoumis (2013)
Maclura Methyl ester	4.66	40.05	180	0.889	48	Saloua et al. (2010)
Soybean	4.29	40.02	158.8	0.8829	51.9	Giakoumis (2013)
Rubber Seed oil FAME	3.89	39.7	152	0.885	54	Ahmad et al. (2014)
Jatropha	4.72	40.38	158.5	0.8787	55.7*	Giakoumis (2013)
Midwest Biofuels methyl soyate	3.9	39.61	185	0.885	58.4	Peterson et al. (1997)
Jatropha Biodiesel	4.8	39.23	135	0.862	57	Onga et al. (2011)
Methyl ester of jatropha Oil	5.65	38.45	170	0.88	50	Ashwanikumar and Sharma (2008)
Rapseed Ethyl ester	6.17	40.51	185	0.876	64.9	Peterson et al. (1997)

Peanut	4.9	41.71	176	0.883	54	Sivaramakrishnan and Ravi kumar (2012)
Jatropha Biodiesel (Assam India)	9.02	44.844	100	0.8981	43.3	Barua (2011)
Nicotiana tabacum (tobacco)	4.23	39.81	165.4	0.885	51.6*	Atabani et al. (2013)
Jatropha Curcas	4.46	40.32	188	0.8816	49.2*	Martı´nez et al. (2014)
Shea butter Biodiesel	4.3	37.247	96	0.87305	46.8374	Ejeh and Aderemi (2014)
Croton	4.48	40.28	174.5	0.8832	50.6	Giakoumis (2013)
Rapeseed	4.63	40.335	164.4	0.8822	54.1*	Giakoumis (2013)
Pongamiapinnata (karanja)	4.85	35.56	180	0.89	58*	Atabani et al. (2013)
Linseed	4.06	40.41	170.3	0.8915	51.3	Giakoumis (2013)
Brassica Carinata	5.31	40.13	177	0.8969	46.4	Martı´nez et al. (2014)
Rice bran oil Methyl ester	4.81	41.38	156.85	0.872	51.6	Peterson et al. (1997)
Rice bran	4.7	40.475	157.8	0.8809	56.3	Giakoumis (2013)
Rubber seed	4.79	40.35	158.3	0.8823	50.4	Giakoumis (2013)
Coconut	2.78	38.985	127.7	0.8708	61	Giakoumis (2013)
Corn	4.32	40.19	165.7	0.8822	52.5	Giakoumis (2013)
Yellow Oleander Methyl ester	4.33	44.986	75	0.875	61.5	Deka and Basumatary (2011).
Rapseeed Methyl ester	5.65	40.54	179	0.8802	61.8*	Peterson et al. (1997)
Canola	4.4	39.975	159	0.8816	54.8	Giakoumis (2013)

Karanja	5.04	40.275	163.6	0.8829	55.4*	Giakoumis (2013)
Sunflower	4.6	41.33	96	0.86	49	Sivaramakrishnan and Ravi kumar (2012)
Terminalia catappa	4.3	36.97	90	0.873	57.1*	Atabani et al. (2013)
Jatropha curcas L.	4.4	41.17	163	0.88	57.1*	Atabani et al. (2013)
Babassu	3.6	41.15	127	0.875	63*	Sivaramakrishnan and Ravi kumar (2012)
Madhuca indica (mahua)	3.98	39.4	129	0.916	51	Atabani et al. (2013)
Soy Methyl ester	4.891	36.921	160	0.883	50*	Jung et al. (2006)
Jatropha oil Methyl ester	4.84	37.2	185.85	0.88	51.6	Banapurmatha et al. (2008)
Palm	4.61	39.985	161.9	0.8747	61.2*	Giakoumis (2013)
Soybean Ethyl Ester	4.493	39.96	171	0.881	52.7	Peterson et al. (1997)
Palm	5.7	41.24	183	0.88	62*	Sivaramakrishnan and Ravi kumar (2012)
Canola Ethyl Ester	4.892	40.03	177	0.878	59.6	Peterson et al. (1997)
Peanut	4.77	39.93	174.5	0.8829	54.9	Giakoumis (2013)
Mahua	5.06	40.18	150.6	0.8745	56.9	Giakoumis (2013)
Sunflower	4.55	40.42	161	0.8786	51	Martı´nez et al. (2014)
Fish	4.3	40.55	162.6	0.8873	51	Giakoumis (2013)
Terminalia belerica Robx.	5.17	39.22	90	0.8828	53	Atabani et al. (2013)

* Test data

REFERENCES

Ahmad J., Yusup S., Bokhari A., Nik R., Kamil M. (2014). Study of fuel properties of rubber seed oil-based biodiesel, *Energy Conversion and Management*, **78**, pp. 266–275.

Ashwanikumar, Sharma S. (2008). An evaluation of multipurpose oil seed crop for industrial uses (Jatropha curcas L.): A review, *Industrial Crops and Products*, **28 (1)**, pp.1-10.

Atabani A.E., Silitonga A.S., Ong H.C., Mahlia T.M.I., Masjuki H.H., Badruddin I.A., Faya H. (2013). Non-edible vegetable oils: A critical evaluation of oil extraction, fatty acid compositions, biodiesel production, characteristics, engine performance and emissions production, *Renewable and Sustainable Energy Reviews*, **18**, pp. 211–245.

Banapurmatha N.R., Tewari P.G., Hosmath R.S. (2008). Performance and emission characteristics of a DI compression ignition engine operated on Honge, Jatropha and sesame oil methyl esters, *Renewable Energy,* **23(9)**, pp. 1982-1988.

Barua P.K. (2011). Biodiesel from Seeds of Jatropha Found in Assam, India, *International Journal of Energy, Information and Communications,* **2(1)**, pp. 53-65.

Deka D.C., Basumatary S. (2011). High quality biodiesel from yellow oleander (Thevetia peruviana) seed oil, Biomass and Bioenergy, 35, pp. 1797-1803.

Demirbas A. (2009). Progress and recent trends in biodiesel fuels, *Energy Conversion and Management* , **50,** pp. 14–34.

Ejeh J., Aderemi B. (2014). Production of Biodiesel from Shea Butter Oil using Homogeneous Catalysts, *Leonardo Journal of Sciences*, **24**, pp. 39- 48.

Eryilmaz T., Arslan M., Yesilyurt M.K., Taner A. (2016). Comparison of Empirical Equations and Artificial Neural Network Results in Terms of Kinematic Viscosity Prediction of Fuels Based on Hazelnut Oil Methyl

Ester, *Environmental Progress & Sustainable Energy*, **35(6)**, pp. 1827-1841.

Giakoumis E.G. (2013). A statistical investigation of biodiesel physical and chemical properties, and their correlation with the degree of unsaturation, *Renewable Energy*, **50**, pp. 858-878.

Isigigur A., Karaosmanoglu F., Aksoy H.A., Hamdullahpur F., Gulder O.L. (1994). Performance and Emission Characteristics of a Diesel Engine Operating on Safflower Seed Oil Methyl Ester, *Applied Biochemistry and Blotechnology*, **45(1)**, pp. 93-102.

Jung H., Kittelson D.B., Zachariah M.R. (2006). Characteristics of SME Biodiesel-Fueled Diesel Particle Emissions and the Kinetics of Oxidation, *Environ. Sci. Technol*., **40**, pp. 4949-4955.

Kaya C., Hamamci C., Baysal A., Akba O., Erdogan S., Saydut A. (2009). Methyl ester of peanut (Arachis hypogea L.) seed oil as a potential feedstock for biodiesel production, *Renewable Energy*, **34**, pp. 1257–1260.

Martı´nez G., Sa´nchez N., Encinar J.M., Gonza´lez J.F. (2014). Fuel properties of biodiesel from vegetable oils and oil mixtures. Influence of methyl esters distribution, *Biomass and Bioenergy*, **63,** pp. 22-32.

Menga X., Chena G., Wang Y. (2008). Biodiesel production from waste cooking oil via alkali catalyst and its engine test, *Fuel Processing Technology*, **89**, pp. 851- 857.

Miraboutalebi S.M.R., Kazemi P., Bahrami P. (2016). Fatty Acid Methyl Esters composition used for estimation of biodiesel cetane number employing random forest and artificial neural networks: A new approach, *Fuel*, **166**, pp. 143-151.

Onga H.C., Mahliaa T.M.I., Masjukia H.H., Norhasyima R.S. (2011). Comparison of palm oil, Jatropha curcas and Calophyllum inophyllum for biodiesel: A review, *Renewable and Sustainable Energy Reviews*, **15**, pp. 3501–3515.

Peterson C. L., Reece D. L., Hammond B. L., Thompson J., Beck S. M.(1997). Processing, characterization, and performance of eight fuels from lipids, *Applied Engineering in Agriculture*, **13(1)**, pp.71-79.

Saloua F., Saber C., Hedi Z. (2010). Methyl ester of [Maclura pomifera (Rafin.) Schneider] seed oil: Biodiesel production and characterization, *Bioresource Technology*, **101**, pp. 3091–3096.

Saydut A., Duz M.Z., Kaya C., Kafadar A.B., Hamamci C. (2008). Transesterified sesame (Sesamum indicum L.) seed oil as a biodiesel fuel, *Bioresource Technology*, **99**, pp. 6656–6660.

Sivaramakrishnan K., Ravikumar P. (2012). Determination of cetane number of biodiesel and its influence on physical properties, *ARPN Journal of Engineering and Applied Sciences*, **7(2)**, pp. 205-211.

Tong D., Hu C., Jiang K., Li Y. (2011). Cetane Number Prediction of Biodiesel from the composition of the Fatty Acid Methyl Esters, *J AM Oil Chem Soc*, **88**, pp. 415-423.

Winayanuwattikun P., Kaewpiboon C., Piriyakananon K., Tantong S., Thakernkarnkit W., Chulalaksananukul W., Yongvanich T. (2008). Potential plant oil feed stock for lipase-catalyzed biodiesel production in Thailand, *BIOMASS AND BIOENERGY*, **32**, pp. 1279–1286.

Yahya A., Stephen J. (1994). Physical and chemical characterization of methyl soyoil and methyl tallow esters as CI engine fuels, *Biomass and Bioenergy*, **6(4)**, pp. 321-328.

CHAPTER 5

PREDICTION OF COALS ASH FUSION TEMPERATURES USING COMPUTATIONAL INTELLIGENCE BASED MODELS

Abstract

In the coal-based gasification and combustion processes, the mineral matter (predominantly oxides) contained in the coal, is left as an incombustible residue, termed *ash*. Usually, ash gets deposited on the heat absorbing surfaces of the exposed equipment of the gasification/ combustion processes. These deposits lead to slagging or fouling and, therefore, reduced process efficiency. The *ash fusion temperatures* (AFTs) represent the range of temperature for the ash deposition. Accordingly, for operating, designing and optimizing coal-based processes, it is significant to have mathematical models predicting accurately the four types of AFTs namely *initial deformation temperature, softening temperature, hemispherical temperature,* and *flow temperature*. A number of linear and nonlinear models possessing varied AFT prediction accuracies and complexities are available. Their applicability is limited to a extent as most of coals are originating from a limited number of geographical regions. Consequently, this chapter presents computational intelligence (CI) based nonlinear models to predict the four AFTs wherein the composition of oxides present in the coal ash is used as the model input. The CI methods used in the modeling are *genetic programming* (GP)*, artificial neural networks* (ANN), and *support vector regression* (SVR). The distinguished characteristics of this study are that models with a wider application potential, better AFT prediction accuracy and generalization performance, and reduced complexity, have been developed. Amongst the three types of CI-based models, GP and MLP based models have produced overall improved performance in predicting all four AFTs.

5.1 INTRODUCTION

The major processes that use coal feedstock are gasification, combustion, and liquefaction. Coal is an intricate substance mainly containing carbon, hydrogen, nitrogen, oxygen, sulfur, and mineral matter that can be intrinsic and/or extraneous with widely divergent composition and forms (Ozbayoglu and Ozbayoglu, 2006).

In coal-based processes, its mineral matter experiences a numerous complex physical and chemical transformations. These led to the formation of ash that has a tendency of depositing on the heat-transfer surfaces and other exposed process equipment (Seggiani and Pannocchia, 2003). The phenomena responsible for this ash deposition are termed *slagging* and *fouling*. Slagging produces considerably viscous or fused deposits of ash in zones that are directly exposed to the hottest parts of the boiler (radiant heat exchange). The consequence of fouling is deposition of species in the vapor form and their condensation on the surfaces due to the convective heat exchange. Such things particularly take place in the cooler parts of the boiler at temperatures below the melting point of the bulk coal ash (Seggiani and Pannocchia, 2003). The low temperature fusing coal ash promotes the clinker formation that usually deposits around the heat transfer pipes which result in corrosion of furnace components. It is well-known that ash clinkering may lead to pressure drop, channel burning, and an unstable gasifier operation (Van Dyk et al., 2001). As a result, coal-based thermal power stations have to take periodic shut-downs to eradicate the clinker from the ash recovery shuts and heat transfer pipes (Yin et al., 1998). The occurrence of ash slag flows for example in the *Integrated Gasification Combined Cycle* (IGCC) and other slagging reactors is directly attributed to the creation of the liquid slag and to the stable solid crystalline phases (Patterson and Hurst, 2000; Skrifvars et al., 2004). The characteristic governing the behavior of ash in several coal-utilizing processes is termed *ash fusion temperature* (AFT).

The AFT analysis consists of the determination of four temperatures signifying four stages (phases) in the melting of the ash.

(i) *Initial deformation temperature* (IDT): Temperature at which the ash just begins to flow.

(ii) *Softening temperature* (ST): It is that temperature when the ash softens and becomes plastic.

(iii) *Hemispherical temperature* (HT): Represents the temperature yielding a hemispherically shaped droplet.

(iv) *Fluid temperature* (FT): Refers to the temperature, at which the ash becomes a free-flowing fluid (Slegeir et al., 1988).

The above-described four AFTs own following attributes and applications.

- ❖ Specify the temperature range for a likely formation of deposits on the heat adsorbing surfaces of the process equipment (Ozbayoglu and Ozbayoglu, 2006).
- ❖ Endow with vital clues to the extent to which the ash clinkering and agglomeration are likely to happen within the gasifier/combustor (Alpern et al., 1984; Seggiani, 1999; Van Dyk et al., 2001).
- ❖ Are of specific significance to the process of all types of gasifiers/combustors (Bryers, 1996; Wall et al., 1998). For example, to permit continuous slug tapping, it is essential that the working temperature in the entrained flow gasifier is greater than the flow temperature (Hurst et al., 1996). In the case of fluid-bed gasifiers, AFTs set the upper limit for the working temperature at which ash agglomeration is started (Song et al., 2010).
- ❖ The boiler and furnace operators, and engineers in power stations regularly utilizes the knowledge of AFTs in power generation stations for predicting the melting and sticking behavior of the coal ash (Seggiani and Pannocchia, 2003).

Conventionally, determination of the ash fusibility is performed using ASTM D1857 procedure. It includes monitoring cones or pyramids of ash—prepared in a muffle furnace at 815 °C—in an oven operated under a reducing atmosphere and whose temperature is continuously increased steadily past 1000 °C to as high as possible [preferably 1600 °C (2910 °F)]. It is commonly witnessed that for a given coal, the AFT analysis conducted by different laboratories may vary by ±20 –100°C (Jak, 2002; Winegartner and Rhodes, 1975).

Since the knowledge of the AFTs is important in operating, designing, and optimizing coal-based processes, a number of mathematical models have been developed pertaining to their prediction. It may be noted that the efforts to build models possessing higher prediction accuracies and generalization performance as also wider applicability still continue. The presently existing AFT prediction models have certain limitations as specified below.

1. Most of the models have been constructed using data of coals from a few or a single geographical region. Since coals from different regions/countires show varying physical and chemical characteristics, the AFT models of coals from a single/few geographies possess limited applicability.
2. A few of the AFT prediction models do study coals from various geographies (see, for example, Seggiani, 1999, and Seggiani and Pannocchia, 2003) and are endowed with reasonably good prediction accuracies. However, these are based upon a large number of predictors (input variables) and therefore are complex, which adversely affects their generalization ability. Also, it is costly and require tedious experimentation for compiling a large number of predictors.

Accordingly, the principal aim of this work is to develop AFT prediction models that are with lower complexity (parsimonious), and applicable to coals from a large number of geographical regions. Towards the stated objective, this Chapter reports the results of the development of *computational intelligence* (CI) based models for the prediction of *IDT*, *ST*, *HT*, and *FT*. The three CI paradigms used in this modeling are *genetic programming* (GP), *multi-layer perceptron neural network* (MLPNN), and *support vector regression* (SVR). The results of the CI-based modeling of AFT prediction models show that the GP and MLP based models predicting *IDT*, *ST*, *HT*, and *FT* have outperformed the prevailing linear models with relatively wider applicability and in terms of possessing better generalization ability. Moreover, the GP and MLPNN based models need a lower number of inputs than the above specified models (Seggiani, 1999, and Seggiani and Pannocchia, 2003) thus requiring reduced cost and effort in collecting the predictor (model inputs) data.

The rest part of this study is structured as follows. An overview of the currently available AFT prediction model is given in Section 5.2. A need for developing the data-driven nonlinear models is described in section 5.3. The "Results and discussion" section (5.4) presents (a) the CI-based models development pertaining to prediction accuracy of four AFTs, and (b) a comparison of the prediction accuracy and generalization performance of the CI-based models. Finally, "Concluding Remarks" summarize the principal findings of this work.

5.2 OVERVIEW OF MODELS FOR PREDICTING AFTs

Multiple studies have demonstrated that the chemical and mineral composition of the coal ash governs its melting behavior and fusion temperatures (Gray, 1987; Kucukbayrak et al., 1993; Vassilev et al., 1995; Vorres, 1979; Winegartner and Rhodes, 1975). As ash composition influences the AFT magnitudes and, thus, the coal-based process performance, it is absolutely necessary to establish quantitative relationships between the composition of the ash and the corresponding four AFTs. The AFT predicting models have following applications:

(a) Based upon the chemical and mineral composition of the coal ashes , the models present a method to calculate thermal properties of coal ashes (Seggiani, 1999),

(b) Offer a method to assess the outcome of an addition of minerals, such as CaO, to modify the slag behavior (Wall et al., 1998). Table 5.1 represents a typical compilation of the existing AFT models.

It is well-known that the high-temperature behavior of the coal slag, ash, and blends in gasification and combustion technologies are unable to predict precisely by the conventional approaches (Gray, 1987; Goni et al., 2003; Huggins et al., 1981; Lloyd et al., 1995; Wall et al., 1998; Yin et al., 1998).

Accordingly, a number of studies have been revealed using modeling methods such as statistical, regression, thermodynamic, and more recently artificial intelligence-based data-driven techniques for the prediction of the AFTs and thereby assessing the deposition characteristics of the coal ashes. Typically, these models use the mineral content of several oxides (expressed as weight percentages), for correlating with the AFT data obtained from the standard ash fusion tests. The stated models do not closely resemble the conditions in the real gasifiers/ combustors as they essentially simulate the ash formation under controlled conditions. However, the models compensate for the said deficiency by obtaining predictions under a consistent set of test conditions (Lolja et al., 2002).

Table 5.1: A representative compilation of AFT predicting correlations/ models

SN	**Authors**	**Type of Model**	**Model inputs (Composition of ash constituents and other parameters)**	**Predicted AFT**	**Coal Region**	**Statistical analysis[#] of model predictions**
1	Winegartner and Rhodes (1975)	Stepwise Regression	FeO %, Fe_2O_3%, FeO×CaO %, Bases/Acids Ratios, silica value	*ST*	USA	*CC* > 0.7
2	Gray (1987)	Multiple Regression	Different combinations of metal oxides	Reducing *IDT* and *HT*	New Zealand	$R^{2\,>}$ 53%
3	Rhinehart and Attar	Thermodynamic Modelling	Metal and other (P_2O_5, TiO_2, SO_3)	*IDT, ST, FT*	US coals	R^2 > 0.67

	(1987)		oxides			
4	Kucuk-bayrak et al. (1993)	Least Square Regression Analysis	Combinations of metal oxides	HT	Turkish lignite	*CC* > 0.26
5	Yin et al. (1998)	Back-Propagation Neural Network	SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO, TiO_2, K_2O+Na_2O	*ST*	Chinese	Average fractional error: 0.049
6	Kahra-man et al. (1998)	Empirical models	Al_2O_3, Fe_2O_3, CaO	*IDT*, Spherical, *FT*	Australian	$R^2 > 0.84$
7	Seggiani (1999)	Linear Regression	49 parameters containing concentrations of nine metal oxides, their squares, and combinations of these values, acid, base, dolomite ratio and Silica value	Reducing *IDT*, *ST*, *HT*, and *FT* for biomass and coal ashes	American, African, Australian, German, Italian, Polish, Spanish, etc.	*CC* range: 0.84 – 0.92
8	Lolja et al. (2002)	Linear Regression	Metal oxides, bases, acids, crystal components and fluxing agents	*IDT*, *ST*, *HT*, *FT*	Albanian	0.93 ≤ *CC*≤ 0.95
9	Jak (2002)	Thermo-dynamic Modelling (FACT Package)	SiO_2, Al_2O_3, Fe_2O_3, CaO	*IDT*, Spherical, *HT*, *FT*	Australian	Liquidus Temperature and AFT strongly correlated

10	Seggiani and Pannocchia (2003)	Partial Least Squares Regression	11 to 13 parameters containing concentrations of nine metal oxides in ash, and their various combinations	Reducing *IDT*, *ST*, *HT*, *FT*	American, African, Albanian, Australian, French, German, Italian, Polish, Spanish	*CC* (training set) range: 0.75 – 0.82; *CC* (validation set) range: 0.76 – 0.84
11	Ozbayoglu and Ozbayoglu (2006)	Linear and Nonlinear Regression	Chemical composition of ash and coal parameters	*ST*, *FT*	Turkey	Regression Coefficient > 0.93
12	Liu et al. (2007)	Back Propagation Neural Network-Ant Colony Optimization	SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO, TiO_2, K_2O+Na_2O	*ST*	Chinese	Average Training (Test) error: 1.55 (1.85) %
13	Zhao et al. (2010)	Least-Squares Support Vector Regression	SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO, TiO_2, K_2O, Na_2O, SO_3	*ST*	Chinese	*CC* =0.927, *MSE*= 0.0128
14	Gao et al. (2011)	Support vector regression by ACO Algorithm	SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO, TiO_2, K_2O+Na_2O	*ST*	China	*MSE* = 1.52, *CC* = 0.999 (training), and = 0.9716 (test)
15	Karimi et al. (2014)	Adaptive Neuro fuzzy	Different combinations of metal	*IDT*, *ST*, *FT*	USA	*CC* = 0.97, 0.98 and 0.99,

		Inference System	oxides			respectively
16	Miao et al. (2016)	Back-Propagation Neural Net	SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO	*ST*	Chinese	-

# *CC*: Correlation coefficient; MSE: mean squared error; R^2: Variance

5.3 AFT DATA USED IN MODELING

In the present work, the commonly employed weight percentages (wt %) of the eight major oxides appearing in the coal ashes (i.e. SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO, TiO_2, Na_2O+K_2O), have been used as the predictors/model inputs of the four AFTs (Gao et al., 2011; Liu et al., 2007; Yin et al., 1998). Large datasets consisting the information of specified oxides composition and the related magnitudes of the four AFTs were compiled from a number of research articles, pertaining to the ash samples of coals from several countries, listed below

1. *IDT*: Albania, Australia, China, Colombia, India, Indonesia, Russia, South Sumatra, South Africa, USA, Venezuela.
2. *ST*: Australia, China, Colombia, Indonesia, Russia, South Africa, South Sumatra, USA, Venezuela.
3. *HT*: Albania, Australia, Colombia, India, Indonesia, Russia, South Africa, South Sumatra, USA, Venezuela.
4. *FT:* Albania, Australia, China, Colombia, India, Indonesia, Russia, South Africa, South Sumatra, USA, Venezuela.

The data sets also contained data in respect of the Indian coal ashes, which were provided by *Central Institute of Mining and Fuel Research* (CIMFR), Dhanbad, India. The data sets containing of the seven model predictors, the related AFTs, and their sources are listed in Tables 5.A, 5.B, 5.C and 5.D in the APPENDIX 5 at the end of this chapter. Specifically, these four tables comprise the oxide composition and the related AFT data pertaining to the *IDT* (184 samples), *ST* (92 samples), *HT* (82 samples), and *FT* (94 samples), respectively. Most of these data were collected by conducting experiments in the reducing conditions.

5.4 NEED FOR NONLINEAR AFT MODELS

The regression studies, for example, by Lolja et al. (2002), Ozbayoglu and Ozbayoglu (2006), and Seggiani (1999), have reported that a linear dependence exists between the mineral composition of the coal ash and its AFTs. On the other hand, a number of studies such as those by Gao et al. (2011), Karimi et al. (2014), Liu et al. (2007), Miao et al. (2016) and Yin et al. (1998) (see Table 5.1) have proposed nonlinear AFT prediction models. In order to analyze the true nature (linear or nonlinear) of the dependencies between the oxide components in the coal ashes and the corresponding AFTs, cross-plots were prepared as shown in Figures 5.1 to 5.4. In here, values of the four AFTs (*IDT*, *ST*, *HT*, and *FT*) are plotted against the weight (%) values of the individual oxide. The observations drawn from the cross-plots are given below.

- In the four panels, namely, *5.1a*, *5.1b*, *5.1c* and *5.1g*, an approximately linear relation is seen between the *IDT* and the weight percentages of SiO_2, Al_2O_3, Fe_2O_3, and K_2O+Na_2O. However, since there exists a significant scatter in the corresponding data, a similar conclusion cannot be drawn from the cross-plots (panels *5.1d*, *5.1e*, and *5.1f*) pertaining to the remaining three ash components (CaO, MgO, and TiO_2).
- Irrespective of the high scatter seen in all panels of Figure 5.2, there exists a strong probability of linear dependencies between the softening temperature and Al_2O_3 (panel *5.2b*), Fe_2O_3 (panel *5.2c*), MgO (panel *5.2e*), and TiO_2 (panel *5.2f*) whereas, most probably the individual relationships between the *ST* and SiO_2 (panel *5.2a*), CaO (panel *5.2d*) and K_2O +Na_2O (panel *5.2g*) are nonlinear.
- The cross-plots in Figure 5.3 are suggestive of a high probability of nonlinear dependencies between *HT* and weight percentages of three ash components, viz. SiO_2 (panel *5.3a*), Al_2O_3 (panel *5.3b*) and CaO (panel *5.3d*); whereas, the relationships between *HT* and Fe_2O_3 (panel *5.3c*), MgO (panel *5.3e*), TiO_2 (panel *5.3f*), and K_2O+Na_2O (panel *5.3g*), could be linear.

- It is possible to state with a high probability that in Figure 5.4, linear dependencies exist between the flow temperature (*FT*) and SiO_2 (panel 5.4*a*), Al_2O_3 (panel 5.4*b*), Fe_2O_3 (panel 5.4*c*), and K_2O + Na_2O (panel 5.4*g*). However, in the remaining three panels, the relationships between *FT* and CaO (panel 5.4*d*), MgO (panel 5.4*e*) and TiO_2 (panel 5.4*f*), appear to be nonlinear.

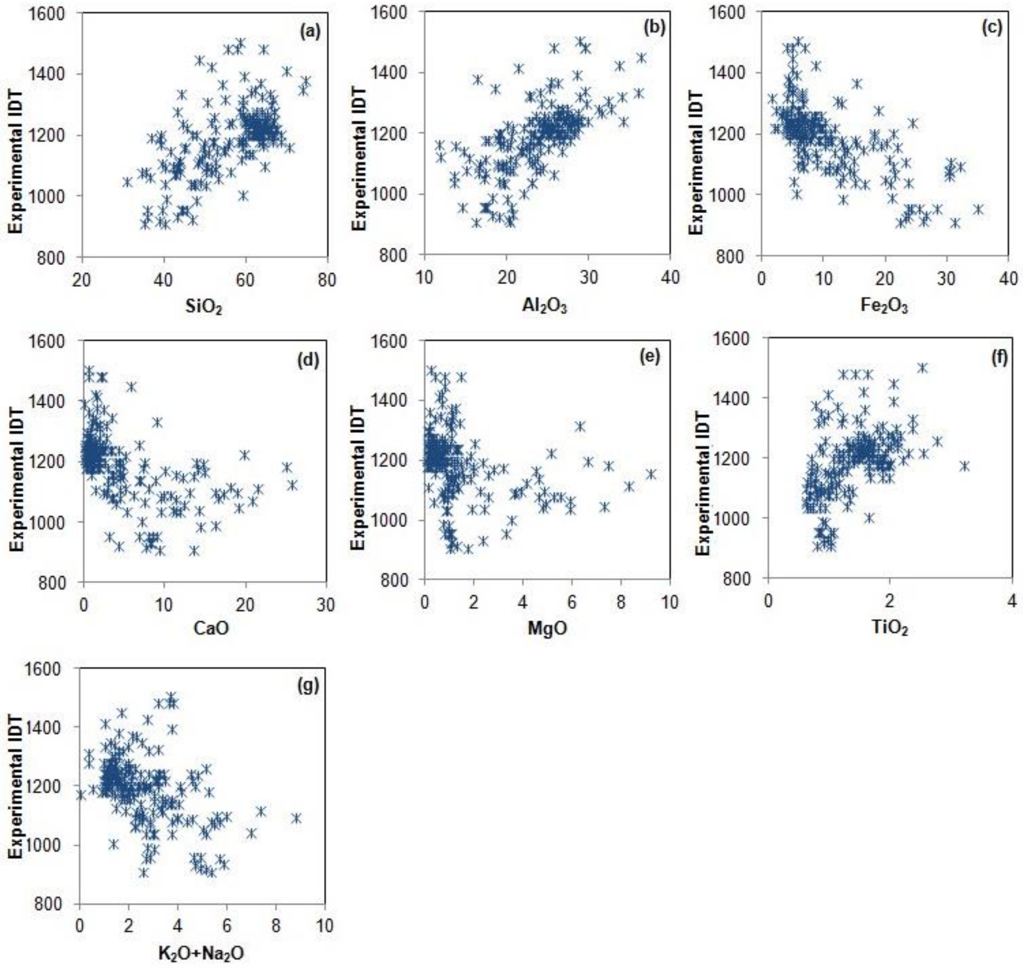


Figure 5.1: Cross-plots pertaining to weight percentages (wt %) of individual metal oxides vis-à-vis *IDT*

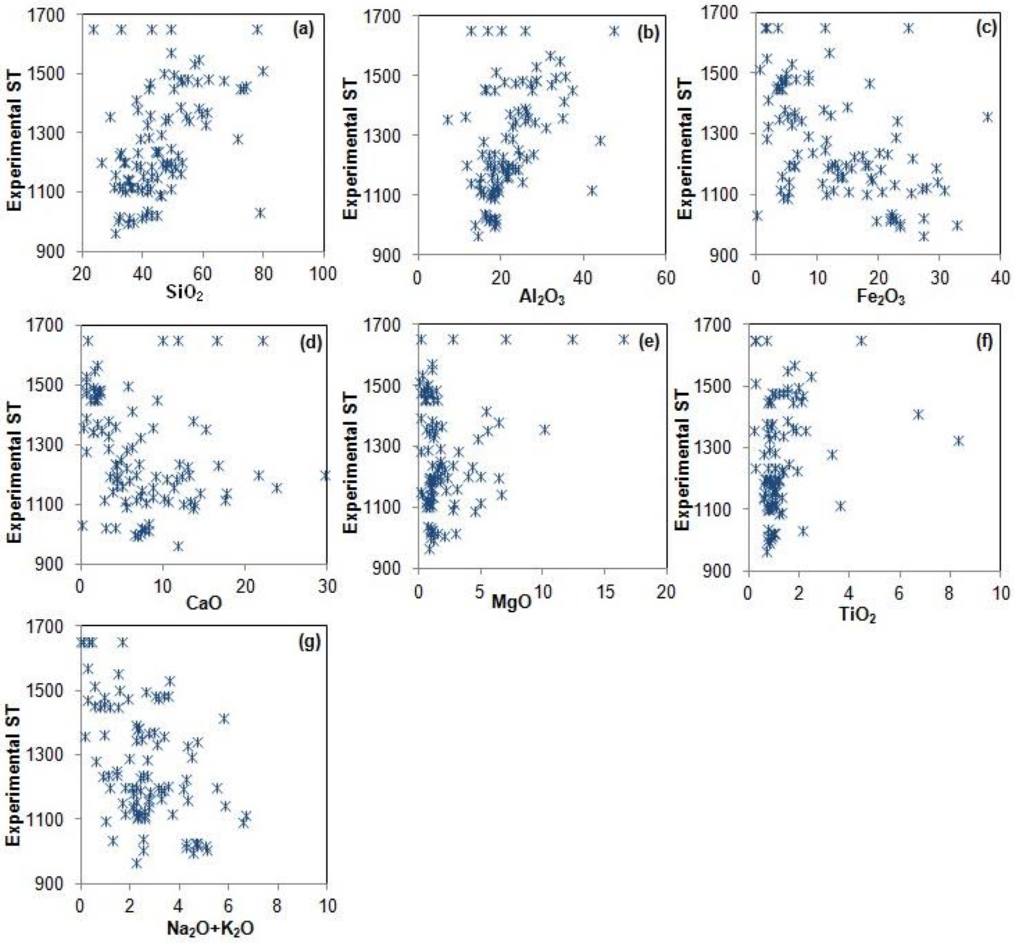


Figure 5.2: Cross-plots pertaining to weight percentages (wt %) of individual metal oxides vis-à-vis *ST*

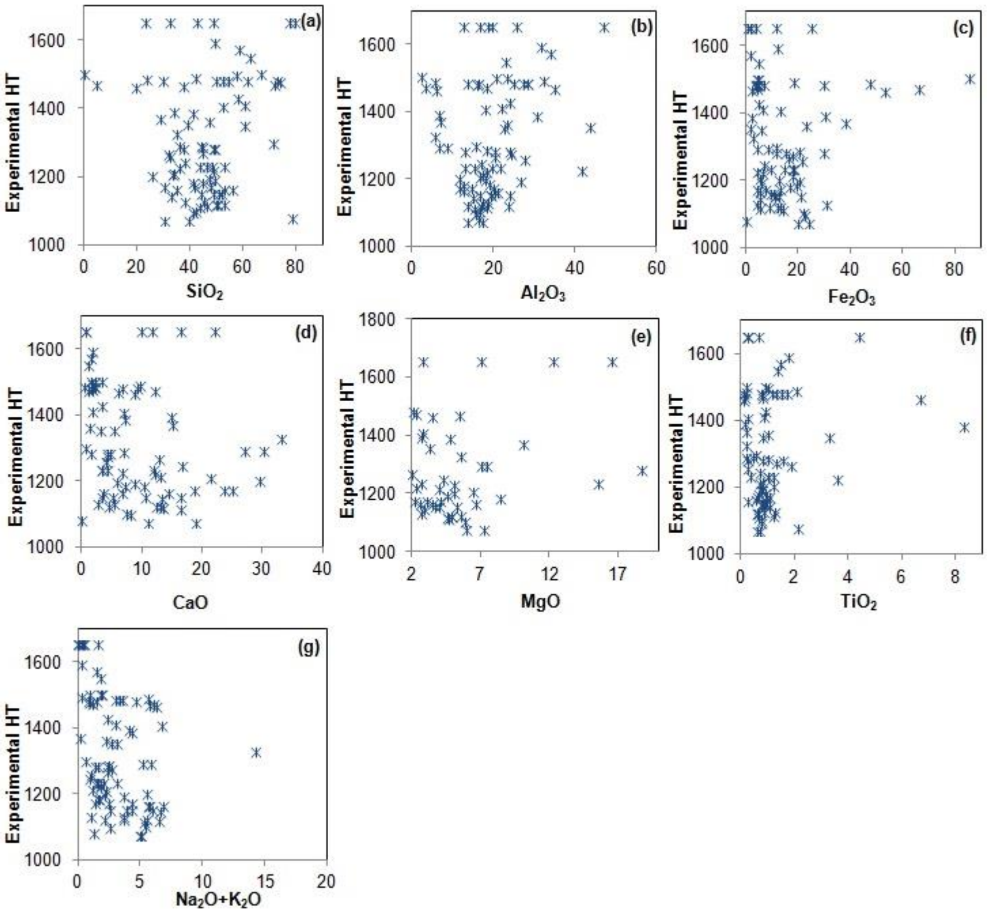


Figure 5.3: Cross-plots pertaining to weight percentages (wt %) of individual metal oxides vis-à-vis *HT*

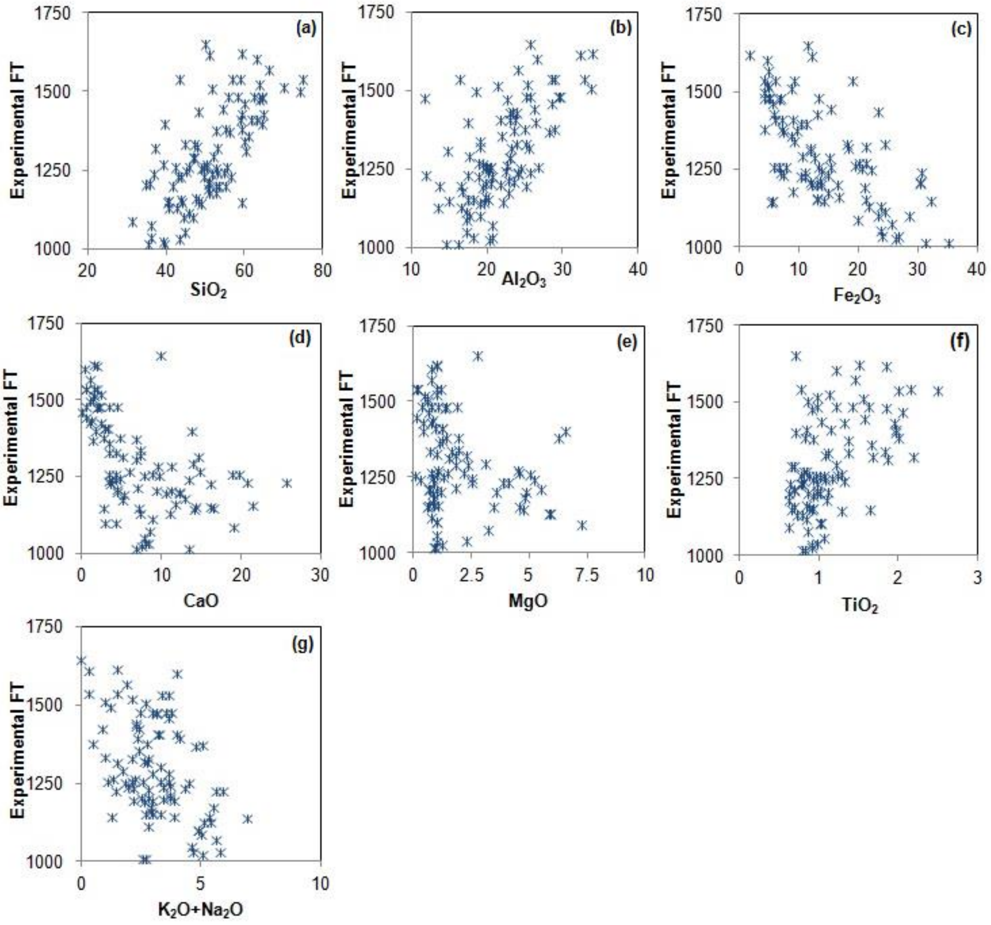


Figure 5.4: Cross-plots pertaining to weight percentages (wt %) of individual metal oxides vis-à-vis *FT*

From the above-stated observations, it is evident that several potential cases exist of the nonlinear relationships between the four AFTs and the weight percentages of the individual oxides present in the coal ashes. Hence, it becomes pertinent to explore nonlinear models for the prediction of the four AFTs since such models are anticipated to capture the oxide-AFT relationships in a manner better than the linear models and, thereby, make more accurate predictions. Towards this objective, three computational intelligence (CI) based purely data-driven modeling formalisms (GP, MLPNN and SVR) have been employed for the prediction of the four AFTs from the information of the mineral composition (oxides) of the coal ashes. The underlying basis for emerging multiple models for each AFT is to compare their prediction accuracy and generalization performances, and thus select the best performing one.

The general forms of the CI-based models developed in this study are given as:

$$IDT = f_1\ (x_1, x_2, x_3, x_4, x_5, x_6, x_7, \beta_{\mathrm{IDT}}\ ) \tag{5.1}$$

$$ST = f_2\ (x_1, x_2, x_3, x_4, x_5, x_6, x_7, \beta_{ST}\ ) \tag{5.2}$$

$$\mathrm{HT} = f_3\ (\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \mathrm{x}_4, \mathrm{x}_5, \mathrm{x}_6, \mathrm{x}_7,\ \beta_{\mathrm{HT}}\ ) \tag{5.3}$$

$$\mathrm{FT} = f_4\ (\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \mathrm{x}_4, \mathrm{x}_5, \mathrm{x}_6, \mathrm{x}_7,\ \beta_{\mathrm{FT}}\ ) \tag{5.4}$$

where, *IDT*, *ST*, *HT*, and *FT* stand for *Initial deformation temperature* (^{0}C), *Softening temperature* (^{0}C), *Hemispherical temperature* (^{0}C), and *Fluid temperature* (^{0}C), respectively; β_{IDT}, β_{ST}, $\beta_{\mathrm{HT,}}$ and β_{FT}, represent the parameter vectors associated with the functions f_1, f_2, $f_{3,}$ and f_4, respectively. The seven predictors (inputs) of the four AFT models are defined as: (i) x_1: weight percentage (wt%) of SiO_2, (ii) x_2: wt% of Al_2O_3, (iii) x_3: wt% of Fe_2O_3, (iv) x_4: wt% of CaO, (v) x_5: wt% of MgO, (vi) x_6: wt% of TiO_2, and (vii) x_7: wt% of K_2O+Na_2O.

5.5 RESULTS AND DISCUSSION

5.5.1 Principal Component Analysis (PCA)

While developing data-driven models, it is desirable to avoid correlated predictors/inputs since these cause redundancies and an unnecessary increase in the computational effort. Thus, the seven predictors/inputs ($x_1 - x_7$) of *IDT, ST, HT,* and *FT* prediction models were subjected to the *principal component analysis* (PCA) (Geladi and Kowalski, 1986), which performs a transformation to get linearly uncorrelated variables. Therefore, only the first few principle components (*PC*s) that capture the maximum amount of variance in the data are selected as model inputs. This enables a reduction in the dimensionality of the model's input space. An in-depth description has been provided in Chapter 2, section 2.6.1. In this study, seven *PC*s were extracted from weight percentages of metal oxides in the coal ashes listed in the four tables in APPENDIX 5.

Prior to performing the PCA, the seven predictors in the example data for *IDT*, *ST*, *HT* and *FT* (listed in Tables 5.A, 5.B, 5.C and 5.D of APPENDIX 5, respectively) were normalized using "*Z-score*" technique as given by,

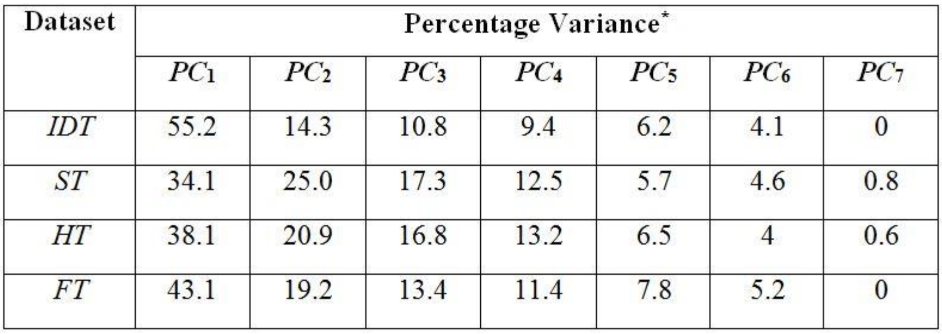
$$\hat{x}_r^i = \frac{x_r^i - \bar{x}_r}{\sigma_r}; i = 1,2, \dots, N_p; r = 1,2, \dots, R \qquad (5.5)$$

where, x_r^i signifies the ith value of the rth non-normalized input variable, x_r; *R* signifies the number of inputs subjected to PCA (=7); N_p represents the number of observations/ samples in the data set; $\overline{x}_r$ denotes to the mean of $\{x_r^i\}$, $i = 1, 2, \dots, N_p$, and σ_r denotes the standard deviation of $\{x_r^i\}, i = 1, 2, \dots, N_p$. Alike to the input variables, the values of the four model outputs, namely *IDT* (y_1), *ST* (y_2), *HT* (y_3) and *FT* (y_4) were also normalized as given below.

$$\hat{y}_S^i = \frac{y_S^i - \overline{y}_s}{\sigma_S}, \; i = 1,2, \dots, N_p \; ; \; S = 1, 2, \dots, S \qquad (5.6)$$

where, $\hat{y}_S^i$, and y_S^i respectively denote the ith normalized and non-normalized magnitudes of the Sth output variable (i.e., AFT), y_S; S denotes to the number of outputs (= the number of AFTs = 4); $\overline{y}_s$ denotes to the mean of $\{y_S^i\}, i = 1, 2, \dots, N_p$ values in the example set, and σ_S signifies the standard deviation of $\{y_S^i\}, i = 1, 2, \dots, N_p$. The mean and standard deviation magnitudes of weight percentages of seven oxides (SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO, TiO_2, and K_2O+Na_2O) and all four ash fusion temperature phases (*IDT*, *ST*, *HT*, and *FT*) used in the normalization procedure are given in Table 5.E in APPENDIX 5. Using the input data, the expressions of the *PC*s have been derived pertaining to the *IDT*, *ST*, *HT* and *FT* models are given in the next section.

Table 5.2: Amount of variance addressed by the individual principal components (*PC*s) in the *IDT*, *ST*, *HT* and *FT* data sets



Dataset	Percentage Variance*						
	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6	PC_7
IDT	55.2	14.3	10.8	9.4	6.2	4.1	0
ST	34.1	25.0	17.3	12.5	5.7	4.6	0.8
HT	38.1	20.9	16.8	13.2	6.5	4	0.6
FT	43.1	19.2	13.4	11.4	7.8	5.2	0

*PC_j represents the jth principal component.

Table 5.2 addresses the amount of the variance captured by the seven *PC*s in the experimental oxides data. In Table 5.2, it is observed that the first five *PC*s have cumulatively captured large percentages (> 95%) of variance in the oxides data pertaining to *IDT*, *HT* and *FT*. In the case of the oxides data for *ST* the cumulative amount of variance captured by the first four *PC*s is ≈ 92%, which is also sufficiently high. This result suggests that since they capture a high percentage of variance in the oxides data, the first four *PC*s, can be considered as the predictors/inputs in the model pertaining to the predictions of *ST*; whereas in case of models pertaining to the predictions of *IDT*, *HT* and *FT*, first five *PC*s can be regarded as predictors.

The prediction and generalization ability of a CI-based model were studied using three statistical metrics, namely *coefficient of correlation* (*CC*), *root mean square error* (*RMSE*), and *mean absolute percent error* (*MAPE*). Their details are given in Chapter 2 section 2.3.1.3

The PCA-transformed variables were considered as the predictors in developing the GP-, MLPNN-, and SVR-based *IDT*, *ST*, HT, and *FT* prediction models. Before constructing and examining the generalization capability of CI-based models, the experimental data set for each AFT was randomly partitioned in 70:20:10 ratio into *training*, *test,* and *validation* sets. While the first set was used in training the CI-based models, the test and the validation sets were respectively used in testing and validating the generalization ability of the models.

5.5.2 AFT Modeling using Genetic Programming

The GP-based symbolic regression strategy explained in Chapter 2, section 2.3.3 was used in developing the four AFT prediction models. These models were constructed using *Eureqa Formulize* software package (Schmidt and Lipson, 2009). A noteworthy characteristic of this package is that it is tailored to search and optimize models possessing a low complexity and, thus, endowed with the much sought-after generalization ability. There are a number of procedural attributes that affect the final solution provided by the GP. These include the sizes and distribution of the training, test and validation sets, choice of the mathematical operators for constructing the candidate solutions, and input normalization schemes. To get parsimonious models (i.e. of low complexity) having a good AFT prediction and generalization ability, a number of GP runs were conducted by rigorously varying each of the specified attributes. In each of the stated runs, post-convergence the solution possessing maximum fitness score was recorded. Among such multiple solutions, a model possessing following characteristics was chosen as an overall optimal solution (Sharma and Tambe, 2014): (a) high and comparable values of *CC*s, and small and comparable values of *RMSE*, and *MAPE*, pertaining to the model predictions in respect of the training, test, and validation set data, and (b) a model possessing a low complexity (i.e., containing a small number of terms and parameters in its structure).

The GP-based overall best models for the four AFTs are given below wherein $\widehat{IDT}$, $\widehat{ST}$, $\widehat{HT}$, and $\widehat{FT}$, respectively refer to the normalized values of *IDT, ST, HT,* and *FT* (see Equations 5.7, 5.13, 5.18 and 5.24).

(a) Model for predicting Initial deformation temperature (IDT)

$$\widehat{IDT} = 0.096PC_1PC_2 + 0.309PC_4^2 + \left(\frac{PC_1 + PC_5}{PC_3 - 1.426PC_5 - 3.252}\right) - 0.096PC_1PC_4^2 - 0.121PC_3 - 0.199 \qquad (5.7)$$

where,

$$PC_1 = -0.423\hat{x}_1 - 0.375\,\hat{x}_2 + 0.343\,\hat{x}_3 + 0.427\,\hat{x}_4 + 0.366\,\hat{x}_5 - 0.384\,\hat{x}_6 + 0.315\,\hat{x}_7 \qquad (5.8)$$

$$PC_2 = 0.406\,\hat{x}_1 - 0.119\,\hat{x}_2 - 0.709\,\hat{x}_3 + 0.163\,\hat{x}_4 + 0.474\,\hat{x}_5 - 0.192\,\hat{x}_6 + 0.171\,\hat{x}_7 \qquad (5.9)$$

$$PC_3 = 0.373\,\hat{x}_1 - 0.479\,\hat{x}_2 + 0.120\,\hat{x}_3 - 0.389\,\hat{x}_4 - 0.405\,\hat{x}_5 - 0.250\,\hat{x}_6 + 0.489\,\hat{x}_7 \qquad (5.10)$$

$$PC_4 = 0.236\,\hat{x}_1 - 0.459\,\hat{x}_2 + 0.085\,\hat{x}_3 + 0.122\,\hat{x}_4 - 0.078\,\hat{x}_5 - 0.314\,\hat{x}_6 - 0.779\,\hat{x}_7 \qquad (5.11)$$

$$PC_5 = 0.064\,\hat{x}_1 - 0.539\,\hat{x}_2 - 0.003\,\hat{x}_3 + 0.252\,\hat{x}_4 + 0.023\,\hat{x}_5 + 0.799\,\hat{x}_6 + 0.052\,\hat{x}_7 \qquad (5.12)$$

The magnitudes of $\hat{x}_r$; r = 1,2,...,7 were evaluated using Eq. (5.5); the mean and standard deviation values pertaining to the *IDT* data are given in Table 5.E in APPENDIX 5 at the end of this Chapter.

(b) Model for predicting softening temperature (*ST*)

$$\widehat{ST} = 0.261PC_2 + 0.230PC_1PC_2 + 0.142PC_1PC_3 - 0.329PC_4 - 0.527PC_1 \; - 0.261PC_3PC_4 \tag{5.13}$$

where,

$$PC_1 = \; -0.494\,\hat{x}_1 - 0.394\,\hat{x}_2 + \; 0.356\,\hat{x}_3 + \; 0.535\,\hat{x}_4 + \; 0.325\,\hat{x}_5 - 0.237\,\hat{x}_6 + \; 0.165\,\hat{x}_7 \tag{5.14}$$

$$PC_2 = \; -0.307\,\hat{x}_1 + 0.457\,\hat{x}_2 - \; 0.316\,\hat{x}_3 + \; 0.237\,\hat{x}_4 + \; 0.399\,\hat{x}_5 + 0.596\,\hat{x}_6 + \; 0.161\,\hat{x}_7 \tag{5.15}$$

$$PC_3 = \; 0.291\,\hat{x}_1 - 0.134\,\hat{x}_2 - 0.426\,\hat{x}_3 + \; 0.25\,\hat{x}_4 + \; 0.478\,\hat{x}_5 - 0.228\,\hat{x}_6 - 0.611\,\hat{x}_7 \tag{5.16}$$

$$PC_4 = \; 0.328\,\hat{x}_1 - 0.362\,\hat{x}_2 - 0.523\,\hat{x}_3 + 0.153\,\hat{x}_4 - \; 0.012\,\hat{x}_5 - 0.066\,\hat{x}_6 + 0.678\,\hat{x}_7 \tag{5.17}$$

In Eqns. (5.14) to (5.17), $\hat{x}_r$; r = 1,2,...,7 were calculated using Eq. (5.5); the mean and standard deviation values pertaining to the *ST* data are given in Table 5.E in APPENDIX 5.

(c) Model for predicting hemispherical temperature (*HT*)

$$\widehat{HT} = 0.529 + \left(\frac{0.736\,PC_4^2 - PC_1 - 1.678 - 0.592PC_5 - 0.578PC_4PC_5}{\exp(0.397PC_2PC_3 + 0.358PC_1PC_2 + 0.358\,PC_1^2}\right) \tag{5.18}$$

where,

$$PC_1 = \; -0.44\,\hat{x}_1 - 0.486\,\hat{x}_2 + \; 0.375\,\hat{x}_3 + \; 0.421\,\hat{x}_4 + \; 0.25\,\hat{x}_5 - 0.303\,\hat{x}_6 + \; 0.314\,\hat{x}_7 \tag{5.19}$$

$$PC_2 = \; 0.202\,\hat{x}_1 - 0.312\,\hat{x}_2 + \; 0.402\,\hat{x}_3 - \; 0.426\,\hat{x}_4 - \; 0.396\,\hat{x}_5 - 0.522\,\hat{x}_6 + \; 0.299\,\hat{x}_7 \tag{5.20}$$

$$PC_3 = -0.549\,\hat{x}_1 + 0.277\,\hat{x}_2 + 0.54\,\hat{x}_3 - 0.25\,\hat{x}_4 - 0.26\,\hat{x}_5 + 0.447\,\hat{x}_6 - 0.014\,\hat{x}_7 \tag{5.21}$$

$$PC_4 = -0.129\,\hat{x}_1 + 0.047\,\hat{x}_2 + 0.13\,\hat{x}_3 - 0.061\,\hat{x}_4 + 0.662\,\hat{x}_5 - 0.017\,\hat{x}_6 - 0.723\,\hat{x}_7 \tag{5.22}$$

$$PC_5 = 0.206\,\hat{x}_1 - 0.043\,\hat{x}_2 + 0.159\,\hat{x}_3 - 0.629\,\hat{x}_4 + 0.517\,\hat{x}_5 + 0.054\,\hat{x}_6 + 0.515\,\hat{x}_7 \tag{5.23}$$

The magnitudes of $\hat{x}_r$; r = 1,2,...,7 in Eqns. (5.19) to (5.23) were computed using Eq. (5.5); the mean and standard deviation values pertaining to the *HT* data are given in Table 5.E in APPENDIX 5

(d) Model for predicting flow temperature (*FT*)

$$\widehat{FT} = 0.133PC_1PC_2 + 0.133PC_4^2 + 0.0577PC_1^2\,(0.058PC_1^2)^{(PC_5+0.133PC_1+0.133PC_4^2)} - 0.354 - 0.133PC_3 - 0.407PC_1 \tag{5.24}$$

where,

$$PC_1 = -0.45\,\hat{x}_1 - 0.389\,\hat{x}_2 + 0.34\,\hat{x}_3 + 0.449\,\hat{x}_4 + 0.318\,\hat{x}_5 - 0.41\,\hat{x}_6 + 0.244\,\hat{x}_7 \tag{5.25}$$

$$PC_2 = 0.391\,\hat{x}_1 - 0.082\,\hat{x}_2 - 0.668\,\hat{x}_3 + 0.209\,\hat{x}_4 + 0.547\,\hat{x}_5 - 0.222\,\hat{x}_6 + 0.05\,\hat{x}_7 \tag{5.26}$$

$$PC_3 = -0.013\,\hat{x}_1 + 0.176\,\hat{x}_2 - 0.007\,\hat{x}_3 - 0.3721\,\hat{x}_4 + 0.109\,\hat{x}_5 + 0.055\,\hat{x}_6 + 0.903\,\hat{x}_7 \tag{5.27}$$

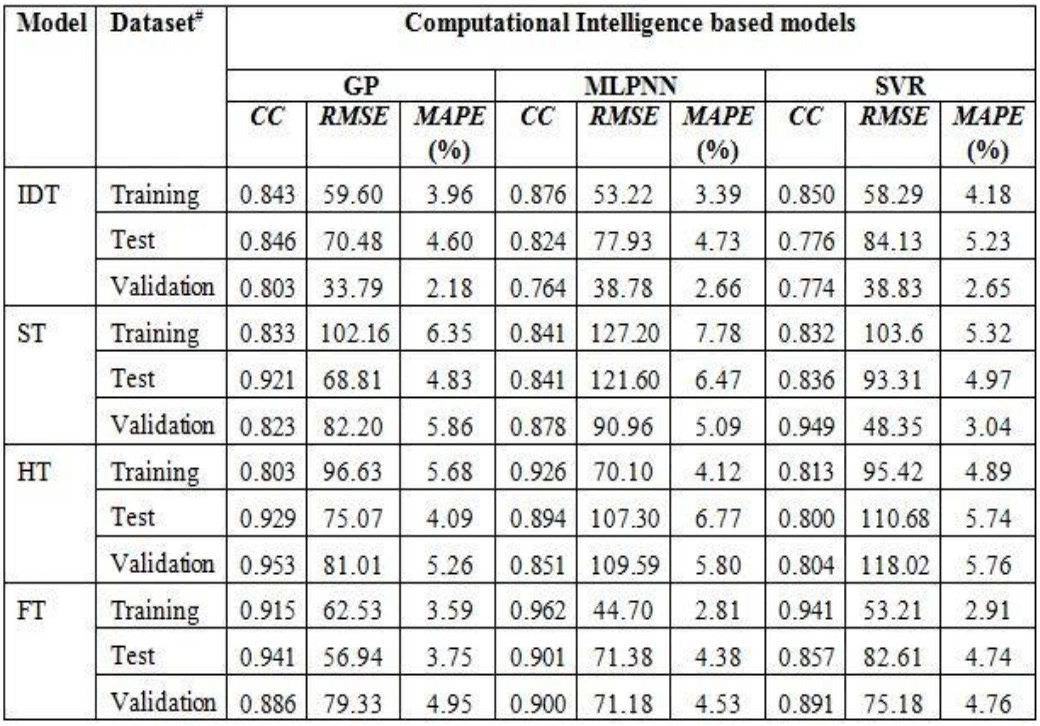
$$PC_4 = -0.474\,\hat{x}_1 + 0.726\,\hat{x}_2 - 0.055\,\hat{x}_3 + 0.152\,\hat{x}_4 + 0.41\,\hat{x}_5 + 0.183\,\hat{x}_6 - 0.146\,\hat{x}_7 \tag{5.28}$$

$$PC_5 = -0.056\,\hat{x}_1 - 0.33\,\hat{x}_2 - 0.1\,\hat{x}_3 + 0.378\,\hat{x}_4 + 0.049\,\hat{x}_5 + 0.841\,\hat{x}_6 + 0.162\,\hat{x}_7 \tag{5.29}$$

In Eqns. (5.25) to (5.29), $\hat{x}_r$; r = 1,2,...,7 were calculated using Eq. (5.5); the mean and standard deviation values pertaining to the *FT* data are given in Table 5.E in APPENDIX 5.

It can be observed that, all the four GP-based models (Eqs. 5.7, 5.13, 5.18 and 5.24) pertaining to the predictions of *IDT*, *ST*, *HT* and *FT* values, respectively have nonlinear structures (forms). It can also be seen that, these models—via principal components—take into account concentrations of all eight principal metal oxides contained in the coal ashes. This result clearly shows that the relationships between the weight percentages of the eight metal oxides and the four AFTs are indeed nonlinear. Therefore, it can be inferred that nonlinear models are more suited for the AFT prediction than the linear ones. The *CC*, *RMSE* and *MAPE* values in respect of the AFT predictions made by the four GP-based models (Eqs. 5.7, 5.13, 5.18 and 5.24) for the training, test, and validation datasets are listed in **Table 5.3**.

Table 5.3: Results of the statistical analysis pertaining to the prediction and generalization performance of the GP-, MLP-, and SVR-based models predicting magnitudes of *IDT*, *ST*, *HT*, and *FT*



Model	Dataset#	Computational Intelligence based models								
		GP			MLPNN			SVR		
		CC	*RMSE*	*MAPE* (%)	*CC*	*RMSE*	*MAPE* (%)	*CC*	*RMSE*	*MAPE* (%)
IDT	Training	0.843	59.60	3.96	0.876	53.22	3.39	0.850	58.29	4.18
	Test	0.846	70.48	4.60	0.824	77.93	4.73	0.776	84.13	5.23
	Validation	0.803	33.79	2.18	0.764	38.78	2.66	0.774	38.83	2.65
ST	Training	0.833	102.16	6.35	0.841	127.20	7.78	0.832	103.6	5.32
	Test	0.921	68.81	4.83	0.841	121.60	6.47	0.836	93.31	4.97
	Validation	0.823	82.20	5.86	0.878	90.96	5.09	0.949	48.35	3.04
HT	Training	0.803	96.63	5.68	0.926	70.10	4.12	0.813	95.42	4.89
	Test	0.929	75.07	4.09	0.894	107.30	6.77	0.800	110.68	5.74
	Validation	0.953	81.01	5.26	0.851	109.59	5.80	0.804	118.02	5.76
FT	Training	0.915	62.53	3.59	0.962	44.70	2.81	0.941	53.21	2.91
	Test	0.941	56.94	3.75	0.901	71.38	4.38	0.857	82.61	4.74
	Validation	0.886	79.33	4.95	0.900	71.18	4.53	0.891	75.18	4.76

# The number of patterns in training, test and validation sets are: (a) *IDT* model: 129, 37, 18, (b) *ST* model: 68, 19, 10, (c) HT model: 57,17,8, (d) *FT* model: 66,19,9.

5.5.3 Multilayer Perceptron Neural Network (MLPNN) based AFT Models

The MLPNN-based four AFT models were trained using *error-back-propagation* (EBP) algorithm in *RapidMiner* data-mining suite (Mierswa et al., 2006; RapidMiner, 2007). To obtain an optimal MLPNN model pertaining to the high AFT prediction, generalization and validation performance, its structural and EBP algorithm-specific parameters, viz. number of nodes/neurons in each hidden layer, the number of hidden layers, *learning rate* (η), and the *momentum coefficient* (μ), were varied in a systematic manner. The particulars of the MLPNN architecture, magnitudes of the EBP-algorithm specific parameters (η, μ), and the type of transfer functions used in securing the optimal *IDT*, *ST*, *HT* and *FT* prediction models are given in Table 5.4 (also see Figure 5.5).

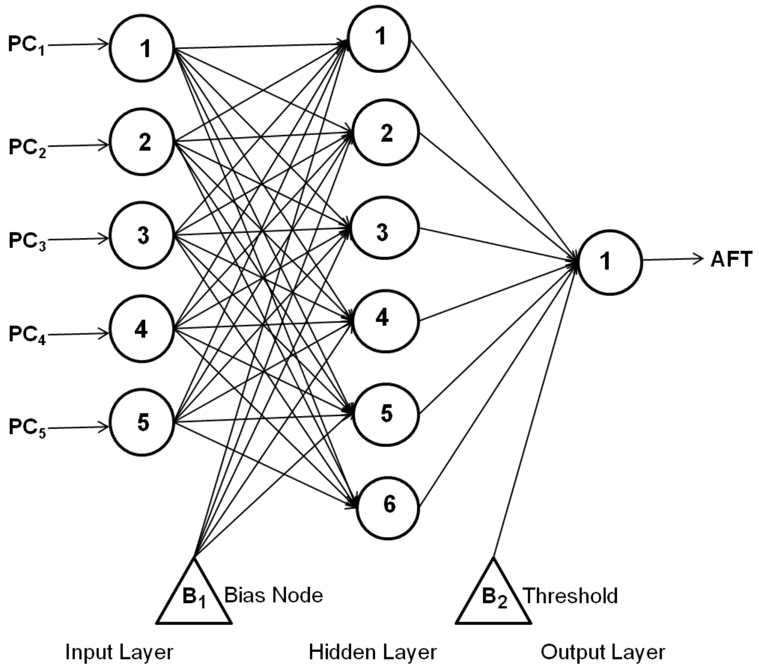
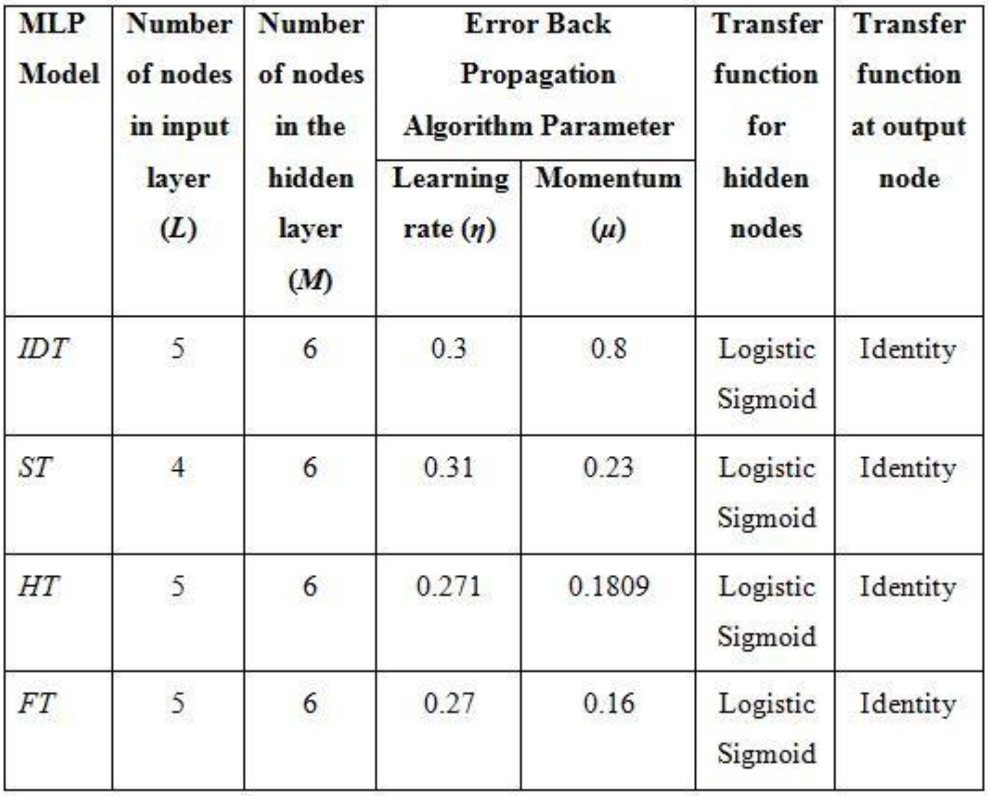


Figure 5.5: A Figure of optimal MLPNN architecture with five Input Neurons, six hidden neurons and a single output node referring to the AFT (For *ST* Model the number of input nodes were four)

The predictor nodes in the four MLPNN-based AFT models signify principal components computed by the oxide data sets given in Tables 5.A, 5.B, 5.C and 5.D of APPENDIX 5. The prediction accuracy and the generalization performance of the optimal MLPNN-based models were evaluated in terms of *CC, RMSE* and *MAPE* magnitudes (see Table 5.3). These statistical quantities were calculated using the experimental (target) and the corresponding model predicted values of the four AFTs.

Table 5.4: Details of the MLPNN-based optimal models for the prediction of *IDT*, *ST*, *HT,* and *FT*

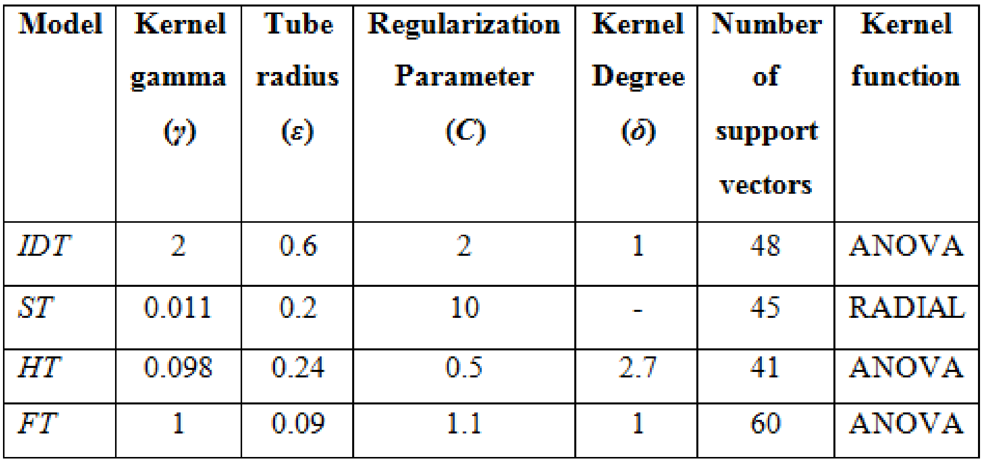


MLP Model	Number of nodes in input layer (*L*)	Number of nodes in the hidden layer (*M*)	Error Back Propagation Algorithm Parameter		Transfer function for hidden nodes	Transfer function at output node
			Learning rate (η)	Momentum (μ)		
IDT	5	6	0.3	0.8	Logistic Sigmoid	Identity
ST	4	6	0.31	0.23	Logistic Sigmoid	Identity
HT	5	6	0.271	0.1809	Logistic Sigmoid	Identity
FT	5	6	0.27	0.16	Logistic Sigmoid	Identity

5.5.4 SVR-based AFT Models

Similar to MLPNN, the SVR-based models pertaining to AFT predictions were developed using *RapidMiner* data-mining suite (Mierswa et al., 2006; RapidMiner, 2007). These models were built using the widely used ε-SVR algorithm wherein *ANOVA* and *radial basis function* (RBF) were used as the kernel functions. There are three basic parameters viz. *kernel gamma* (γ), *regularization constant* (*C*), and, the *radius of the tube* (ε) when *ANOVA* and *RBF* kernels are used in the framework of ε-SVR. For *ANOVA* kernel, an additional parameter namely *kernel degree* needs to be considered. To find four optimal SVR models pertaining to high prediction and generalization ability, the stated parameters were varied systematically. The magnitudes of the stated ε-SVR parameters that led to optimal SVR models and the corresponding number of support vectors are given in **Table 5.5.** Table 5.3 lists the *CC*, *RMSE*, and *MAPE* magnitudes of the SVR-based model predictions of the four AFTs for training, test, and validation set data. The SVR models pertaining to the *IDT, ST, HT and FT* predictions are presented in APPENDIX 5, SVR Model 5. F.1, 5. F.2, 5. F.3, 5. F.4 respectively.

Table 5.5: Details of the SVR-based optimal models predicting *IDT*, *ST*, *HT*, and *FT*



Model	Kernel gamma (γ)	Tube radius (ε)	Regularization Parameter (C)	Kernel Degree (δ)	Number of support vectors	Kernel function
IDT	2	0.6	2	1	48	ANOVA
ST	0.011	0.2	10	-	45	RADIAL
HT	0.098	0.24	0.5	2.7	41	ANOVA
FT	1	0.09	1.1	1	60	ANOVA

5.5.5 Comparison of CI-based AFT prediction Models

(a) Initial Deforming Temperature (*IDT*) prediction models

It is observed in Table 5.3 that the *CC* magnitudes in respect of the *IDT* predictions by the GP-based model are sufficiently high (range: ~0.80 to ~0.85). Alos, the corresponding *MAPE* values (range: ~ 2.20 to ~ 4.0) are low. These values show that the GP-based model has good prediction accuracy and the much-desired generalization ability pertaining to *IDT* predictions. It can also be seen in Table 5.3 that the *CC*, *RMSE*, and *MAPE* values in respect of the *IDT* predictions made by the GP model for training, test and validation set data are superior (high *CC* and low *RMSE* and *MAPE* values) to the corresponding values in respect of the predictions made by the MLPNN and SVR based models. Figure 5.6 displays three parity plots wherein the GP-, MLPNN-, and SVR model predicted *IDT* values are plotted against the corresponding experimental

IDT values. In all the three plots, a good match is seen between the experimental and model-predicted *IDT* values. It is also noticed in the panel (a) of Figure 5.6 that taken together the GP-model predictions for the training, test and validation data exhibit a lower scatter compared to the predictions by the MLPNN and SVR based models thus further supporting the inference of its superior performance based on the *CC*, *RMSE* and *MAPE* values.

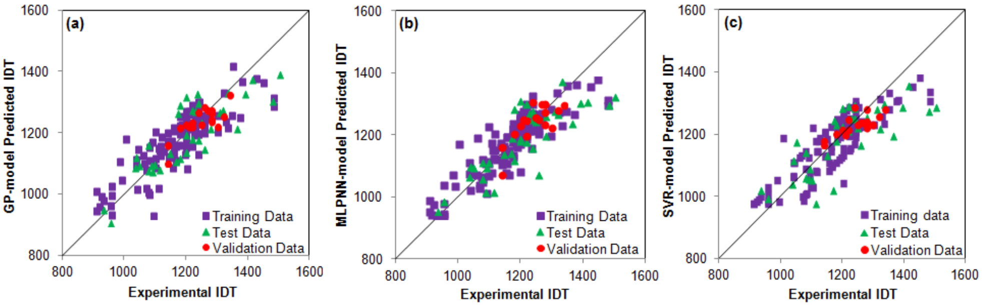


Figure 5.6: Plots pertaining to experimental *IDT* values vis-à-vis those predicted by the GP (panel *a*), MLPNN (panel *b*), and SVR based (panel *c*) models.

(b) Softening Temperature (ST) prediction Models

The *CC*, *RMSE* and *MAPE* values pertaining to the *ST* predictions made by the GP, MLPNN and SVR based models listed in Table 5.3 show the following.

- ❖ There exists only a small variation in the *ST* prediction accuracy and generalization performance of the three CI-based models.
- ❖ The overall prediction accuracy and generalization performance of the GP-based model is a tad better than the MLPNN- and SVR-based models. This observation is also explicitly supported by the lower scatter seen in the ST predictions made by the GP-based model (see Figure 5.7 panel (*a*)), compared to the predictions of the other two models.

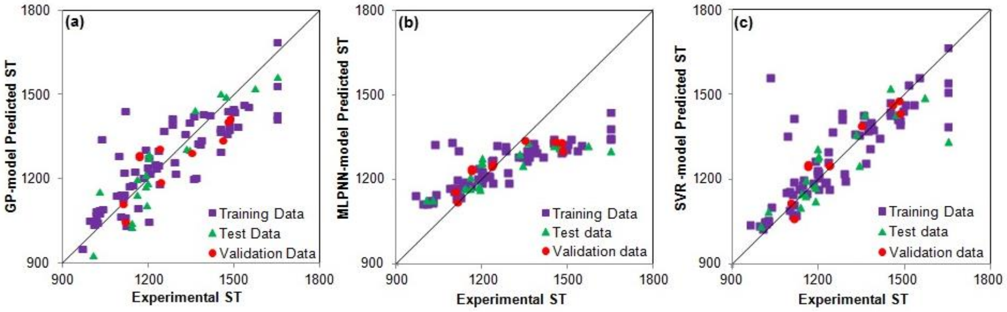


Figure 5.7: Plots pertaining to experimental *ST* values vis-à-vis those predicted by the GP (panel *a*), MLPNN (panel *b*), and SVR based (panel *c*) models. Hemispherical temperature (*HT*) prediction models.

The observations made from the *CC*, *RMSE* and *MAPE* values of the *HT* predictions by the three CI-based models are given below.

- The *CC* value in respect of the *HT* predictions by the GP-model considering the training set data (termed “recall” ability) is lower (0.803) than that of the corresponding MLPNN (0.926) and SVR (0.813) based models. The *CC* values in respect of the GP-model predictions for the test and validation sets (0.929, 0.953) are however higher than that for the MLPNN (0.894, 0.851) and SVR (0.800, 0.804) based models. Higher *CC* values corresponding to the test and validation set data are suggest better generalization capability of the model, which is crucial for correctly predicting the *HT* values for a completely new set of inputs/predictors (that is those which are not part of the training set data). It is thus clear that the GP model for *HT* prediction has better generalization capability than the MLPNN- and SVR-based models. This conclusion is also supported by the parity plots depicted in Figure 5.8 where it is seen that although the predictions of the training set outputs by the GP model (shown by the “square” symbol) exhibit a higher scatter relative to the *HT* predictions by the MLPNN and SVR models, the GP-model predictions pertaining to the test and validation set data show lower scatter (better generalization capability) than the predictions by the remaining two models.

- The parity plots corresponding to predictions of the hemispherical temperature show that the MLPNN-based model has produced better prediction and generalization performance than the GP- and SVR-based models. The predictions of *HT* by MLPNN model have also resulted in the higher (lower) magnitudes of the correlation coefficient (*RMSE/MAPE*) compared to the two other CI-based models.

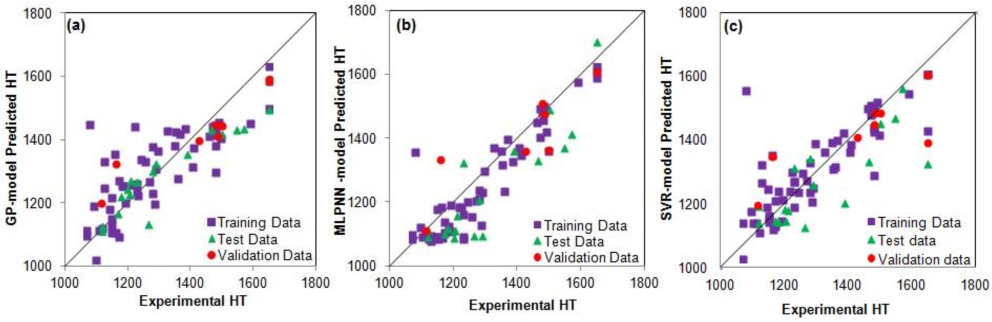


Figure 5.8: Plots pertaining to experimental *HT* values vis-à-vis those predicted by the GP (panel *a*), MLPNN (panel *b*), and SVR based (panel *c*) models.

(c) Fluid Temperature (*FT*) predicting models

In general, the trends exhibited by the *CC*, *RMSE* and *MAPE* values computed using the *FT* predictions made by the three CI-based models are similar to those for the prediction of the hemispherical temperature, *HT*. Specifically, it is witnessed that although the recall capability of the GP-based model is a tad inferior to that of the MLPNN and SVR models, its much-desired generalization capability is better than the stated two models. In Figure 5.9, it can be seen that the parity plots shown by the GP-model is also witnessed this trend of better generalization. In this figure, it is clearly realized that the scatter in the points corresponding to the predictions in respect of the test and validation set data by the GP-based model is lesser than that for the corresponding predictions by the MLPNN- and SVR-based models.

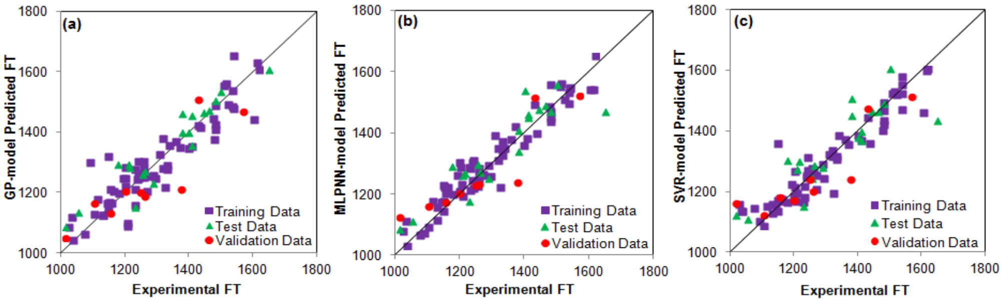


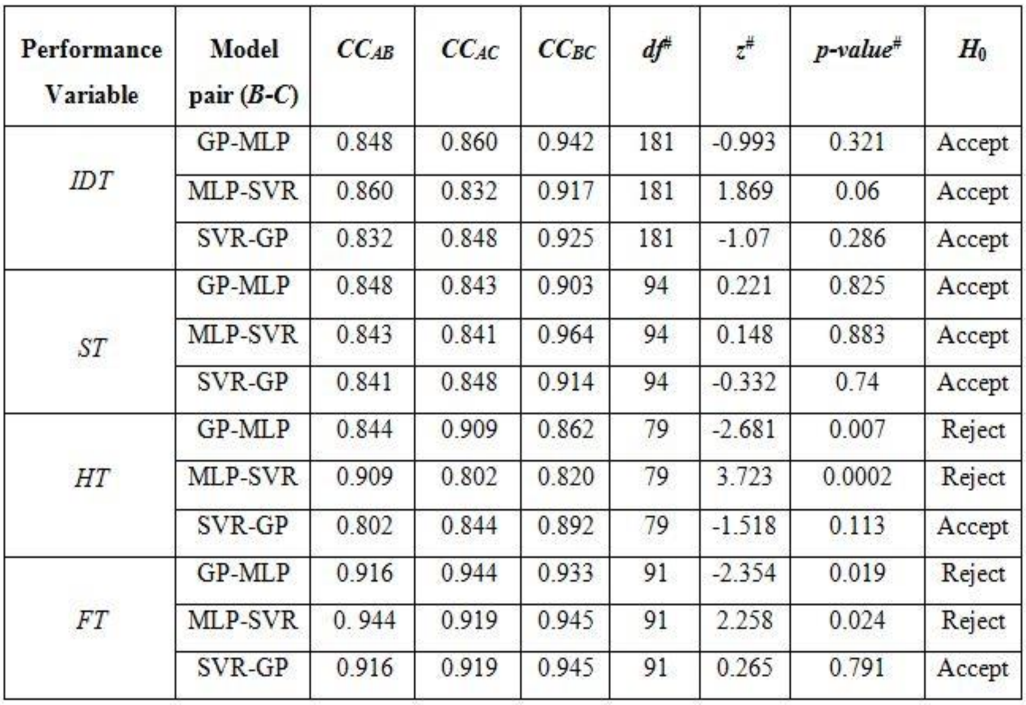
Figure 5.9: Plots pertaining to experimental *FT values* vis-à-vis those predicted by the GP (panel *a*), MLP (panel *b*), and SVR based (panel *c*) models.

For statistically comparing the AFT prediction accuracy and generalization performance of the CI-based models, the rigorous Steiger's Z-test (Steiger 1980) was also performed. This test scrutinizes whether the two correlation coefficients computed using the predictions of the same entity by two competing models are statistically equal. In particular, it tests the null hypothesis (H_0) that statistically two *CC* magnitudes are equal, i.e., CC_{AB} = CC_{AC}. In the present study subscripts, *A*, *B,* and *C*, respectively refer to the experimental AFT magnitudes and those predicted by the models *B* and *C*. In the case of CI-based models, the Steiger's Z-test has been performed to examine the validity of the null hypothesis (CC_{AB} = CC_{AC}) in respect of the following three pairs of AFT magnitudes.

1. [Experimental AFT (*A*) – GP model predicted AFT (*B*)] & [Experimental AFT (*A*) – MLPNN model predicted AFT (*C*)]
2. [Experimental AFT (*A*) – MLPNN model predicted AFT (*B*)] & [Experimental AFT (*A*) - SVR model predicted AFT (*C*)]
3. [Experimental AFT (*A*) –SVR model predicted AFT (*B*)] & [Experimental AFT (*A*) – GP model predicted AFT (*C*)]

The results of the Steiger's Z-test test are tabulated in **Table 5.6** and indicate the following: (a) The performance of the GP-, MLPNN-, and SVR-based models pertaining to the predictions of the *IDT* and *ST* values are comparable, and (b) In the case of the models pertaining to the predictions of *HT* and *FT*, the performance of the MLP- based models is better than that of the GP- and SVR-based models.

Table 5.6: Results of the Steiger's Z-test (testing the null hypothesis H_0, $CC_{AB} = CC_{AC}$)



Performance Variable	Model pair (*B*-*C*)	CC_{AB}	CC_{AC}	CC_{BC}	*df*#	*z*#	*p-value*#	H_0
IDT	GP-MLP	0.848	0.860	0.942	181	-0.993	0.321	Accept
	MLP-SVR	0.860	0.832	0.917	181	1.869	0.06	Accept
	SVR-GP	0.832	0.848	0.925	181	-1.07	0.286	Accept
ST	GP-MLP	0.848	0.843	0.903	94	0.221	0.825	Accept
	MLP-SVR	0.843	0.841	0.964	94	0.148	0.883	Accept
	SVR-GP	0.841	0.848	0.914	94	-0.332	0.74	Accept
HT	GP-MLP	0.844	0.909	0.862	79	-2.681	0.007	Reject
	MLP-SVR	0.909	0.802	0.820	79	3.723	0.0002	Reject
	SVR-GP	0.802	0.844	0.892	79	-1.518	0.113	Accept
FT	GP-MLP	0.916	0.944	0.933	91	-2.354	0.019	Reject
	MLP-SVR	0. 944	0.919	0.945	91	2.258	0.024	Reject
	SVR-GP	0.916	0.919	0.945	91	0.265	0.791	Accept

# *df:* degrees of freedom*; z:* test statistics*; p:* significance level

The overall inferences that can be drawn from the results obtained above are as follows. Depending upon the relationship that exists between the oxide weight percentages (model inputs) and the corresponding AFT (model output), the GP formalism has an ability to search and optimize an appropriate linear or a nonlinear input-output model. The nonlinear forms fitted by the GP for predicting all four AFTs (Eqs. 5.7, 5.13, 5.18 and 5.24) are revealing that the relationships between the individual AFTs and concentrations of oxides in the coal ashes are nonlinear.

- A comparison of the performance of the CI-based models developed in this study with the currently available high performing ones with relatively wider applicability (Seggiani and Pannocchia, 2003) show that:
    - ➢ The GP and MLPNN based models pertaining to the predictions of *IDT*, *ST*, *HT*, and *FT* have outperformed the corresponding currently available ones in terms of having better generalization ability.

  - The GP and MLPNN based models require lower number of inputs/predictors (= 7) than those needed by the models proposed by Seggiani and Pannocchia (2003) that consider 13, 11, 11 and 12 predictors, respectively for the prediction of *IDT*, *ST*, *HT*, and *FT*. The GP-, MLPNN-, and SVR-based models have used lower number of inputs/predictors, that reduced the effort and computational cost involved in compiling the input/predictor data.

- In the present study, the root means square error (*RMSE*), which has the same units as the specific AFT measures the closeness between the models predicted AFT magnitudes and the corresponding experimentally measured ones. It is an absolute measure of how well the model has fitted the example data and can be interpreted as the standard deviation of the unexplained variance. It is known that the reproducibility of the AFT magnitudes determined using the same instrument and measured by the same analyst differs by 30-50 °C. The corresponding variation between the measurements done at different laboratories is ~50-80 °C (Seggiani and Pannocchia, 2003). In Table 5.3, it is observed that the *RMSE* magnitudes pertaining to the *IDT*, *ST*, *HT*, and *FT* predictions by the GP-based models for the test and validation set data vary between 33.79 and 82.20. These magnitudes are essentially a measure of the generalization ability of the GP-based models. Considering the extent of the inherent variability in the experimental measurements of the AFT values, the stated *RMSE* magnitudes can be considered as reasonable and, therefore, indicative of good prediction and generalization performance of the GP-based models.
- Owing to their compact size (parsimonisity), and ease of evaluation, the GP-based models are more convenient to use and deploy in the practical applications. Notwithstanding, the stated positive attribute of the GP-based models, in situations when the highest AFT prediction accuracy is required, the MLPNN-based models should be used preferentially for the prediction of *HT* and *FT*.

5.6 CONCLUSION

In the coal-based processes such as combustion, gasification and liquefaction, the mineral matter in the coal, predominantly consisting of metal oxides, is left as an incombustible residue termed *ash*. Often, in the stated processes the ash deposits on the heat absorbing surfaces of the exposed process equipment and lead to the undesirable slagging or fouling phenomena. This has adverse implications for the process equipment such as corrosion of the furnace components, pressure drop in the heat transfer equipment, burning of channels, and an overall unstable operation. There are four attributes namely *Initial deformation temperature* (IDT), *Softening temperature* (ST), *Hemispherical temperature* (HT) and *Fluid temperature* (FT) that form the important proerties of the coal ashes. These are comprehensively termed as *ash fusion temperatures* (AFTs) and indicate the temperature range over which the ash deposits are formed on the heat absorbing surfaces of the process equipment. The presently available models pertaining to the predictions of AFT are mainly linear although a detailed data analysis suggests that nonlinear models are more appropriate. Also, the existing models are applicable to coals belonging to a limited number of geographical regions and, therefore, do not have wider applicability.

To address the above-stated deficiencies of the prevailing AFT prediction models, this study has developed nonlinear models using three computational intelligence (CI) based exclusively data-driven formalisms, namely, genetic programming (GP), multi-layer perceptron neural network (MLP), and support vector regression (SVR). Additionally, a large dataset of coals ashes from multiple countries/geographies has been employed in the development of CI-based models. As a result, the models developed have a wider applicability than the existing models. In this study, the prediction accuracy and generalization performance of the three CI-based models has been rigorously compared. The results of this comparison are given below.

- In general, the performance of the GP-, MLPNN- and SVR-based models pertaining to the predictions of *IDT* and *ST* magnitudes is comparable.

- For predicting *HT* and *FT* magnitudes the MLPNN-based models are better suited than the GP- and SVR-based models.
- Owing to their parsimonious (less complex) nature, the GP-based models are simple to understand and easy to apply.

Since they are capable with good AFT prediction accuracy and generalization performance with wider applicability, the CI-based models developed in this study have a potential to be the preferred ones for predicting AFT magnitudes of coal ashes from different geographies in the world. There were only a few observations pertaining to high ash coals in the data used in building the AFT models and therefore the models may not make accurate predictions of AFTs of coals containing high percentage of ash.

5.7 REFERENCES

Alpern B., Nahuys J., Martinez L. (1984). Mineral Matter in ashy and non-washable coals – its influence on chemical properties, *Comun Serv Geol Portugal*, **70(2)**, pp.299–317.

Bryers R.W. (1996). Fireside slagging, fouling, and high-temperature corrosion of heat-transfer surface due to impurities in steam raising fuels, *Prog Energy Combus Sci*, **22**, pp. 29-120.

Gao F, Han P, Zhai Y.J., Chen L.X. (2011). Application of support vector machine and ant colony algorithm in optimization of coal ash fusion temperature, *Proceedings of the 2011 International Conference on Machine Learning and Cybernetics*, Guilin, 10-13 July, 2011.

Geladi P., Kowalski B.R. (1986). Partial least squares regression (PLS): A tutorial, *Anal Chim Acta*, **85**, pp. 1–17.

Goni C, Helle S, Garcia X, Gordon A et al (2003) Coal blend combustion: Fusibility ranking from mineral matter composition, *Fuel*, **82(15-17)**, pp.2087-2095.

Gray V.R. (1987). Prediction of ash fusion temperature from ash composition for some New Zealand coals, *Fuel*, **66(9)**, pp. 1230-1239.

Huggins F.E., Kosmack D.A., Huffman G.P. (1981). Correlation between ash-fusion temperatures and ternary equilibrium phase diagrams, *Fuel*, **60(7)**, pp.577-584.

Hurst H.J., Novak F., Patterson J.H. (1996). Phase diagram approach to the fluxing effect of additions of $CaCO_3$ on Australian coal ashes, *Energy Fuels*, **10(6)**, pp.1215–1219.

Jak E. (2002). Prediction of coal ash fusion temperatures with the F*A*C*T thermodynamic computer package, *Fuel*, **81(13)**, pp.1655–1668.

Kahraman H., Bos F., Reifenstein A., Coin C.D. (1998). Application of a new ash fusion test to Theodore coals, *Fuel*, **77(9-10)**, pp.1005-1011.

Karimi S, Jorjani E, Chelgani S.C., Mesroghli S. (2014). Multivariable regression and adaptive neurofuzzy inference system predictions of ash fusion temperatures using ash chemical composition of US coals, *Journal of Fuels*, **2014**, Article ID 392698. DOI:10.1155/2014/392698.

Kucukbayrak S., Ersoy-Mericboyu A., Haykiri-Acma H., Guner H., Urkan K. (1993). Investigation of the relation between chemical composition and ash fusion temperatures for soke Turkish lignites, *Fuel Sci. Technol. International*, **11(9)**, pp. 1231-1249. https://doi.org/10.1080/08843759308916127.

Liu Y.P., Wu M.G., Qian J.X. (2007). Predicting coal ash fusion temperature based on its chemical composition using ACO-BP neural network, *Thermochim Acta*, **454(1)**, pp.64-68.

Lolja S.A., Haxhi H., Dhimitri R., Drushku S., Malja A. (2002). Correlation between ash fusion temperatures and chemical composition in Albanian coal ashes, *Fuel*, **81(17)**, pp.2257-2261.

Lloyd W.G., Riley J.T., Zhou S., Risen M.A., Tibbitts R.L. (1995). Ash fusion temperatures under oxidizing conditions, *Energy Fuels*, **7(4)**, pp. 490-494.

Miao S., Jiang Q., Zhou H., Shi J., Cao Z. (2016). Modelling and prediction of coal ash fusion temperature based on BP neural network, *MATEC Web of Conferences 40*, EDP Sciences, https://doi.org/10.1051/matecconf/20164005010.

Mierswa I., Wurst M., Klinkenberg R., Scholz M., Euler T. (2006). Rapid prototyping for complex data mining tasks, In: *Proceeding of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*, pp.935–940. Doi:10.1145/1150402.1150531.

Ozbayoglu G., Ozbayoglu M.E. (2006). A new approach for the prediction of ash fusion temperatures: A case study using Turkish Lignites, *Fuel*, **85(4)**, pp.545-552. https://doi.org/10.1016/j.fuel.2004.12.020.

Patterson J.H., Hurst H.J. (2000). Ash and slag qualities of Australian bituminous coals for use in slagging gasifiers, *Fuel*, **79(13)**, pp.1671-1678.

RapidMiner, 2007. https://rapidminer.com/products/studio.

Rhinehart R.R., Attar A.A. (1987). Ash fusion temperature: A thermodynamically-based model, *Am Soc Mech Engg.*, **8**, pp.97-101.

Schmidt M., Lipson H. (2009). Distilling free-form natural laws from experimental data, *Science*, **324(5923)**, pp. 81–85.

Sharma S., Tambe S.S. (2014). Soft-sensor development for biochemical systems using genetic programming, *Biochemical Engineering Journal*, **85**, pp. 89–100.

Seggiani M. (1999). Empirical correlations of the ash fusion temperatures and temperature of critical viscosity for coal and biomass ashes, *Fuel*, **78(9)**, pp. 1121-1125.

Seggiani M., Pannocchia G. (2003). Prediction of coal ash thermal properties using Partial Least-Squares Regression, *Ind Eng Chem Res*, **42(20)**, pp. 4919-4926. DOI: 10.1021/ie030074u.

Sharma S., Tambe S.S. (2014). Soft-sensor development for biochemical systems using genetic programming, *Biochemical Engineering Journal*, **85**, pp. 89–100.

Skrifvars B.J., Lauren T., Hupa M., Korbee R. , Ljung P. (2004) Ash behavior in a pulverized wood fired boiler-a case study, *Fuel*, **83(10)**, pp. 1371-1379.

Slegeir W.A., Singletary J.H., Kohut J.F. (1988). Application of a microcomputer to the determination of coal ash fusibility characteristics, *J. Coal Quality*, 7, pp. 248–254.

Song W.J., Tang L.H., Zhu X.D., Wu Y.Q., Zhu Z.B. (2010). Effect of coal ash composition on ash fusion temperatures, *Energy Fuels*, **24(1)**, pp.182–189.

Steiger J.H. (1980). Tests for comparing elements of a correlation matrix, *Psychological Bulletin*, **87(2)**, pp. 245-251.

Van Dyk J.C., Keyser M.J., Van Zyl J.W. (2001). Suitability of feed-stocks for the Sasol–Lurgi fixed bed dry bottom gasification process, In: *Gasification Technology Conference*, Gasification Technologies Council, Arlington, VA, USA, pages 11.

Vassilev S.V., Kitano K., Takeda S., Tsurue T. (1995). Influence of mineral and chemical composition of coal ashes on their fusibility, *Fuel Processing Technology*, **45(1)**, pp. 27-51.

Vorres K.S. (1979). Effect of composition on melting behaviour of coal ash, *Journal of Engg. Power*, **101(4)**, pp. 497-499.

Wall T.F., Creelman R.A., Gupta R.P. (1998). Coal ash fusion temperatures—New characterization techniques, and implications for slagging and fouling, *Prog Energy Combus Sci*, **24**, pp. 345-353.

Winegartner E.C., Rhodes B.T. (1975). An empirical study of the relation of chemical properties to ash fusion temperatures, *J Eng Power*, **97**, pp. 395-406.

Yin C., Luo Z., Ni M., Cen K. (1998). Predicting coal ash fusion temperature with a back-propagation neural network model, *Fuel*, **77(15)**, pp. 1777-1782.

Zhao B., Zhang Z., Wu X. (2010). Prediction of coal ash fusion temperature by least squares support vector machine model, *Energy Fuels*, **24(5)**, pp.3066-3071.

APPENDIX 5

Table 5.A: Experimental data consisting of the oxide composition (wt %) and the corresponding magnitudes of *Initial Deformation Temperature* (IDT)

SiO_2	**Al_2O_3**	**Fe_2O_3**	**CaO**	**MgO**	**TiO_2**	**K_2O + Na_2O**	**IDT**	**Reference[§]**
30.884	17.236	19.829	19.116	7.256	0.613	4.995	1050	Lolja et al. (2002)
52.416	16.490	12.089	10.585	3.619	0.852	3.909	1090	Lolja et al. (2002)
40.114	13.455	23.607	11.035	5.904	0.713	5.101	1040	Lolja et al. (2002)
43.397	18.920	10.768	16.236	3.904	0.767	5.940	1100	Lolja et al. (2002)
42.405	16.473	21.543	7.415	5.862	0.772	5.421	1070[*]	Lolja et al. (2002)
50.145	23.544	11.842	4.482	5.505	0.672	3.690	1080	Lolja et al. (2002)
51.323	23.953	10.834	3.468	5.201	0.792	4.330	1080[*]	Lolja et al. (2002)
56.241	20.719	10.239	3.553	2.562	0.971	5.635	1080	Lolja et al. (2002)
53.195	13.602	13.000	12.328	4.895	0.612	2.217	1060	Lolja et al. (2002)
42.130	19.808	13.185	18.900	2.221	0.949	2.564	1100[#]	Lolja et al. (2002)
49.814	26.655	5.677	8.791	4.592	0.754	3.667	1140	Lolja et al. (2002)
43.826	14.992	14.131	16.465	4.597	0.641	5.338	1080	Lolja et al. (2002)
50.511	16.693	8.858	12.976	4.800	0.611	5.521	1100	Lolja et al. (2002)
54.683	28.810	3.954	2.723	6.259	0.802	2.768	1320	Qiu et al. (1999)
39.131	26.327	11.063	13.829	6.587	0.697	2.366	1200	Qiu et al. (1999)
44.187	36.039	6.445	8.931	1.045	2.340	1.013	1335	Hanxu et al. (2006)
48.512	36.358	4.861	5.840	0.772	2.019	1.638	1450[*]	Hanxu et al. (2006)
51.364	33.730	8.632	1.405	0.636	1.548	2.686	1425[#]	Hanxu et al. (2006)
48.019	23.324	23.171	1.321	0.863	1.016	2.286	1110	Vorres (1979)
39.315	21.005	30.583	4.439	1.306	0.846	2.507	1110	Vorres (1979)
49.916	18.771	21.145	6.040	1.071	0.819	2.237	1093	Vorres (1979)
49.231	11.795	7.179	25.641	4.103	0.615	1.436	1126	Vorres (1979)
38.847	20.250	26.129	7.544	1.252	0.894	5.085	917.77[#]	Rees (1964)
35.846	20.592	25.327	8.520	3.250	0.838	5.628	956.66	Rees (1964)
35.704	20.633	26.501	8.095	2.306	0.961	5.800	933.88[#]	Rees (1964)
43.143	18.145	23.738	8.361	1.053	0.901	4.658	932.77[*]	Rees (1964)
46.622	19.034	23.186	4.246	1.025	0.994	4.894	924.44[#]	Rees (1964)
44.194	17.553	28.286	3.022	1.035	1.014	4.895	957.22[#]	Rees (1964)
35.004	16.109	31.145	13.426	0.970	0.776	2.569	909.44[*]	Rees (1964)
39.296	14.465	34.984	6.847	0.931	0.802	2.675	956.11	Rees (1964)
44.525	17.275	23.643	7.906	1.007	1.051	4.592	957.22	Rees (1964)
40.157	20.311	22.240	9.248	1.713	0.999	5.333	909.44	Rees (1964)
47.261	19.653	16.650	11.832	0.863	0.829	2.913	1036.11	Rees (1964)
47.763	20.072	12.984	14.325	0.937	0.913	3.006	986.11	Rees (1964)
40.302	18.205	21.082	16.142	0.684	0.861	2.724	991.11	Rees (1964)
35.282	20.400	30.310	9.430	0.765	0.972	2.841	1080.55[*]	Rees (1964)

34.296	19.782	30.330	11.278	0.696	1.101	2.516	1077.77	Rees (1964)
52.643	23.955	13.666	4.378	0.865	1.074	3.419	1147.77	Rees (1964)
42.591	17.366	14.806	20.744	0.901	0.771	2.821	1071.66	Rees (1964)
43.258	17.337	12.924	21.386	1.094	0.681	3.319	1114.44*	Rees (1964)
54.701	21.674	13.660	4.498	1.044	1.002	3.423	1147.77*	Rees (1964)
52.474	22.708	14.720	5.149	0.939	1.087	2.923	1114.44	Rees (1964)
44.724	22.379	21.963	4.755	1.310	1.145	3.723	1155.55	Rees (1964)
55.775	28.160	7.209	1.444	1.282	1.363	4.766	1237.22	Rees (1964)
52.461	24.388	6.605	9.755	1.050	1.225	4.517	1089.44*	Rees (1964)
46.978	23.008	12.504	11.324	1.869	0.634	3.683	1038.88	Rees (1964)
46.580	22.947	14.976	9.473	2.382	0.674	2.969	1038.88	Rees (1964)
50.729	22.728	13.345	6.974	1.839	0.676	3.709	1141.66*	Rees (1964)
41.309	25.077	16.253	12.201	1.224	0.971	2.966	1149.44	Rees (1964)
49.260	22.622	19.207	3.499	0.814	1.268	3.330	1160.00	Moreno (2008)
48.412	20.275	20.460	7.904	0.119	0.998	1.832	1115.55*	Moreno (2008)
40.046	19.550	32.132	2.806	0.652	0.904	3.910	1093.33	Moreno (2008)
55.435	29.577	6.752	2.265	0.790	1.612	3.571	1482.22	Moreno (2008)
57.772	29.465	4.044	2.117	1.443	1.411	3.749	1482.22	Moreno (2008)
62.087	25.120	4.792	3.453	1.116	0.956	2.476	1348.88*	Moreno (2008)
52.242	23.494	9.420	6.850	1.981	0.917	5.095	1260.00	Moreno (2008)
63.583	22.641	6.445	1.908	1.397	0.897	3.129	1326.66	Moreno (2008)
64.084	25.679	4.937	0.539	0.394	1.203	3.163	1482.22	Moreno (2008)
50.117	19.456	20.780	5.277	0.835	0.907	2.628	1037.77	Moreno (2008)
45.335	17.165	24.208	8.891	0.770	0.825	2.806	957.222	Moreno (2008)
59.037	22.009	5.412	7.121	3.470	1.631	1.321	1004.44*	Moreno (2008)
48.642	19.057	5.159	14.214	4.759	1.263	6.907	1043.33	Moreno (2008)
43.086	19.293	6.588	17.258	3.679	1.370	8.727	1093.33	Moreno (2008)
39.008	19.441	20.057	14.823	4.515	0.787	1.368	1168.88	Moreno (2008)
43.193	32.753	18.731	1.842	1.024	2.149	0.307	1280.00#	Bryant et al. (2000)
50.563	32.344	12.078	1.842	1.024	1.842	0.307	1310.00	Bryant et al. (2000)
62.680	22.990	5.773	3.299	1.237	0.825	3.196	1240.00	Bryant et al. (2000)
62.613	26.533	4.422	0.402	0.804	1.206	4.020	1200.00	Bryant et al. (2000)
45.120	13.723	17.894	11.308	9.178	0.703	1.789	1160.00	Akiyama et al. (2011)
65.871	23.982	4.796	1.131	0.827	1.435	1.927	1334.00	Akiyama et al. (2011)
52.545	25.440	11.813	5.133	1.846	1.665	1.526	1220.00	Steyn and Minnitt (2010)
49.697	25.608	11.306	9.919	2.763	0.691	0.015	1174.00*	Bahrinet al. (2009)
36.782	23.103	18.161	14.713	2.299	2.184	2.759	1195.00	Matuszewski (2012)
46.667	25.104	18.021	4.375	1.667	1.354	2.813	1200.00	Matuszewski (2012)
44.572	23.779	24.270	3.412	0.746	1.098	2.122	1237.77	Matuszewski (2012)
47.619	19.048	21.164	7.407	1.058	1.058	2.646	1201.11	Matuszewski (2012)
38.725	18.899	6.029	24.928	7.420	1.391	2.609	1187.77	Matuszewski (2012)
45.491	19.981	7.715	19.778	5.076	0.860	1.099	1225.55	Matuszewski (2012)
43.896	15.388	6.410	18.017	8.276	0.726	7.287	1118.33#	Matuszewski (2012)
51.773	17.645	10.519	13.982	3.135	1.209	1.737	1177.77	Matuszewski (2012)
58.438	28.861	5.823	0.512	0.205	2.487	3.674	1504.44	U.S.DOE (2001)

59.553	28.515	5.429	0.110	0.669	2.040	3.685	1393.33$^{\#}$	U.S.DOE (2001)
56.539	28.483	8.972	0.533	0.139	1.984	3.350	1241.66	U.S.DOE (2001)
54.065	26.209	15.229	0.484	0.165	1.564	2.284	1365.55	U.S.DOE (2001)
36.323	25.628	30.371	3.844	0.310	1.312	2.211	1065.00	U.S.DOE (2001)
64.036	11.751	13.044	4.420	1.919	1.843	2.986	1166.11	U.S.DOE (2001)
69.782	21.404	4.756	1.554	0.532	0.960	1.012	1414.00*	Kim et al. (1994)
54.398	20.033	6.718	13.448	2.519	0.906	1.978	1154.00*	Kim et al. (1994)
63.482	25.283	4.507	2.451	1.033	1.121	2.122	1373.00*	Kim et al. (1994)
55.254	20.434	15.221	4.379	1.512	1.043	2.158	1165.00	CEA Report (2012)
58.924	33.944	1.612	1.501	1.007	1.491	1.521	1320.00	CEA Report (2012)
74.467	16.322	4.080	1.632	1.224	0.755	1.520	1380.00	CEA Report (2012)
73.755	18.439	4.098	1.024	0.615	0.840	1.229	1350.00	CEA Report (2012)
59.024	31.583	4.636	0.879	0.455	2.020	1.384	1270.00$^{\#}$	CIMFR
59.572	28.740	6.426	1.508	0.442	1.649	1.649	1250.00*	CIMFR
58.722	23.656	8.929	2.939	1.281	1.139	3.285	1120.00	CIMFR
62.037	24.247	6.458	2.628	0.809	1.132	2.648	1200.00	CIMFR
61.681	26.187	5.630	1.340	0.453	1.612	3.072	1220.00	CIMFR
61.679	27.043	4.875	1.822	0.723	1.466	2.341	1200.00$^{\#}$	CIMFR
55.756	26.835	9.385	2.893	1.155	1.963	1.942	1180.00	CIMFR
66.078	24.923	4.023	1.633	0.575	1.129	1.613	1220.00	CIMFR
58.874	26.892	7.623	1.861	0.654	1.770	2.313	1180 .00$^{\#}$	CIMFR
51.477	34.204	5.620	4.233	1.327	1.609	1.508	1240.00	CIMFR
63.498	28.237	4.015	0.875	0.312	1.449	1.580	1210.00*	CIMFR
58.522	31.128	6.013	0.693	0.201	2.008	1.425	1280.00	CIMFR
62.270	28.659	2.415	1.449	0.403	1.610	3.160	1220.00*	CIMFR
60.889	27.816	6.838	0.232	0.121	2.095	1.994	1260.00	CIMFR
62.174	27.276	5.492	0.703	0.326	2.527	1.447	1220.00	CIMFR
59.872	22.687	10.204	2.403	1.227	2.086	1.442	1240.00*	CIMFR
62.217	27.065	6.475	1.165	0.253	1.702	1.094	1240.00*	CIMFR
66.754	26.760	2.222	0.434	0.323	1.536	1.950	1280.00$^{\#}$	CIMFR
61.149	29.563	4.834	1.467	0.354	1.214	1.396	1340.00	CIMFR
66.459	22.797	6.357	0.923	0.170	1.189	1.719	1320. 00	CIMFR
60.348	25.638	9.636	0.966	0.483	1.690	1.217	1220.00	CIMFR
59.162	24.158	12.801	0.401	0.110	2.085	1.273	1300.00*	CIMFR
52.000	27.691	15.048	0.642	0.102	3.179	1.294	1180.00	CIMFR
61.773	27.762	6.800	0.230	0.100	2.283	1.042	1220.00	CIMFR
59.097	29.700	6.662	0.698	0.121	2.345	1.355	1300.00	CIMFR
61.380	24.597	8.482	1.164	0.455	2.753	1.134	1260.00	CIMFR
61.375	28.189	6.897	0.609	0.264	1.625	0.995	1200.00	CIMFR
60.646	29.900	4.825	1.481	0.655	1.451	1.017	1240.00	CIMFR
58.201	26.385	11.045	0.865	0.744	1.851	0.895	1180.00	CIMFR
59.703	26.735	9.460	0.666	0.474	1.896	1.039	1220.00	CIMFR
64.371	26.411	6.395	0.170	0.080	1.441	1.111	1260.00	CIMFR
62.428	25.386	8.074	0.992	0.223	1.862	1.002	1200.00	CIMFR
64.258	27.446	4.431	0.696	0.151	1.938	1.060	1240.00*	CIMFR

62.412	27.348	6.018	0.713	0.482	1.849	1.175	1200.00	CIMFR
64.787	27.848	4.013	0.231	0.181	1.529	1.388	1230.00$^{\#}$	CIMFR
62.409	24.388	8.400	1.103	0.381	1.524	1.784	1230.00	CIMFR
60.813	26.803	8.032	0.634	0.272	1.610	1.822	1250 .00$^{\#}$	CIMFR
65.086	26.581	3.047	1.172	0.387	1.712	1.957	1260.00	CIMFR
61.844	27.247	6.635	0.111	0.101	2.094	1.943	1250.00	CIMFR
62.853	24.824	8.007	0.110	0.110	1.525	2.549	1210.00*	CIMFR
63.143	24.478	6.061	1.346	0.557	1.619	2.762	1240.00	CIMFR
60.756	24.515	8.081	2.015	0.638	1.499	2.461	1240.00	CIMFR
65.463	24.127	4.803	1.675	0.441	1.685	1.805	1200.00	CIMFR
59.049	23.815	12.881	0.989	0.424	1.937	0.878	1180.00*	CIMFR
59.628	19.071	9.105	7.269	1.960	1.825	0.996	1180.00	CIMFR
64.442	22.992	5.669	2.373	0.791	1.298	2.394	1100.00	CIMFR
60.632	21.901	8.621	3.518	1.132	1.646	2.438	1140.00*	CIMFR
61.651	24.333	4.905	3.912	1.239	1.802	2.069	1200.00	CIMFR
64.143	22.793	6.878	1.894	0.881	1.621	1.742	1200.00	CIMFR
64.978	24.714	6.435	0.493	0.211	1.611	1.541	1220.00	CIMFR
65.837	26.132	2.032	1.778	0.691	1.463	2.022	1220.00	CIMFR
63.642	23.285	5.287	4.176	1.151	1.141	1.263	1200.00*	CIMFR
63.051	26.205	5.473	0.142	0.061	1.868	3.137	1200.00$^{\#}$	CIMFR
62.699	26.033	4.675	0.173	0.092	1.793	4.471	1240.00	CIMFR
68.281	21.807	6.095	0.326	0.142	1.628	1.659	1180.00	CIMFR
66.554	22.881	6.513	0.194	0.061	1.631	2.100	1180.00	CIMFR
59.795	14.665	11.745	6.786	1.544	1.849	3.341	1140.00	CIMFR
64.180	20.761	9.261	0.494	0.212	1.614	3.450	1220.00*	CIMFR
66.549	20.515	8.037	1.178	0.423	1.289	1.984	1190.00	CIMFR
60.551	28.212	6.029	1.157	0.584	1.771	1.671	1200.00*	CIMFR
70.551	19.243	5.679	0.843	0.356	1.626	1.656	1160.00	CIMFR
59.059	25.008	7.058	4.781	1.414	1.996	0.509	1190.00	CIMFR
61.359	21.744	6.941	2.965	0.961	1.963	3.997	1140.00	CIMFR
63.844	23.608	4.467	0.997	0.234	1.628	5.180	1180.00	CIMFR
64.176	17.360	10.186	1.704	0.439	1.960	4.113	1180.00	CIMFR
67.294	22.025	4.022	0.605	0.222	1.290	4.526	1220	CIMFR
68.401	19.043	4.842	1.304	0.303	1.456	4.629	1200	CIMFR
66.080	25.947	4.032	0.788	0.232	1.374	1.516	1200	CIMFR
66.713	19.129	10.085	1.157	0.364	1.455	1.081	1200	CIMFR
67.061	25.022	4.399	0.802	0.170	1.603	0.912	1220	CIMFR
63.661	27.378	5.631	0.342	0.081	1.934	0.957	1280	CIMFR
66.151	25.268	4.807	0.331	0.130	1.766	1.525	1300	CIMFR
61.503	26.212	6.835	1.913	0.221	1.852	1.450	1260	CIMFR
65.106	27.388	3.212	0.695	0.191	1.571	1.822	1270	CIMFR
66.403	23.872	5.110	1.270	0.881	1.311	1.086	1240	CIMFR
66.865	25.034	3.637	1.084	0.537	1.499	1.307	1250*	CIMFR
59.201	23.128	10.173	3.354	1.019	1.182	1.876	1160	CIMFR
59.258	25.624	8.606	2.249	0.709	1.273	2.188	1260	CIMFR

66.678	26.639	3.209	0.523	0.252	1.449	1.227	1280[#]	CIMFR
60.513	25.337	9.311	1.521	0.639	1.542	1.095	1220[*]	CIMFR
62.306	28.681	4.042	1.398	0.760	1.621	1.155	1270	CIMFR
62.540	26.214	4.717	2.785	0.678	1.809	1.171	1280[*]	CIMFR
63.780	25.452	5.422	0.755	0.332	1.207	3.038	1220[*]	CIMFR
64.295	24.649	5.022	1.157	0.352	1.449	3.060	1230	CIMFR
64.696	26.057	4.198	0.291	0.230	1.362	3.146	1240	CIMFR
63.425	24.386	6.891	1.898	0.700	1.624	1.035	1220[*]	CIMFR
63.197	23.990	7.679	1.682	0.567	1.783	1.064	1180	CIMFR
60.103	28.005	5.837	2.611	1.008	1.371	1.028	1250[*]	CIMFR
61.349	28.980	4.226	1.826	0.857	1.533	1.190	1240	CIMFR
64.139	25.369	6.458	0.930	0.344	1.455	1.273	1230[*]	CIMFR
61.475	28.722	4.468	2.209	0.774	1.303	1.008	1240[#]	CIMFR

* Test data; # Validation data; [§] CIMFR: C*entral Institute of Mining and Fuel Research*, Dhanbad, India from where data were sourced.

APPENDIX 5

Table 5.B: Experimental data consisting of the oxide composition (wt %) and the corresponding magnitudes of *Softening Temperature* (ST)

SiO_2	**Al_2O_3**	**Fe_2O_3**	**CaO**	**MgO**	**TiO_2**	**K_2O + Na_2O**	**ST**	**Reference**
48.40	25.50	3.50	2.41	5.54	0.71	2.45	1350.00[#]	Qiu et al. (1999)
38.20	25.70	10.80	13.50	6.43	0.68	2.31	1380.00	Qiu et al. (1999)
42.30	34.50	6.17	8.55	1.00	2.24	0.97	1360.00[*]	Hanxu et al. (2006)
42.00	36.90	3.21	1.93	0.44	2.08	0.54	1451.00	Hanxu et al. (2006)
47.10	35.30	4.72	5.67	0.75	1.96	1.59	1500.00	Hanxu et al. (2006)
50.10	32.90	8.42	1.37	0.62	1.51	2.62	1495.00	Hanxu et al. (2006)
47.27	22.96	22.81	1.30	0.85	1.00	2.25	1343.00[*]	Vorres (1979)
37.64	20.11	29.28	4.25	1.25	0.81	2.40	1190.00	Vorres (1979)
47.52	17.87	20.13	5.75	1.02	0.78	2.13	1182.00	Vorres (1979)
34.76	18.12	23.38	6.75	1.12	0.80	4.55	993.88	Rees (1964)
32.10	18.44	22.68	7.63	2.91	0.75	5.04	1016.66	Rees (1964)
31.58	18.25	23.44	7.16	2.04	0.85	5.13	1005.55	Rees (1964)
39.73	16.71	21.86	7.70	0.97	0.83	4.29	1013.33	Rees (1964)
44.58	18.20	22.17	4.06	0.98	0.95	4.68	1024.44	Rees (1964)
42.70	16.96	27.33	2.92	1.00	0.98	4.73	1024.44	Rees (1964)
30.66	14.11	27.28	11.76	0.85	0.68	2.25	963.88	Rees (1964)
36.73	13.52	32.70	6.40	0.87	0.75	2.50	1001.66[*]	Rees (1964)
41.11	15.95	21.83	7.30	0.93	0.97	4.24	1024.44	Rees (1964)
35.39	17.90	19.60	8.15	1.51	0.88	4.70	1015.00	Rees (1964)
42.18	17.54	14.86	10.56	0.77	0.74	2.60	1111.11	Rees (1964)
41.31	17.36	11.23	12.39	0.81	0.79	2.60	1101.11	Rees (1964)
34.18	15.44	17.88	13.69	0.58	0.73	2.31	1101.11	Rees (1964)
32.29	18.67	27.74	8.63	0.70	0.89	2.60	1123.33	Rees (1964)
30.53	17.61	27.00	10.04	0.62	0.98	2.24	1118.88	Rees (1964)
50.50	22.98	13.11	4.20	0.83	1.03	3.28	1187.77	Rees (1964)
35.93	14.65	12.49	17.50	0.76	0.65	2.38	1114.44[*]	Rees (1964)
35.58	14.26	10.63	17.59	0.90	0.56	2.73	1138.88[*]	Rees (1964)
52.42	20.77	13.09	4.31	1.00	0.96	3.28	1163.33[#]	Rees (1964)
49.73	21.52	13.95	4.88	0.89	1.03	2.77	1163.33[#]	Rees (1964)
51.54	25.79	25.31	5.48	1.51	1.32	4.29	1222.77	Rees (1964)
55.24	27.89	7.14	1.43	1.27	1.35	4.72	1342.22	Rees (1964)
47.97	22.30	6.04	8.92	0.96	1.12	4.13	1195.00	Rees (1964)
42.98	21.05	11.44	10.36	1.71	0.58	3.37	1188.33	Rees (1964)
42.83	21.10	13.77	8.71	2.19	0.62	2.73	1156.11[*]	Rees (1964)
48.01	21.51	12.63	6.60	1.74	0.64	3.51	1201.66	Rees (1964)
40.09	17.54	18.60	11.16	1.03	1.00	2.25	1156.11[*]	Rees (1964)
41.59	18.64	22.48	7.10	1.20	1.00	2.15	1132.22	Rees (1964)
39.70	17.74	25.20	7.72	0.98	0.86	2.29	1105.55	Rees (1964)

39.14	23.76	15.40	11.56	1.16	0.92	2.81	1186.66*	Rees (1964)
46.60	21.40	18.17	3.31	0.77	1.20	3.15	1197.22#	Moreno (2008)
44.65	18.70	18.87	7.29	0.11	0.92	1.69	1148.88	Moreno (2008)
38.10	18.60	30.57	2.67	0.62	0.86	3.72	1115.55*	Moreno (2008)
52.63	28.08	6.41	2.15	0.75	1.53	3.39	1482.22	Moreno (2008)
54.86	27.98	3.84	2.01	1.37	1.34	3.56	1482.22	Moreno (2008)
58.43	23.64	4.51	3.25	1.05	0.90	2.33	1382.22	Moreno (2008)
46.14	20.75	8.32	6.05	1.75	0.81	4.50	1293.33	Moreno (2008)
60.97	21.71	6.18	1.83	1.34	0.86	3.00	1371.11	Moreno (2008)
61.79	24.76	4.76	0.52	0.38	1.16	3.05	1482.22	Moreno (2008)
49.20	19.10	20.40	5.18	0.82	0.89	2.58	1112.77#	Moreno (2008)
41.20	15.60	22.00	8.08	0.70	0.75	2.55	1037.77*	Moreno (2008)
45.60	17.00	4.18	5.50	2.68	1.26	1.02	1093.33*	Moreno (2008)
46.20	18.10	4.90	13.50	4.52	1.20	6.56	1087.77*	Moreno (2008)
30.58	14.60	4.07	23.63	3.08	1.06	4.31	1160.00	Moreno (2008)
33.03	14.79	5.05	13.23	2.82	1.05	6.69	1110.00	Moreno (2008)
34.21	17.05	17.59	13.00	3.96	0.69	1.20	1200.00	Moreno (2008)
42.20	32.00	18.30	1.80	1.00	2.10	0.30	1470.00*	Bryant et al. (2000)
49.40	31.60	11.80	1.80	1.00	1.80	0.30	1570.00	Bryant et al. (2000)
60.80	22.30	5.60	3.20	1.20	0.80	3.10	1330.00	Bryant et al. (2000)
26.00	11.40	12.20	29.60	5.00	0.80	5.50	1200.00	Bryant et al. (2000)
49.24	23.84	11.07	4.81	1.73	1.56	1.43	1250.00	Steyn and Minnitt (2010)
32.80	12.40	24.70	16.50	12.30	0.26	0.35	1650.00	Bahrinet al. (2009)
79.70	18.49	0.39	0.57	0.07	0.24	0.53	1512.00*	Bahrinet al. (2009)
77.60	19.60	1.52	0.76	0.09	0.25	0.11	1650.00	Bahrinet al. (2009)
71.40	15.38	11.30	0.57	0.17	0.58	0.59	1281.00	Bahrinet al. (2009)
42.90	16.50	1.30	22.00	16.50	0.25	0.44	1650.00	Bahrinet al. (2009)
29.10	6.75	37.70	15.10	10.10	0.20	0.19	1356.00*	Bahrinet al. (2009)
49.10	25.30	11.17	9.80	2.73	0.68	0.02	1650.00	Bahrinet al. (2009)
37.60	34.73	1.73	6.15	5.43	6.68	5.77	1414.00	Bahrinet al. (2009)
39.24	43.62	1.45	5.40	3.21	3.29	2.69	1283.00	Bahrinet al. (2009)
41.18	30.53	1.73	7.17	4.75	8.32	4.32	1326.00	Bahrinet al. (2009)
36.26	41.52	3.90	6.67	5.00	3.58	1.81	1116.00	Bahrinet al. (2009)
23.58	46.98	3.36	11.73	7.00	4.41	1.66	1650.00*	Bahrinet al. (2009)
78.80	16.07	0.01	0.06	0.95	2.11	1.26	1032.00	Bahrinet al. (2009)
32.00	20.10	15.80	12.80	2.00	1.90	2.40	1226.00	Matuszewski (2012)
44.80	24.10	17.30	4.20	1.60	1.30	2.70	1232.00#	Matuszewski (2012)
41.80	22.30	22.76	3.20	0.70	1.03	1.99	1287.77	Matuszewski (2012)
45.00	18.00	20.00	7.00	1.00	1.00	2.50	1237.77	Matuszewski (2012)
33.40	16.30	5.20	21.50	6.40	1.20	2.25	1198.88	Matuszewski (2012)
38.09	16.73	6.46	16.56	4.25	0.72	0.92	1234.44	Matuszewski (2012)
35.06	12.29	5.12	14.39	6.61	0.58	5.82	1140.55@	Matuszewski (2012)
44.10	15.03	8.96	11.91	2.67	1.03	1.48	1237.77#	Matuszewski (2012)
57.10	28.20	5.69	0.50	0.20	2.43	3.59	1532.22	U.S.DOE (2001)
54.30	26.00	4.95	0.10	0.61	1.86	3.36	1357.22*	U.S.DOE (2001)

53.00	26.70	8.41	0.50	0.13	1.86	3.14	1475.55	U.S.DOE (2001)
52.54	25.47	14.80	0.47	0.16	1.52	2.22	1390.55	U.S.DOE (2001)
35.15	24.80	29.39	3.72	0.30	1.27	2.14	1143.33	U.S.DOE (2001)
59.40	10.90	12.10	4.10	1.78	1.71	2.77	1365.00	U.S.DOE (2001)
74.20	15.78	3.45	1.69	0.50	0.80	0.94	1457.00#	Kim et al. (1994)
66.90	20.52	4.56	1.49	0.51	0.92	0.97	1478.00#	Kim et al. (1994)
49.23	18.13	6.08	12.17	2.28	0.82	1.79	1197.00#	Kim et al. (1994)
57.75	23.00	4.10	2.23	0.94	1.02	1.93	1474.00	Kim et al. (1994)
32.58	27.49	21.23	4.11	1.85	0.25	1.11	1236.00	Kim et al. (1994)
50.00	27.00	3.50	9.20	1.50	1.70	0.80	1450.00	CEA Report (2012)
53.00	19.60	14.60	4.20	1.45	1.00	2.07	1200.00	CEA Report (2012)
58.50	33.70	1.60	1.49	1.00	1.48	1.51	1550.00	CEA Report (2012)
73.00	16.00	4.00	1.60	1.20	0.74	1.49	1450.00*	CEA Report (2012)
72.00	18.00	4.00	1.00	0.60	0.82	1.20	1450.00	CEA Report (2012)

* Test data; # Validation data

APPENDIX 5

Table 5.C: Experimental data consisting of the oxide composition (wt %) and the corresponding magnitudes of *Hemispherical Temperature* (HT)

SiO_2	Al_2O_3	Fe_2O_3	CaO	MgO	TiO_2	$Na_2O + K_2O$	RH	Ref
30.73	17.15	19.73	19.02	7.22	0.61	4.97	1070.00	CIMFR
52.29	16.45	12.06	10.56	3.61	0.85	3.90	1150.00*	CIMFR
39.95	13.40	23.51	10.99	5.88	0.71	5.08	1070.00$^{\#}$	CIMFR
44.13	19.24	10.95	16.51	3.97	0.78	6.04	1150.00	CIMFR
42.32	16.44	21.50	7.40	5.85	0.77	5.41	1100 .00*	CIMFR
50.01	23.48	11.81	4.47	5.49	0.67	3.68	1120.00	CIMFR
51.21	23.90	10.81	3.46	5.19	0.79	4.32	1150.00	CIMFR
48.43	12.71	17.54	3.44	15.51	0.34	1.65	1230.00 *	CIMFR
49.28	12.86	15.17	1.63	18.73	0.27	1.65	1280.00	CIMFR
56.19	20.70	10.23	3.55	2.56	0.97	5.63	1160.00 *	CIMFR
53.23	20.02	9.01	6.81	3.50	0.25	6.88	1160.00 *	CIMFR
53.03	13.56	12.96	12.29	4.88	0.61	2.21	1120.00	CIMFR
41.73	19.62	13.06	18.72	2.20	0.94	2.54	1170.00	CIMFR
49.58	26.53	5.65	8.75	4.57	0.75	3.65	1190.00$^{\#}$	Lolja et al. (2002)
44.57	6.46	8.26	27.09	7.06	0.46	5.86	1290.00	Lolja et al. (2002)
43.76	14.97	14.11	16.44	4.59	0.64	5.33	1110.00	Lolja et al. (2002)
50.41	16.66	8.84	12.95	4.79	0.61	5.51	1120.00 *	Lolja et al. (2002)
47.27	22.96	22.81	1.30	0.85	1.00	2.25	1360.00	Vorres (1979)
37.64	20.11	29.28	4.25	1.25	0.81	2.40	1280.00	Vorres (1979)
47.52	17.87	20.13	5.75	1.02	0.78	2.13	1193.00	Vorres (1979)
48.00	11.50	7.00	25.00	4.00	0.60	1.40	1171.00	Vorres (1979)
46.60	21.40	18.17	3.31	0.77	1.20	3.15	1230.55	Moreno (2008)
44.65	18.70	18.87	7.29	0.11	0.92	1.69	1182.22$^{\#}$	Moreno (2008)
38.10	18.60	30.57	2.67	0.62	0.86	3.72	1126.66 *	Moreno (2008)
52.63	28.08	6.41	2.15	0.75	1.53	3.39	1482.22$^{\#}$	Moreno (2008)
54.86	27.98	3.84	2.01	1.37	1.34	3.56	1482.22	Moreno (2008)
58.43	23.64	4.51	3.25	1.05	0.90	2.33	1426.66	Moreno (2008)
60.97	21.71	6.18	1.83	1.34	0.86	3.00	1410.00$^{\#}$	Moreno (2008)
61.79	24.76	4.76	0.52	0.38	1.16	3.05	1482.22	Moreno (2008)
49.20	19.10	20.40	5.18	0.82	0.89	2.58	1148.88	Moreno (2008)
41.20	15.60	22.00	8.08	0.70	0.75	2.55	1093.33	Moreno (2008)
45.60	17.00	4.18	5.50	2.68	1.26	1.02	1126.66	Moreno (2008)
46.20	18.10	4.90	13.50	4.52	1.20	6.56	1115.55	Moreno (2008)
30.58	14.60	4.07	23.63	3.08	1.06	4.31	1171.11	Moreno (2008)
33.03	14.79	5.05	13.23	2.82	1.05	6.69	1143.33	Moreno (2008)
34.21	17.05	17.59	13.00	3.96	0.69	1.20	1210.00 *	Moreno (2008)
42.20	32.00	18.30	1.80	1.00	2.10	0.30	1490.00	Bryant et al. (2000)
49.40	31.60	11.80	1.80	1.00	1.80	0.30	1590.00	Bryant et al. (2000)

60.80	22.30	5.60	3.20	1.20	0.80	3.10	1350.00	Bryant et al. (2000)
26.00	11.40	12.20	29.60	5.00	0.80	5.50	1200.00	Bryant et al. (2000)
41.10	12.50	16.30	10.30	8.36	0.64	1.63	1180.00	Akiyama et al. (2011)
62.90	22.90	4.58	1.08	0.79	1.37	1.84	1548.00	Akiyama et al. (2011)
49.24	23.84	11.07	4.81	1.73	1.56	1.43	1280.00	Steyn and Minnitt (2010)
32.80	12.40	24.70	16.50	12.30	0.26	0.35	1650.00	Bahrinet al. (2009)
79.70	18.49	0.39	0.57	0.07	0.24	0.53	1650.00	Bahrinet al. (2009)
77.60	19.60	1.52	0.76	0.09	0.25	0.11	1650.00	Bahrinet al. (2009)
71.40	15.38	11.30	0.57	0.17	0.58	0.59	1296.00 *	Bahrinet al. (2009)
42.90	16.50	1.30	22.00	16.50	0.25	0.44	1650.00 *	Bahrinet al. (2009)
29.10	6.75	37.70	15.10	10.10	0.20	0.19	1367.00#	Bahrinet al. (2009)
49.10	25.30	11.17	9.80	2.73	0.68	0.02	1650.00	Bahrinet al. (2009)
37.60	34.73	1.73	6.15	5.43	6.68	5.77	1466.00	Bahrinet al. (2009)
39.24	43.62	1.45	5.40	3.21	3.29	2.69	1352.00 *	Bahrinet al. (2009)
41.18	30.53	1.73	7.17	4.75	8.32	4.32	1384.00	Bahrinet al. (2009)
36.26	41.52	3.90	6.67	5.00	3.58	1.81	1224.00	Bahrinet al. (2009)
23.58	46.98	3.36	11.73	7.00	4.41	1.66	1650.00	Bahrinet al. (2009)
78.80	16.07	0.01	0.06	0.95	2.11	1.26	1078.00	Bahrinet al. (2009)
32.00	20.10	15.80	12.80	2.00	1.90	2.40	1265.00	Matuszewski (2012)
44.80	24.10	17.30	4.20	1.60	1.30	2.70	1271.00	Matuszewski (2012)
45.00	18.00	20.00	7.00	1.00	1.00	2.50	1285.00	Matuszewski (2012)
33.40	16.30	5.20	21.50	6.40	1.20	2.25	1204.44	Matuszewski (2012)
38.09	16.73	6.46	16.56	4.25	0.72	0.92	1243.33	Matuszewski (2012)
35.06	12.29	5.12	14.39	6.61	0.58	5.82	1162.77	Matuszewski (2012)
44.10	15.03	8.96	11.91	2.67	1.03	1.48	1231.11 *	Matuszewski (2012)
19.60	5.74	52.70	8.69	3.41	0.12	6.26	1460.00	Kim et al. (2010)
23.74	5.43	47.31	9.64	0.79	0.21	5.68	1485.00	Kim et al. (2010)
34.18	6.30	29.87	14.82	2.62	0.14	4.16	1390.00	Kim et al. (2010)
0.55	2.20	85.33	3.33	0.84	0.18	1.82	1500.00 *	Kim et al. (2010)
4.93	3.08	65.71	12.21	2.25	0.16	6.14	1470.00 *	Kim et al. (2010)
30.04	13.47	29.39	6.71	2.06	0.12	4.64	1480.00	Kim et al. (2010)
52.67	17.75	12.97	6.86	2.76	0.26	6.71	1406.00	Kim et al. (2010)
35.13	5.55	2.45	33.06	5.52	0.19	14.29	1325.00	Kim et al. (2010)
36.38	8.54	3.94	30.15	7.41	0.22	5.26	1290.00 *	Kim et al. (2010)
74.20	15.78	3.45	1.69	0.50	0.80	0.94	1476.00#	Kim et al. (1994)
66.90	20.52	4.56	1.49	0.51	0.92	0.97	1499.00	Kim et al. (1994)
49.23	18.13	6.08	12.17	2.28	0.82	1.79	1217.00#	Kim et al. (1994)
57.75	23.00	4.10	2.23	0.94	1.02	1.93	1498.00	Kim et al. (1994)
32.58	27.49	21.23	4.11	1.85	0.25	1.11	1257.00 *	Kim et al. (1994)
50.00	27.00	3.50	9.20	1.50	1.70	0.80	1480.00	CEA Report (2012)
53.00	19.60	14.60	4.20	1.45	1.00	2.07	1230.00	CEA Report (2012)
58.50	33.70	1.60	1.49	1.00	1.48	1.51	1570.00	CEA Report (2012)
73.00	16.00	4.00	1.60	1.20	0.74	1.49	1480.00	CEA Report (2012)
72.00	18.00	4.00	1.00	0.60	0.82	1.20	1470.00*	CEA Report (2012)

*Test data; # Validation data

APPENDIX 5

Table 5.D: Experimental data consisting of the oxide composition (wt %) and the corresponding magnitudes of *Fluid Temperature* (FT)

SiO_2	**Al_2O_3**	**Fe_2O_3**	**CaO**	**MgO**	**TiO_2**	**K_2O + Na_2O**	**FT**	**Reference§**
30.88	17.24	19.83	19.12	7.26	0.61	4.99	1090.00*	Lolja et al. (2002)
52.42	16.49	12.09	10.59	3.62	0.85	3.91	1200.00	Lolja et al. (2002)
40.11	13.46	23.61	11.04	5.90	0.71	5.10	1130.00	Lolja et al. (2002)
43.40	18.92	10.77	16.24	3.90	0.77	5.94	1230.00	Lolja et al. (2002)
42.40	16.47	21.54	7.41	5.86	0.77	5.42	1130.00	Lolja et al. (2002)
50.15	23.54	11.84	4.48	5.50	0.67	3.69	1210.00	Lolja et al. (2002)
51.32	23.95	10.83	3.47	5.20	0.79	4.33	1240.00	Lolja et al. (2002)
56.24	20.72	10.24	3.55	2.56	0.97	5.64	1230.00	Lolja et al. (2002)
53.19	13.60	13.00	12.33	4.90	0.61	2.22	1200.00	Lolja et al. (2002)
42.13	19.81	13.19	18.90	2.22	0.95	2.56	1260.00	Lolja et al. (2002)
49.81	26.66	5.68	8.79	4.59	0.75	3.67	1260.00*	Lolja et al. (2002)
43.83	14.99	14.13	16.46	4.60	0.64	5.34	1150.00*	Lolja et al. (2002)
50.51	16.69	8.86	12.98	4.80	0.61	5.52	1180.00*	Lolja et al. (2002)
54.68	28.81	3.95	2.72	6.26	0.80	2.77	1380.00#	Qiu et al. (1999)
39.13	26.33	11.06	13.83	6.59	0.70	2.37	1400.00*	Qiu et al. (1999)
51.36	33.73	8.63	1.40	0.64	1.55	2.69	1510.00	Hanxu et al. (2006)
48.02	23.32	23.17	1.32	0.86	1.02	2.29	1437.00	Vorres (1979)
49.92	18.77	21.14	6.04	1.07	0.82	2.24	1271.00*	Vorres (1979)
49.23	11.79	7.18	25.64	4.10	0.62	1.44	1232.00	Vorres (1979)
38.85	20.25	26.13	7.54	1.25	0.89	5.08	1025.55	Rees (1964)
35.85	20.59	25.33	8.52	3.25	0.84	5.63	1075.55	Rees (1964)
35.70	20.63	26.50	8.09	2.31	0.96	5.80	1036.66*	Rees (1964)
43.14	18.15	23.74	8.36	1.05	0.90	4.66	1034.44	Rees (1964)
46.62	19.03	23.19	4.25	1.02	0.99	4.89	1103.33#	Rees (1964)
44.19	17.55	28.29	3.02	1.03	1.01	4.90	1103.33 #	Rees (1964)
35.00	16.11	31.15	13.43	0.97	0.78	2.57	1015.00	Rees (1964)
39.30	14.46	34.98	6.85	0.93	0.80	2.67	1015.00#	Rees (1964)
44.53	17.27	23.64	7.91	1.01	1.05	4.59	1054.44	Rees (1964)
47.26	19.65	16.65	11.83	0.86	0.83	2.91	1163.33	Rees (1964)
47.76	20.07	12.98	14.33	0.94	0.91	3.01	1154.44*	Rees (1964)
40.30	18.21	21.08	16.14	0.68	0.86	2.72	1154.44*	Rees (1964)
35.28	20.40	30.31	9.43	0.76	0.97	2.84	1208.88	Rees (1964)
34.30	19.78	30.33	11.28	0.70	1.10	2.52	1206.66	Rees (1964)
52.64	23.95	13.67	4.38	0.87	1.07	3.42	1243.33	Rees (1964)
42.59	17.37	14.81	20.74	0.90	0.77	2.82	1232.22*	Rees (1964)
43.26	17.34	12.92	21.39	1.09	0.68	3.32	1156.66	Rees (1964)
54.70	21.67	13.66	4.50	1.04	1.00	3.42	1202.22	Rees (1964)
52.47	22.71	14.72	5.15	0.94	1.09	2.92	1177.22*	Rees (1964)

44.72	22.38	21.96	4.76	1.31	1.15	3.72	1248.33	Rees (1964)
55.78	28.16	7.21	1.44	1.28	1.36	4.77	1371.11	Rees (1964)
52.46	24.39	6.61	9.76	1.05	1.22	4.52	1257.22	Rees (1964)
46.98	23.01	12.50	11.32	1.87	0.63	3.68	1288.33	Rees (1964)
46.58	22.95	14.98	9.47	2.38	0.67	2.97	1288.33	Rees (1964)
50.73	22.73	13.35	6.97	1.84	0.68	3.71	1215.00	Rees (1964)
41.31	25.08	16.25	12.20	1.22	0.97	2.97	1201.111	Rees (1964)
49.26	22.62	19.21	3.50	0.81	1.27	3.33	1261.66	Moreno (2008)
48.41	20.28	20.46	7.90	0.12	1.00	1.83	1254.44	Moreno (2008)
40.05	19.55	32.13	2.81	0.65	0.90	3.91	1148.88*	Moreno (2008)
55.44	29.58	6.75	2.26	0.79	1.61	3.57	1482.22	Moreno (2008)
57.77	29.47	4.04	2.12	1.44	1.41	3.75	1482.22*	Moreno (2008)
62.09	25.12	4.79	3.45	1.12	0.96	2.48	1482.22	Moreno (2008)
52.24	23.49	9.42	6.85	1.98	0.92	5.10	1376.66#	Moreno (2008)
63.58	22.64	6.44	1.91	1.40	0.90	3.13	1476.66*	Moreno (2008)
64.08	25.68	4.94	0.54	0.39	1.20	3.16	1482.22*	Moreno (2008)
50.12	19.46	20.78	5.28	0.84	0.91	2.63	1193.33	Moreno (2008)
45.33	17.17	24.21	8.89	0.77	0.83	2.81	1115.55	Moreno (2008)
59.04	22.01	5.41	7.12	3.47	1.63	1.32	1148.88	Moreno (2008)
48.64	19.06	5.16	14.21	4.76	1.26	6.91	1143.33 *	Moreno (2008)
39.01	19.44	20.06	14.82	4.52	0.79	1.37	1270.00	Moreno (2008)
43.19	32.75	18.73	1.84	1.02	2.15	0.31	1540.00	Bryant et al. (2000)
50.56	32.34	12.08	1.84	1.02	1.84	0.31	1615.00	Bryant et al. (2000)
62.68	22.99	5.77	3.30	1.24	0.82	3.20	1410.00	Bryant et al. (2000)
62.61	26.53	4.42	0.40	0.80	1.21	4.02	1605.00	Bryant et al. (2000)
65.87	23.98	4.80	1.13	0.83	1.43	1.93	1570.00	Akiyama et al. (2011)
52.55	25.44	11.81	5.13	1.85	1.66	1.53	1320.00	Steyn and Minnitt (2010)
49.70	25.61	11.31	9.92	2.76	0.69	0.02	1650.00	Bahrinet al. (2009)
36.78	23.10	18.16	14.71	2.30	2.18	2.76	1319.00 *	Matuszewski (2012)
46.67	25.10	18.02	4.38	1.67	1.35	2.81	1333.00*	Matuszewski (2012)
44.57	23.78	24.27	3.41	0.75	1.10	2.12	1332.22	Matuszewski (2012)
47.62	19.05	21.16	7.41	1.06	1.06	2.65	1323.88	Matuszewski (2012)
45.49	19.98	7.72	19.78	5.08	0.86	1.10	1258.88	Matuszewski (2012)
51.77	17.64	10.52	13.98	3.13	1.21	1.74	1294.44	Matuszewski (2012)
58.44	28.86	5.82	0.51	0.20	2.49	3.67	1537.77	U.S.DOE (2001)
59.55	28.52	5.43	0.11	0.67	2.04	3.69	1463.88 #	U.S.DOE (2001)
56.54	28.48	8.97	0.53	0.14	1.98	3.35	1537.77	U.S.DOE (2001)
54.06	26.21	15.23	0.48	0.16	1.56	2.28	1445.00*	U.S.DOE (2001)
36.32	25.63	30.37	3.84	0.31	1.31	2.21	1240.556	U.S.DOE (2001)
64.04	11.75	13.04	4.42	1.92	1.84	2.99	1480.55	U.S.DOE (2001)
69.78	21.40	4.76	1.55	0.53	0.96	1.01	1516.00	Kim et al. (1994)
54.40	20.03	6.72	13.45	2.52	0.91	1.98	1243.00	Kim et al. (1994)
63.48	25.28	4.51	2.45	1.03	1.12	2.12	1523.00	Kim et al. (1994)
55.25	20.43	15.22	4.38	1.51	1.04	2.16	1260.00#	CEA Report (2012)
58.92	33.94	1.61	1.50	1.01	1.49	1.52	1620.00#	CEA Report (2012)

74.47	16.32	4.08	1.63	1.22	0.75	1.52	1540.00	CEA Report (2012)
73.76	18.44	4.10	1.02	0.61	0.84	1.23	1500.00	CEA Report (2012)
58.72	23.66	8.93	2.94	1.28	1.14	3.28	1410.00	CIMFR
59.05	23.82	12.88	0.99	0.42	1.94	0.88	1430.00	CIMFR
59.63	19.07	9.10	7.27	1.96	1.83	1.00	1340.00 #	CIMFR
64.44	22.99	5.67	2.37	0.79	1.30	2.39	1430.00	CIMFR
60.63	21.90	8.62	3.52	1.13	1.65	2.44	1360.00	CIMFR
59.80	14.67	11.74	6.79	1.54	1.85	3.34	1310.00	CIMFR
59.06	25.01	7.06	4.78	1.41	2.00	0.51	1380.00	CIMFR
61.36	21.74	6.94	2.96	0.96	1.96	4.00	1410.00	CIMFR
64.18	17.36	10.19	1.70	0.44	1.96	4.11	1400.00	CIMFR

*: Test data; #: Validation data; § CIMFR: *Central Institute of Mining and Fuel Research*, Dhanbad, India, from where data were sourced.

APPENDIX 5

Table 5.E: Mean and Standard deviation of oxide weight percentages (wt %) used in the normalization of the oxides and AFT data

AFT model		SiO_2 wt (%) (x_1)	Al_2O_3 wt (%) (x_2)	Fe_2O_3 wt (%) (x_3)	CaO wt (%) (x_4)	MgO wt (%) (x_5)	TiO_2 wt (%) (x_6)	K_2O+Na_2O wt (%) (x_7)	y (°C)#
IDT	Mean	55.75	23.63	10.51	4.8	1.37	1.38	2.56	1189.36
	Std. dev.	9.53	4.61	7.32	5.50	1.69	0.48	1.47	113.36
ST	Mean	45.85	21.60	12.93	7.14	2.04	1.25	2.61	1264.87
	Std. dev.	12.01	7.14	8.99	5.80	2.57	1.13	1.48	179.17
HT	Mean	45.70	19.05	13.80	8.95	3.53	1.08	3.07	1314.96
	Std. dev.	14.69	8.39	14.27	7.73	3.61	1.26	2.10	169.95
FT	Mean	50.77	21.79	14.06	7.21	1.93	1.13	3.09	1300.42
	Std. dev.	9.30	4.69	8.03	5.75	1.70	0.44	1.41	158.02

y denotes the specific ash fusion temperature

The mean values ($\bar{x}_r$) and standard deviations (σ_r) of AFTs are calculated as per the equations (5.30) and (5.31) respectively, given below.

$$\bar{x}_r = \frac{\sum_{i=1}^{N_p} x_r^i}{N_p};\ i = 1,2,\dots,N_p\ ;\ r = 1,2,\dots.,R \qquad (5.30)$$

$$\sigma_r = \sqrt{\frac{\sum_{i=1}^{N_p}(x_r^i - \bar{x}_r)}{N_p}};\ i = 1,2,\dots,N_p\ ;\ r = 1,2,\dots.,R \qquad (5.31)$$

APPENDIX 5

SVR Models 5.F: SVR models pertaining to the *Ash Fusion Temperatures* (AFTs) predictions

5. F.1: SVR model pertaining to the *Initial Deformation Temperature* (IDT) prediction

$\widehat{IDT} =$ -0.4782293782502961 * (pow((exp(-2.0 * (0.6249796012154752 - pc1) * (0.6249796012154752 - pc1)) + exp(-2.0 * (-3.0086738013710557 - pc2) * (-3.0086738013710557 - pc2)) + exp(-2.0 * (-0.5132805311768492 - pc3) * (-0.5132805311768492 - pc3)) + exp(-2.0 * (-0.3679991158006609 - pc4) * (-0.3679991158006609 - pc4)) + exp(-2.0 * (-0.8503918051375732 - pc5) * (-0.8503918051375732- pc5))),1.0))-1.8906674653578492*(pow((exp(-2.0*(-0.7234797465934767 - pc1) * (-0.7234797465934767 - pc1)) + exp(-2.0 * (0.7093038060261255 - pc2) * (0.7093038060261255 - pc2)) + exp(-2.0 * (0.43893205970101024 - pc3) * (0.43893205970101024 - pc3)) + exp(-2.0 * (0.7718088160781258 - pc4) * (0.7718088160781258 - pc4)) + exp(-2.0 * (-1.0583139659230756 - pc5) * (-1.0583139659230756 - pc5))), 1.0))- 0.7956968092437257 * (pow((exp(-2.0 * (0.368178417920692 - pc1) * (0.368178417920692 - pc1)) + exp(-2.0 * (-1.5974136903002447 - pc2) * (-1.5974136903002447 - pc2)) + exp(-2.0 * (0.44933429361869115 - pc3) * (0.44933429361869115 - pc3)) + exp(-2.0 * (0.2923618758436084 - pc4) * (0.2923618758436084 - pc4)) + exp(-2.0 * (-1.2461731099601343 - pc5) * (-1.2461731099601343 - pc5))), 1.0)) - 0.38468733516512377 * (pow((exp(-2.0 * (1.4665349179369038 - pc1) * (1.4665349179369038 - pc1)) + exp(-2.0 * (1.358911950742573 - pc2) * (1.358911950742573 - pc2)) + exp(-2.0 * (-0.16054090747582972 - pc3) * (-0.16054090747582972 - pc3)) + exp(-2.0 * (2.030709498061156 - pc4) * (2.030709498061156 - pc4)) + exp(-2.0 * (0.4737104336365257 - pc5) * (0.4737104336365257 - pc5))), 1.0))- 1.1816724156016745 * (pow((exp(-2.0 * (-0.8222581473749505 - pc1) * (-0.8222581473749505 - pc1)) + exp(-2.0 * (-0.6416554927981424 - pc2) * (-0.6416554927981424 - pc2)) + exp(-2.0 * (-0.6909565112975972 - pc3) * (-0.6909565112975972 - pc3)) + exp(-2.0 * (0.2868559327333015 - pc4) * (0.2868559327333015 - pc4)) + exp(-2.0 * (0.4093456105054995 - pc5) * (0.4093456105054995 - pc5))), 1.0))+ 0.44991087865600987 * (pow((exp(-2.0 * (-0.3197112012561222 - pc1) * (-0.3197112012561222 - pc1)) + exp(-2.0 * (0.829359422770002 - pc2) * (0.829359422770002 - pc2)) + exp(-2.0 * (0.38417188865775154 - pc3) * (0.38417188865775154 - pc3)) + exp(-2.0 * (0.20095450585105593 - pc4) * (0.20095450585105593 - pc4)) + exp(-2.0 * (-1.4052705953846751 - pc5) * (-1.4052705953846751 - pc5))), 1.0)) + 0.31956097369833397 * (pow((exp(-2.0 * (-0.45870579925180843 - pc1) * (-0.45870579925180843 - pc1)) + exp(-2.0 * (0.13459370317838007 - pc2) * (0.13459370317838007 - pc2)) + exp(-2.0 * (1.6022003154273965 - pc3) * (1.6022003154273965 - pc3)) + exp(-2.0 * (-0.3583951355985099 - pc4) * (-0.3583951355985099 - pc4)) + exp(-2.0 * (1.0121647449117983 - pc5) * (1.0121647449117983 - pc5))), 1.0))+ 1.048444223568384 * (pow((exp(-2.0 * (1.3548772675448715 - pc1) * (1.3548772675448715 - pc1)) + exp(-2.0 * (-2.4320911946977897 - pc2) * (-2.4320911946977897 - pc2)) + exp(-2.0 * (-0.05460183514580577 - pc3) * (-0.05460183514580577 - pc3)) + exp(-2.0 * (0.26253791475361904 - pc4) * (0.26253791475361904 - pc4)) + exp(-2.0 * (-0.3226465321951865 - pc5) * (-0.3226465321951865 - pc5))), 1.0))- 0.12758944921830406 * (pow((exp(-2.0 * (0.9797138981230443 - pc1) * (0.9797138981230443 - pc1)) + exp(-2.0 * (1.176467871237364 - pc2) * (1.176467871237364 - pc2)) + exp(-2.0 * (-0.5237066713755737 - pc3) * (-0.5237066713755737 - pc3)) + exp(-2.0 * (-0.7024915945471241 - pc4) * (-0.7024915945471241 - pc4)) + exp(-2.0 * (-1.7602935735835943 - pc5) * (-1.7602935735835943 - pc5))), 1.0))

+ 0.1495357423225121 * (pow((exp(-2.0 * (1.9195869069872125 - pc1) * (1.9195869069872125 - pc1)) + exp(-2.0 * (2.0705611663740746 - pc2) * (2.0705611663740746 - pc2)) + exp(-2.0 * (-3.7519043717396667 - pc3) * (-3.7519043717396667 - pc3)) + exp(-2.0 * (0.055852417854658463 - pc4) * (0.055852417854658463 - pc4)) + exp(-2.0 * (2.4136677601732526 - pc5) * (2.4136677601732526 - pc5))), 1.0))- 1.2172184145617744 * (pow((exp(-2.0 * (1.0174236501575316 - pc1) * (1.0174236501575316 - pc1)) + exp(-2.0 * (-1.12074169563642 - pc2) * (-1.12074169563642 - pc2)) + exp(-2.0 * (1.6768154565281947 - pc3) * (1.6768154565281947 - pc3)) + exp(-2.0 * (-0.9011207121827194 - pc4) * (-0.9011207121827194 - pc4)) + exp(-2.0 * (-0.14124257287243114 - pc5) * (-0.14124257287243114 - pc5))), 1.0))+ 0.13368510889006355 * (pow((exp(-2.0 * (-0.9537135665142016 - pc1) * (-0.9537135665142016 - pc1)) + exp(-2.0 * (-0.5413637842912423 - pc2) * (-0.5413637842912423 - pc2)) + exp(-2.0 * (-2.7460702573198192 - pc3) * (-2.7460702573198192 - pc3)) + exp(-2.0 * (-1.9232156270420993 - pc4) * (-1.9232156270420993 - pc4)) + exp(-2.0 * (-0.7161064514941698 - pc5) * (-0.7161064514941698 - pc5))), 1.0))- 0.02490307243702141 * (pow((exp(-2.0 * (1.4636427746711345 - pc1) * (1.4636427746711345 - pc1)) + exp(-2.0 * (2.0630112395589144 - pc2) * (2.0630112395589144 - pc2)) + exp(-2.0 * (0.1362221264255667 - pc3) * (0.1362221264255667 - pc3)) + exp(-2.0 * (-2.662740833671813 - pc4) * (-2.662740833671813 - pc4)) + exp(-2.0 * (1.5442825227539574 - pc5) * (1.5442825227539574 - pc5))), 1.0))+ 0.931309548534361 * (pow((exp(-2.0 * (-1.319266030336142 - pc1) * (-1.319266030336142 - pc1)) + exp(-2.0 * (-0.5882451493447722 - pc2) * (-0.5882451493447722 - pc2)) + exp(-2.0 * (-1.0788007284083023 - pc3) * (-1.0788007284083023 - pc3)) + exp(-2.0 * (-0.8825161915615355 - pc4) * (-0.8825161915615355 - pc4)) + exp(-2.0 * (1.116805257102981 - pc5) * (1.116805257102981 - pc5))), 1.0))- 0.5012012781409488 * (pow((exp(-2.0 * (1.437625240530321 - pc1) * (1.437625240530321 - pc1)) + exp(-2.0 * (-1.5312305274649929 - pc2) * (-1.5312305274649929 - pc2)) + exp(-2.0 * (1.0118718163099176 - pc3) * (1.0118718163099176 - pc3)) + exp(-2.0 * (-1.2305525094996559 - pc4) * (-1.2305525094996559 - pc4)) + exp(-2.0 * (-0.46319684888177665 - pc5) * (-0.46319684888177665 - pc5))), 1.0)) - 1.2899288326801295 * (pow((exp(-2.0 * (0.9183753272479042 - pc1) * (0.9183753272479042 - pc1)) + exp(-2.0 * (-0.061871576361406104 - pc2) * (-0.061871576361406104 - pc2)) + exp(-2.0 * (-0.1304078247905717 - pc3) * (-0.1304078247905717 - pc3)) + exp(-2.0 * (0.5125232826877286 - pc4) * (0.5125232826877286 - pc4)) + exp(-2.0 * (0.10422468761514365 - pc5) * (0.10422468761514365 - pc5))), 1.0))+ 0.9502838799953855 * (pow((exp(-2.0 * (-0.3353053913603851 - pc1) * (-0.3353053913603851 - pc1)) + exp(-2.0 * (1.5715466069091688 - pc2) * (1.5715466069091688 - pc2)) + exp(-2.0 * (1.9132352482794084 - pc3) * (1.9132352482794084 - pc3)) + exp(-2.0 * (2.4996072093387713 - pc4) * (2.4996072093387713 - pc4)) + exp(-2.0 * (-0.3334810698177196 - pc5) * (-0.3334810698177196 - pc5))), 1.0))+ 1.2733612545031952 * (pow((exp(-2.0 * (1.3288412107554888 - pc1) * (1.3288412107554888 - pc1)) + exp(-2.0 * (0.10828833966609695 - pc2) * (0.10828833966609695 - pc2)) + exp(-2.0 * (-1.3107280501935084 - pc3) * (-1.3107280501935084 - pc3)) + exp(-2.0 * (0.6977134235041241 - pc4) * (0.6977134235041241 - pc4)) + exp(-2.0 * (0.5294942742665419 - pc5) * (0.5294942742665419 - pc5))), 1.0)) + 1.0397074131047952 * (pow((exp(-2.0 * (2.002804947328749 - pc1) * (2.002804947328749 - pc1)) + exp(-2.0 * (0.46272190213547704 - pc2) * (0.46272190213547704 - pc2)) + exp(-2.0 * (0.4609485832373895 - pc3) * (0.4609485832373895 - pc3)) + exp(-2.0 * (-1.1032587697221352 - pc4) * (-1.1032587697221352 - pc4)) + exp(-2.0 * (0.07549424982275604 - pc5) * (0.07549424982275604 - pc5))), 1.0))+ 0.2196471407551924 * (pow((exp(-2.0 * (0.8376633953274 - pc1) * (0.8376633953274 - pc1)) + exp(-2.0 * (0.7214110993768963 - pc2) * (0.7214110993768963 - pc2)) + exp(-2.0 * (-0.9768446387350148 - pc3) * (-0.9768446387350148 - pc3)) + exp(-2.0 * (1.401881678508821 - pc4) * (1.401881678508821 - pc4)) + exp(-2.0 * (1.348507523873152 - pc5) * (1.348507523873152 - pc5))), 1.0))+ 1.8695652173913044 * (pow((exp(-2.0 * (0.7495555314952123 - pc1) * (0.7495555314952123 - pc1)) + exp(-2.0 * (-1.2659851491839984 - pc2) * (-1.2659851491839984 - pc2)) + exp(-2.0 * (0.46180718067303445 - pc3) * (0.46180718067303445 - pc3)) + exp(-2.0 * (-

0.7375057122251859 - pc4) * (-0.7375057122251859 - pc4)) + exp(-2.0 * (-0.41023638043474425 - pc5) * (-0.41023638043474425 - pc5))), 1.0))- 0.16824273284751876 * (pow((exp(-2.0 * (1.3295900248912609 - pc1) * (1.3295900248912609 - pc1)) + exp(-2.0 * (-1.1590918203217693 - pc2) * (-1.1590918203217693 - pc2)) + exp(-2.0 * (1.2414623454275604 - pc3) * (1.2414623454275604 - pc3)) + exp(-2.0 * (-0.5321160680732857 - pc4) * (-0.5321160680732857 - pc4)) + exp(-2.0 * (0.033927991886748284 - pc5) * (0.033927991886748284 - pc5))), 1.0))+ 1.1911127951469993 * (pow((exp(-2.0 * (-0.1428138373352612 - pc1) * (-0.1428138373352612 - pc1)) + exp(-2.0 * (0.9259027831650705 - pc2) * (0.9259027831650705 - pc2)) + exp(-2.0 * (1.142507817689114 - pc3) * (1.142507817689114 - pc3)) + exp(-2.0 * (0.13428815974921687 - pc4) * (0.13428815974921687 - pc4)) + exp(-2.0 * (-1.1515570241742061 - pc5) * (-1.1515570241742061 - pc5))), 1.0))- 1.2682966560221163 * (pow((exp(-2.0 * (-0.6272713337685498 - pc1) * (-0.6272713337685498 - pc1)) + exp(-2.0 * (-0.3547334194756257 - pc2) * (-0.3547334194756257 - pc2)) + exp(-2.0 * (-0.8360148057069603 - pc3) * (-0.8360148057069603 - pc3)) + exp(-2.0 * (-0.6580212637886858 - pc4) * (-0.6580212637886858 - pc4)) + exp(-2.0 * (0.8169045258260675 - pc5) * (0.8169045258260675 - pc5))), 1.0))+ 0.9384584963253777 * (pow((exp(-2.0 * (1.8099909444394253 - pc1) * (1.8099909444394253 - pc1)) + exp(-2.0 * (1.2349589222195292 - pc2) * (1.2349589222195292 - pc2)) + exp(-2.0 * (0.013061097888562878 - pc3) * (0.013061097888562878 - pc3)) + exp(-2.0 * (-1.5333613398292425 - pc4) * (-1.5333613398292425 - pc4)) + exp(-2.0 * (0.25884681251589364 - pc5) * (0.25884681251589364 - pc5))), 1.0)) + 0.9981423167729659 * (pow((exp(-2.0 * (-0.8519312281819134 - pc1) * (-0.8519312281819134 - pc1)) + exp(-2.0 * (-0.9426573887483322 - pc2) * (-0.9426573887483322 - pc2)) + exp(-2.0 * (-0.10821766918659066 - pc3) * (-0.10821766918659066 - pc3)) + exp(-2.0 * (0.1990604195362753 - pc4) * (0.1990604195362753 - pc4)) + exp(-2.0 * (1.4382693931806791 - pc5) * (1.4382693931806791 - pc5))), 1.0))- 0.3385623479814199 * (pow((exp(-2.0 * (-0.3911907986791449 - pc1) * (-0.3911907986791449 - pc1)) + exp(-2.0 * (0.41142853774388205 - pc2) * (0.41142853774388205 - pc2)) + exp(-2.0 * (0.8801720832721799 - pc3) * (0.8801720832721799 - pc3)) + exp(-2.0 * (-1.2574169422324988 - pc4) * (-1.2574169422324988 - pc4)) + exp(-2.0 * (1.9596848106910578 - pc5) * (1.9596848106910578 - pc5))), 1.0))+ 1.1781984966148809 * (pow((exp(-2.0 * (0.4741544400940217 - pc1) * (0.4741544400940217 - pc1)) + exp(-2.0 * (-1.8750004470821673 - pc2) * (-1.8750004470821673 - pc2)) + exp(-2.0 * (-0.007406412163360328 - pc3) * (-0.007406412163360328 - pc3)) + exp(-2.0 * (0.24933637213053095 - pc4) * (0.24933637213053095 - pc4)) + exp(-2.0 * (-1.007018249542925 - pc5) * (-1.007018249542925 - pc5))), 1.0))- 1.0230364248903814 * (pow((exp(-2.0 * (-0.47866574247905735 - pc1) * (-0.47866574247905735 - pc1)) + exp(-2.0 * (0.7074148658177847 - pc2) * (0.7074148658177847 - pc2)) + exp(-2.0 * (1.7689850046381295 - pc3) * (1.7689850046381295 - pc3)) + exp(-2.0 * (-2.001576930862287 - pc4) * (-2.001576930862287 - pc4)) + exp(-2.0 * (0.647905103890755 - pc5) * (0.647905103890755 - pc5))), 1.0))- 1.4148192085561346 * (pow((exp(-2.0 * (1.415082440194339 - pc1) * (1.415082440194339 - pc1)) + exp(-2.0 * (-0.953941381903367 - pc2) * (-0.953941381903367 - pc2)) + exp(-2.0 * (0.7469862815304837 - pc3) * (0.7469862815304837 - pc3)) + exp(-2.0 * (-1.492787362270537 - pc4) * (-1.492787362270537 - pc4)) + exp(-2.0 * (-0.02823923324093792 - pc5) * (-0.02823923324093792 - pc5))), 1.0))- 1.9402208752367123 * (pow((exp(-2.0 * (1.7977503564553874 - pc1) * (1.7977503564553874 - pc1)) + exp(-2.0 * (-2.200262135950341 - pc2) * (-2.200262135950341 - pc2)) + exp(-2.0 * (0.09856976181816329 - pc3) * (0.09856976181816329 - pc3)) + exp(-2.0 * (1.2787349267348331 - pc4) * (1.2787349267348331 - pc4)) + exp(-2.0 * (0.24534436619214062 - pc5) * (0.24534436619214062 - pc5))), 1.0)) + 0.012934500674523704 * (pow((exp(-2.0 * (0.7859877144943733 - pc1) * (0.7859877144943733 - pc1)) + exp(-2.0 * (-1.2099767182344687 - pc2) * (-1.2099767182344687 - pc2)) + exp(-2.0 * (-2.229307909041825 - pc3) * (-2.229307909041825 - pc3)) + exp(-2.0 * (-1.1302414729505748 - pc4) * (-1.1302414729505748 - pc4)) + exp(-2.0 * (2.8478094247680334 - pc5) * (2.8478094247680334 - pc5))), 1.0))+ 1.824315016860721 * (pow((exp(-2.0 * (-

0.69991913550904 - pc1) * (-0.69991913550904 - pc1)) + exp(-2.0 * (-0.5770183224299656 - pc2) * (-0.5770183224299656 - pc2)) + exp(-2.0 * (-1.0869804191904973 - pc3) * (-1.0869804191904973 - pc3)) + exp(-2.0 * (-1.8782077250767808 - pc4) * (-1.8782077250767808 - pc4)) + exp(-2.0 * (-1.720137523421571 - pc5) * (-1.720137523421571 - pc5))), 1.0))+ 0.44145082575680494 * (pow((exp(-2.0 * (1.3398837821684602 - pc1) * (1.3398837821684602 - pc1)) + exp(-2.0 * (1.4188426611979807 - pc2) * (1.4188426611979807 - pc2)) + exp(-2.0 * (-2.7013820495623873 - pc3) * (-2.7013820495623873 - pc3)) + exp(-2.0 * (1.64111939321946 - pc4) * (1.64111939321946 - pc4)) + exp(-2.0 * (0.34273487022380905 - pc5) * (0.34273487022380905 - pc5))), 1.0)) - 1.2114773096141542 * (pow((exp(-2.0 * (-0.9998980459063925 - pc1) * (-0.9998980459063925 - pc1)) + exp(-2.0 * (0.3023648137017133 - pc2) * (0.3023648137017133 - pc2)) + exp(-2.0 * (-0.16680743143778567 - pc3) * (-0.16680743143778567 - pc3)) + exp(-2.0 * (0.005297260419573907 - pc4) * (0.005297260419573907 - pc4)) + exp(-2.0 * (-0.9215340255652208 - pc5) * (-0.9215340255652208 - pc5))), 1.0))- 0.25511181002762634 * (pow((exp(-2.0 * (-0.7604036884903581 - pc1) * (-0.7604036884903581 - pc1)) + exp(-2.0 * (-0.1903571737754973 - pc2) * (-0.1903571737754973 - pc2)) + exp(-2.0 * (-2.1439065135842768 - pc3) * (-2.1439065135842768 - pc3)) + exp(-2.0 * (-1.1804706485341157 - pc4) * (-1.1804706485341157 - pc4)) + exp(-2.0 * (-1.4874031733009239 - pc5) * (-1.4874031733009239 - pc5))), 1.0))+ 0.38280407428067903 * (pow((exp(-2.0 * (1.3819862710742987 - pc1) * (1.3819862710742987 - pc1)) + exp(-2.0 * (-2.510366370106826 - pc2) * (-2.510366370106826 - pc2)) + exp(-2.0 * (-0.36949558841945435 - pc3) * (-0.36949558841945435 - pc3)) + exp(-2.0 * (0.48374966658366714 - pc4) * (0.48374966658366714 - pc4)) + exp(-2.0 * (0.24331201604320066 - pc5) * (0.24331201604320066 - pc5))), 1.0))+ 0.059137137054607554 * (pow((exp(-2.0 * (0.3005481786566235 - pc1) * (0.3005481786566235 - pc1)) + exp(-2.0 * (-0.24614645462999427 - pc2) * (-0.24614645462999427 - pc2)) + exp(-2.0 * (0.4802887074845427 - pc3) * (0.4802887074845427 - pc3)) + exp(-2.0 * (0.8947823881606074 - pc4) * (0.8947823881606074 - pc4)) + exp(-2.0 * (-0.31164642051828945 - pc5) * (-0.31164642051828945 - pc5))), 1.0))- 0.6478293120307111 * (pow((exp(-2.0 * (0.7409007756356958 - pc1) * (0.7409007756356958 - pc1)) + exp(-2.0 * (1.505210233104074 - pc2) * (1.505210233104074 - pc2)) + exp(-2.0 * (-1.211292396002696 - pc3) * (-1.211292396002696 - pc3)) + exp(-2.0 * (-1.0840280013262154 - pc4) * (-1.0840280013262154 - pc4)) + exp(-2.0 * (-1.8315956999409644 - pc5) * (-1.8315956999409644 - pc5))), 1.0))+ 1.8695652173913044 * (pow((exp(-2.0 * (0.7989792217569944 - pc1) * (0.7989792217569944 - pc1)) + exp(-2.0 * (-1.0687002253363964 - pc2) * (-1.0687002253363964 - pc2)) + exp(-2.0 * (0.4690235536659604 - pc3) * (0.4690235536659604 - pc3)) + exp(-2.0 * (0.6834366460319818 - pc4) * (0.6834366460319818 - pc4)) + exp(-2.0 * (0.14744730514254903 - pc5) * (0.14744730514254903 - pc5))), 1.0))+ 1.8695652173913044 * (pow((exp(-2.0 * (-0.4853599114219964 - pc1) * (-0.4853599114219964 - pc1)) + exp(-2.0 * (0.6320243283706766 - pc2) * (0.6320243283706766 - pc2)) + exp(-2.0 * (-0.12882083356623392 - pc3) * (-0.12882083356623392 - pc3)) + exp(-2.0 * (-1.8329611456616952 - pc4) * (-1.8329611456616952 - pc4)) + exp(-2.0 * (-1.0765715985839077 - pc5) * (-1.0765715985839077 - pc5))), 1.0)) + 1.8695652173913044 * (pow((exp(-2.0 * (-0.6096327194362757 - pc1) * (-0.6096327194362757 - pc1)) + exp(-2.0 * (0.5885698632695061 - pc2) * (0.5885698632695061 - pc2)) + exp(-2.0 * (0.9851347529377144 - pc3) * (0.9851347529377144 - pc3)) + exp(-2.0 * (-0.5284787934883101 - pc4) * (-0.5284787934883101 - pc4)) + exp(-2.0 * (-1.0224902800157527 - pc5) * (-1.0224902800157527 - pc5))), 1.0))- 2.15 * (pow((exp(-2.0 * (-0.09829245276931184 - pc1) * (-0.09829245276931184 - pc1)) + exp(-2.0 * (1.0747602929685736 - pc2) * (1.0747602929685736 - pc2)) + exp(-2.0 * (-1.2218657144756986 - pc3) * (-1.2218657144756986 - pc3)) + exp(-2.0 * (0.705595914460183 - pc4) * (0.705595914460183 - pc4)) + exp(-2.0 * (1.2093581134205564 - pc5) * (1.2093581134205564 - pc5))), 1.0))- 2.15 * (pow((exp(-2.0 * (-0.4929541598578989 - pc1) * (-0.4929541598578989 - pc1)) + exp(-2.0 * (0.6383911238387872 - pc2) * (0.6383911238387872 - pc2)) + exp(-2.0 * (0.7164101987899022 - pc3) * (0.7164101987899022 - pc3)) + exp(-2.0 *

(0.3217408004550902 - pc4) * (0.3217408004550902 - pc4)) + exp(-2.0 * (-0.1567076957285209 - pc5) * (-0.1567076957285209 - pc5))), 1.0)) + 1.8695652173913044 * (pow((exp(-2.0 * (-0.7432387953763673 - pc1) * (-0.7432387953763673 - pc1)) + exp(-2.0 * (0.6236850156066523 - pc2) * (0.6236850156066523 - pc2)) + exp(-2.0 * (0.46751846385646484 - pc3) * (0.46751846385646484 - pc3)) + exp(-2.0 * (0.3945136928018495 - pc4) * (0.3945136928018495 - pc4)) + exp(-2.0 * (-0.08066064018756429 - pc5) * (-0.08066064018756429 - pc5))), 1.0))- 2.15 * (pow((exp(-2.0 * (0.9748641475924591 - pc1) * (0.9748641475924591 - pc1)) + exp(-2.0 * (0.22643208033140397 - pc2) * (0.22643208033140397 - pc2)) + exp(-2.0 * (-0.1115560115016746 - pc3) * (-0.1115560115016746 - pc3)) + exp(-2.0 * (-0.2735892846738488 - pc4) * (-0.2735892846738488 - pc4)) + exp(-2.0 * (-1.3685510080343917 - pc5) * (-1.3685510080343917 - pc5))), 1.0))- 2.15 * (pow((exp(-2.0 * (1.8198794218009606 - pc1) * (1.8198794218009606 - pc1)) + exp(-2.0 * (-0.9386633003465147 - pc2) * (-0.9386633003465147 - pc2)) + exp(-2.0 * (0.4101123387086363 - pc3) * (0.4101123387086363 - pc3)) + exp(-2.0 * (-1.8016835798373312 - pc4) * (-1.8016835798373312 - pc4)) + exp(-2.0 * (-0.5636798038906089 - pc5) * (-0.5636798038906089 - pc5))), 1.0)) + 1.8695652173913044 * (pow((exp(-2.0 * (1.4895434319611935 - pc1) * (1.4895434319611935 - pc1)) + exp(-2.0 * (-0.1695193290126769 - pc2) * (-0.1695193290126769 - pc2)) + exp(-2.0 * (-0.5191352539611838 - pc3) * (-0.5191352539611838 - pc3)) + exp(-2.0 * (1.159696341085127 - pc4) * (1.159696341085127 - pc4)) + exp(-2.0 * (0.6402377758434645 - pc5) * (0.6402377758434645 - pc5))), 1.0))+ 0.08469058526770479

... (5.32)

where $\widehat{IDT}$ refer to the normalized value of *IDT*; the magnitudes of *PCs* in equation (5.32) are calculated using the equations (5.8) to (5.12).

5. F.2: SVR model pertaining to the *Softening Temperature* (ST) prediction

$\widehat{ST} =$ 10.0 * (exp(-0.011 * ((1.23021476198636 - PC1) * (1.23021476198636 - PC1) + (1.14201809221745 - PC2) * (1.14201809221745 - PC2) + (1.16809154021076 - PC3) * (1.16809154021076 - PC3) + (-0.250442970770229 - PC4) * (-0.250442970770229 - PC4)))) + 7.721529939005738 * (exp(-0.011 * ((-1.69224952475161 - PC1) * (-1.69224952475161 - PC1) + (1.13675031961557 - PC2) * (1.13675031961557 - PC2) + (0.136547150448529 - PC3) * (0.136547150448529 - PC3) + (-0.72223447889088 - PC4) * (-0.72223447889088 - PC4)))) + 10.0 * (exp(-0.011 * ((-1.74070525109069 - PC1) * (-1.74070525109069 - PC1) + (0.455502395428974 - PC2) * (0.455502395428974 - PC2) + (-0.463639699591029 - PC3) * (-0.463639699591029 - PC3) + (-0.350429666149045 - PC4) * (-0.350429666149045 - PC4)))) + 9.999999999999998 * (exp(-0.011 * ((-1.67548528377675 - PC1) * (-1.67548528377675 - PC1) + (0.113370635568498 - PC2) * (0.113370635568498 - PC2) + (-0.689868561379388 - PC3) * (-0.689868561379388 - PC3) + (0.240550296040923 - PC4) * (0.240550296040923 - PC4)))) - 10.0 * (exp(-0.011 * ((1.21887582269466 - PC1) * (1.21887582269466 - PC1) + (-0.492259467454954 - PC2) * (-0.492259467454954 - PC2) + (-1.59390236364332 - PC3) * (-1.59390236364332 - PC3) + (0.170982974095913 - PC4) * (0.170982974095913 - PC4)))) - 10.0 * (exp(-0.011 * ((1.96013932642214 - PC1) * (1.96013932642214 - PC1) + (-0.931179306555024 - PC2) * (-0.931179306555024 - PC2) + (-0.667452959725247 - PC3) * (-0.667452959725247 - PC3) + (-0.873917204499567 - PC4) * (-0.873917204499567 - PC4)))) - 9.00277608956272 * (exp(-0.011 * ((0.323493327036093 - PC1) * (0.323493327036093 - PC1) + (-0.551142796816673 - PC2) * (-0.551142796816673 - PC2) + (-1.38054046113041 - PC3) * (-1.38054046113041 - PC3) + (-1.7103271646003 - PC4) * (-1.7103271646003 - PC4)))) - 10.0 * (exp(-0.011 * ((1.23369629098308 - PC1) * (1.23369629098308 - PC1) + (-0.213399006540639 - PC2) * (-0.213399006540639 - PC2) + (-1.34034515443713 - PC3) * (-1.34034515443713 - PC3) + (0.518267358452195 - PC4) * (0.518267358452195 - PC4)))) - 4.452221426081214 * (exp(-0.011 * ((0.711585313090217 - PC1) * (0.711585313090217 - PC1) + (-0.560738719942625 - PC2) * (-0.560738719942625 - PC2) + (-0.0867109272618593 - PC3) * (-0.0867109272618593 - PC3) + (0.114912512084976 - PC4) * (0.114912512084976 - PC4)))) - 10.0 * (exp(-0.011 * ((0.77678651548445 - PC1) * (0.77678651548445 - PC1) + (-0.31502798512925 - PC2) * (-0.31502798512925 - PC2) + (0.143763468984693 - PC3) * (0.143763468984693 - PC3) + (0.356740916881283 - PC4) * (0.356740916881283 - PC4)))) + 10.0 * (exp(-0.011 * ((1.34736274642148 - PC1) * (1.34736274642148 - PC1) + (-0.699814658598038 - PC2) * (-0.699814658598038 - PC2) + (-1.08392732994988 - PC3) * (-1.08392732994988 - PC3) + (-1.02181656478666 - PC4) * (-1.02181656478666 - PC4)))) + 8.966885984435134 * (exp(-0.011 * ((1.50991148437421 - PC1) * (1.50991148437421 - PC1) + (-0.643207144005209 - PC2) * (-0.643207144005209 - PC2) + (-0.895869935448295 - PC3) * (-0.895869935448295 - PC3) + (-1.10500770123343 - PC4) * (-1.10500770123343 - PC4)))) - 10.0 * (exp(-0.011 * ((-0.563199188986795 - PC1) * (-0.563199188986795 - PC1) + (-0.387728585163602 - PC2) * (-0.387728585163602 - PC2) + (-0.504681285917712 - PC3) * (-0.504681285917712 - PC3) + (0.293644099151864 - PC4) * (0.293644099151864 - PC4)))) - 10.0 * (exp(-0.011 * ((-1.77895159744526 - PC1) * (-1.77895159744526 - PC1) + (0.239134158727805 - PC2) * (0.239134158727805 - PC2) + (-0.499665090001785 - PC3) * (-0.499665090001785 - PC3) + (0.600533899096768 - PC4) * (0.600533899096768 - PC4)))) - 10.0 * (exp(-0.011 * ((1.40195327075401 - PC1) * (1.40195327075401 - PC1) + (0.380219358734338 - PC2) * (0.380219358734338 - PC2) + (1.82855658052048 - PC3) * (1.82855658052048 - PC3) + (-0.0910666808645269 - PC4) * (-0.0910666808645269 - PC4)))) + 9.999999999999998 * (exp(-0.011 * ((-1.53406159561956 - PC1) * (-1.53406159561956 - PC1) + (-0.469444437824621 - PC2) * (-0.469444437824621 - PC2) + (0.590497565355632 - PC3) * (0.590497565355632 - PC3) + (0.347185644580347 - PC4) * (0.347185644580347 - PC4)))) + 10.0 * (exp(-0.011 * ((-0.0230665876369244 - PC1) * (-0.0230665876369244 - PC1) + (-0.244335191133702 - PC2) * (-0.244335191133702 - PC2) + (-1.40245339694862 - PC3) * (-1.40245339694862 - PC3) + (-0.0559426014688602 - PC4) * (-0.0559426014688602 - PC4)))) - 10.0 * (exp(-0.011 * ((-0.174769167731904 - PC1) * (-0.174769167731904 - PC1) + (0.234574362092708 - PC2) * (0.234574362092708 - PC2) + (-0.3585832764023 - PC3) * (-0.3585832764023 - PC3) + (1.17714476357427 - PC4) * (1.17714476357427 - PC4)))) + 10.0 * (exp(-0.011 * ((-2.09477041995102 - PC1) * (-2.09477041995102 - PC1) +

(0.559507747867692 - PC2) * (0.559507747867692 - PC2) + (-0.776504219103968 - PC3) * (-0.776504219103968 - PC3) + (0.606879563371099 - PC4) * (0.606879563371099 - PC4)))) + 10.0 * (exp(-0.011 * ((-0.0379776798421314 - PC1) * (-0.0379776798421314 - PC1) + (-0.749347146279755 - PC2) * (-0.749347146279755 - PC2) + (-0.696499689534873 - PC3) * (-0.696499689534873 - PC3) + (-1.08595056880636 - PC4) * (-1.08595056880636 - PC4)))) - 10.0 * (exp(-0.011 * ((0.181533453057431 - PC1) * (0.181533453057431 - PC1) + (-0.93071588891874 - PC2) * (-0.93071588891874 - PC2) + (-0.164117744654376 - PC3) * (-0.164117744654376 - PC3) + (-0.618675651019506 - PC4) * (-0.618675651019506 - PC4)))) + 10.0 * (exp(-0.011 * ((0.409419665148064 - PC1) * (0.409419665148064 - PC1) + (-0.768190969999876 - PC2) * (-0.768190969999876 - PC2) + (-0.392179322926475 - PC3) * (-0.392179322926475 - PC3) + (-0.286259634030824 - PC4) * (-0.286259634030824 - PC4)))) + 10.0 * (exp(-0.011 * ((-1.48858521223006 - PC1) * (-1.48858521223006 - PC1) + (0.298871188698009 - PC2) * (0.298871188698009 - PC2) + (-0.479923642216785 - PC3) * (-0.479923642216785 - PC3) + (0.450573538785118 - PC4) * (0.450573538785118 - PC4)))) + 10.0 * (exp(-0.011 * ((-1.55238686861092 - PC1) * (-1.55238686861092 - PC1) + (0.335021304073101 - PC2) * (0.335021304073101 - PC2) + (-0.224592818164722 - PC3) * (-0.224592818164722 - PC3) + (0.748108375534561 - PC4) * (0.748108375534561 - PC4)))) + 10.0 * (exp(-0.011 * ((0.0175791040018881 - PC1) * (0.0175791040018881 - PC1) + (-0.0155242561150388 - PC2) * (-0.0155242561150388 - PC2) + (-0.548187286209886 - PC3) * (-0.548187286209886 - PC3) + (1.18032154004909 - PC4) * (1.18032154004909 - PC4)))) - 10.0 * (exp(-0.011 * ((1.20927955253359 - PC1) * (1.20927955253359 - PC1) + (1.0957890536542 - PC2) * (1.0957890536542 - PC2) + (-0.424703135994616 - PC3) * (-0.424703135994616 - PC3) + (2.61629357993536 - PC4) * (2.61629357993536 - PC4)))) - 10.0 * (exp(-0.011 * ((-2.67949391301711 - PC1) * (-2.67949391301711 - PC1) + (-0.89455503026447 - PC2) * (-0.89455503026447 - PC2) + (1.38593480062983 - PC3) * (1.38593480062983 - PC3) + (1.08802514223279 - PC4) * (1.08802514223279 - PC4)))) - 10.0 * (exp(-0.011 * ((-1.3079433963958 - PC1) * (-1.3079433963958 - PC1) + (3.41644207595301 - PC2) * (3.41644207595301 - PC2) + (0.215260966727677 - PC3) * (0.215260966727677 - PC3) + (-1.27514394384923 - PC4) * (-1.27514394384923 - PC4))))+ 5.008925182037618 * (exp(-0.011 * ((1.12287130825826 - PC1) * (1.12287130825826 - PC1) + (0.700628251676151 - PC2) * (0.700628251676151 - PC2) + (-0.251297310277388 - PC3) * (-0.251297310277388 - PC3) + (-0.453418747963015 - PC4) * (-0.453418747963015 - PC4)))) + 2.842127935677052 * (exp(-0.011 * ((1.74536940273848 - PC1) * (1.74536940273848 - PC1) + (0.875696563695493 - PC2) * (0.875696563695493 - PC2) + (-1.04000676221927 - PC3) * (-1.04000676221927 - PC3) + (2.48455452414513 - PC4) * (2.48455452414513 - PC4)))) - 10.0 * (exp(-0.011 * ((0.0334222365791599 - PC1) * (0.0334222365791599 - PC1) + (-0.970589035965259 - PC2) * (-0.970589035965259 - PC2) + (-0.452652391808868 - PC3) * (-0.452652391808868 - PC3) + (-0.254786525990694 - PC4) * (-0.254786525990694 - PC4)))) - 10.0 * (exp(-0.011 * ((-1.82798438391173 - PC1) * (-1.82798438391173 - PC1) + (3.17439449310042 - PC2) * (3.17439449310042 - PC2) + (-0.327689141587259 - PC3) * (-0.327689141587259 - PC3) + (-0.76355227516701 - PC4) * (-0.76355227516701 - PC4)))) - 10.0 * (exp(-0.011 * ((0.896589583214106 - PC1) * (0.896589583214106 - PC1) + (-1.02356627608125 - PC2) * (-1.02356627608125 - PC2) + (-0.513694786250737 - PC3) * (-0.513694786250737 - PC3) + (-0.317522744152691 - PC4) * (-0.317522744152691 - PC4)))) - 10.0 * (exp(-0.011 * ((-0.330509432890552 - PC1) * (-0.330509432890552 - PC1) + (-0.114759607575904 - PC2) * (-0.114759607575904 - PC2) + (1.19552994820281 - PC3) * (1.19552994820281 - PC3) + (-0.0366152951785867 - PC4) * (-0.0366152951785867 - PC4)))) + 6.746479305677011 * (exp(-0.011 * ((4.20752103219153 - PC1) * (4.20752103219153 - PC1) + (1.33285222440913 - PC2) * (1.33285222440913 - PC2) + (0.164231584182479 - PC3) * (0.164231584182479 - PC3) + (1.94097657968963 - PC4) * (1.94097657968963 - PC4)))) - 10.0 * (exp(-0.011 * ((-0.785852989241174 - PC1) * (-0.785852989241174 - PC1) + (0.0140462566239975 - PC2) * (0.0140462566239975 - PC2) + (0.393554773666952 - PC3) * (0.393554773666952 - PC3) + (-0.529063709775171 - PC4) * (-0.529063709775171 - PC4)))) + 10.0 * (exp(-0.011 * ((3.62540451398198 - PC1) * (3.62540451398198 - PC1) + (0.537420351897033 - PC2) * (0.537420351897033 - PC2) + (2.74033486327062 - PC3) * (2.74033486327062 - PC3) + (-1.35106971165229 - PC4) * (-1.35106971165229 - PC4)))) - 8.190320002861124 * (exp(-0.011 * ((-1.68721483517694 - PC1) * (-1.68721483517694 - PC1) + (5.40742499843657 - PC2) * (5.40742499843657 - PC2) + (-1.36715287754979 - PC3) * (-1.36715287754979 - PC3) + (0.427853508701133 - PC4) * (0.427853508701133 - PC4)))) + 10.0 * (exp(-0.011 * ((-2.54765320295838 - PC1) *

(-2.54765320295838 - PC1) + (-1.90079839667525 - PC2) * (-1.90079839667525 - PC2) + (1.94142375429526 - PC3) * (1.94142375429526 - PC3) + (0.39028498793247 - PC4) * (0.39028498793247 - PC4)))) - 10.0 * (exp(-0.011 * ((-1.69657423107131 - PC1) * (-1.69657423107131 - PC1) + (-2.12747852995009 - PC2) * (-2.12747852995009 - PC2) + (1.14886929023336 - PC3) * (1.14886929023336 - PC3) + (0.0603306253481178 - PC4) * (0.0603306253481178 - PC4)))) + 0.35936917167250976 * (exp(-0.011 * ((3.10612987940629 - PC1) * (3.10612987940629 - PC1) + (2.24824538537725 - PC2) * (2.24824538537725 - PC2) + (5.00097585946766 - PC3) * (5.00097585946766 - PC3) + (0.243855040400053 - PC4) * (0.243855040400053 - PC4)))) + 10.0 * (exp(-0.011 * ((-0.390008776024683 - PC1) * (-0.390008776024683 - PC1) + (-0.907868223859072 - PC2) * (-0.907868223859072 - PC2) + (0.230236633542902 - PC3) * (0.230236633542902 - PC3) + (0.929384804700769 - PC4) * (0.929384804700769 - PC4)))) + 10.0 * (exp(-0.011 * ((-0.244604616889346 - PC1) * (-0.244604616889346 - PC1) + (-0.147673537432589 - PC2) * (-0.147673537432589 - PC2) + (1.51735784350833 - PC3) * (1.51735784350833 - PC3) + (-1.08098173559071 - PC4) * (-1.08098173559071 - PC4)))) + 10.0 * (exp(-0.011 * ((-1.27600901315776 - PC1) * (-1.27600901315776 - PC1) + (5.12733229073173 - PC2) * (5.12733229073173 - PC2) + (-1.71710821121203 - PC3) * (-1.71710821121203 - PC3) + (0.844355083314624 - PC4) * (0.844355083314624 - PC4)))) + 1.2218254297672049

... (5.33)

where $\widehat{ST}$ refer to the normalized values of *ST*; the magnitudes of *PCs* in equation (5.33) are calculated using the equations (5.14) to (5.17).

5. F.3: SVR model pertaining to the *Hemispherical Temperature* (HT) prediction

$\widehat{HT} =$ 0.10091889723756443 * (pow((exp(-0.098 * (-0.24330426808179723 - PC1) * (-0.24330426808179723 - PC1)) + exp(-0.098 * (-0.48032849601574185 - PC2) * (-0.48032849601574185 - PC2)) + exp(-0.098 * (-0.4582751827072967 - PC3) * (-0.4582751827072967 - PC3)) + exp(-0.098 * (0.07790018099031398 - PC4) * (0.07790018099031398 - PC4)) + exp(-0.098 * (0.2899211689219697 - PC5) * (0.2899211689219697 - PC5))), 2.7))- 0.015333943107213081 * (pow((exp(-0.098 * (2.6623318781955123 - PC1) * (2.6623318781955123 - PC1)) + exp(-0.098 * (-2.3100105997709557 - PC2) * (-2.3100105997709557 - PC2)) + exp(-0.098 * (-1.847027970043396 - PC3) * (-1.847027970043396 - PC3)) + exp(-0.098 * (-4.017484720947124 - PC4) * (-4.017484720947124 - PC4)) + exp(-0.098 * (0.9314719404292501 - PC5) * (0.9314719404292501 - PC5))), 2.7)) + 0.0191869328239112 * (pow((exp(-0.098 * (-0.9145626719960867 - PC1) * (-0.9145626719960867 - PC1)) + exp(-0.098 * (1.0084500300204007 - PC2) * (1.0084500300204007 - PC2)) + exp(-0.098 * (-1.2749746960449873 - PC3) * (-1.2749746960449873 - PC3)) + exp(-0.098 * (-0.14092265789287553 - PC4) * (-0.14092265789287553 - PC4)) + exp(-0.098 * (0.2746411120987484 - PC5) * (0.2746411120987484 - PC5))), 2.7))+ 0.44015518035448586 * (pow((exp(-0.098 * (0.19842100431666673 - PC1) * (0.19842100431666673 - PC1)) + exp(-0.098 * (0.8205283840322323 - PC2) * (0.8205283840322323 - PC2)) + exp(-0.098 * (1.0942336164737463 - PC3) * (1.0942336164737463 - PC3)) + exp(-0.098 * (0.20088201312616139 - PC4) * (0.20088201312616139 - PC4)) + exp(-0.098 * (-0.13178395964591733 - PC5) * (-0.13178395964591733 - PC5))), 2.7)) + 0.49031343844342246 * (pow((exp(-0.098 * (-0.36292139414261393 - PC1) * (-0.36292139414261393 - PC1)) + exp(-0.098 * (0.8108874893261458 - PC2) * (0.8108874893261458 - PC2)) + exp(-0.098 * 0.7730223382802965 - PC3) * (0.7730223382802965 - PC3)) + exp(-0.098 * (0.036420564818957335 - PC4) * (0.036420564818957335 - PC4)) + exp(-0.098 * (0.21188958216089857 - PC5) * (0.21188958216089857 - PC5))), 2.7))+ 0.10392439856633578 * (pow((exp(-0.098 * (-0.8872331185997165 - PC1) * (-0.8872331185997165 - PC1)) + exp(-0.098 * (-0.04651798635775228 - PC2) * (-0.04651798635775228 - PC2)) + exp(-0.098 * (0.2818189526759598 - PC3) * (0.2818189526759598 - PC3)) + exp(-0.098 * (-0.6413862405230264 - PC4) * (-0.6413862405230264 - PC4)) + exp(-0.098 * (0.31197380692218313 - PC5) * (0.31197380692218313 - PC5))), 2.7))- 0.03936365270872982 * (pow((exp(-0.098 * (2.816126109422459 - PC1) * (2.816126109422459 - PC1)) + exp(-0.098 * (1.216424668738208 - PC2) * (1.216424668738208 - PC2)) + exp(-0.098 * (2.5659724395189354 - PC3) * (2.5659724395189354 - PC3)) + exp(-0.098 * (-0.4215421754628234 - PC4) * (-0.4215421754628234 - PC4)) + exp(-0.098 * (0.4280245015571141 - PC5) * (0.4280245015571141 - PC5))), 2.7))+ 0.10904638860018348 * (pow((exp(-0.098 * (-1.1784494301572983 - PC1) * (-1.1784494301572983 - PC1)) + exp(-0.098 * (0.18000581987747274 - PC2) * (0.18000581987747274 - PC2)) + exp(-0.098 * (0.8325626944280203 - PC3) * (0.8325626944280203 - PC3)) + exp(-0.098 * (0.709525347277057 - PC4) * (0.709525347277057 - PC4)) + exp(-0.098 * (-0.7163454388476432 - PC5) * (-0.7163454388476432 - PC5))), 2.7))+ 0.08093306629327965 * (pow((exp(-0.098 * (0.13633605265697057 - PC1) * (0.13633605265697057 - PC1)) + exp(-0.098 * (0.08296077976645468 - PC2) * (0.08296077976645468 - PC2)) + exp(-0.098 * (-0.3630250489847945 - PC3) * (-0.3630250489847945 - PC3)) + exp(-0.098 * (0.4533418598501077 - PC4) * (0.4533418598501077 - PC4)) + exp(-0.098 * (-1.333261730549658 - PC5) * (-1.333261730549658 - PC5))), 2.7)) - 0.024580325901031225 * (pow((exp(-0.098 * (-1.4497740382260942 - PC1) * (-1.4497740382260942 - PC1)) + exp(-0.098 * (-3.543060023861607 - PC2) * (-3.543060023861607 - PC2)) + exp(-0.098 * (2.6055385404161133 - PC3) * (2.6055385404161133 - PC3)) + exp(-0.098 * (-0.17273006763736126 - PC4) * (-0.17273006763736126 - PC4)) + exp(-0.098 * (1.0539406067422772 - PC5) * (1.0539406067422772 - PC5))), 2.7))- 0.19232534944363597 * (pow((exp(-0.098 * (0.7327337035165653 - PC1) * (0.7327337035165653 - PC1)) + exp(-0.098 * (-0.39531394456336794 - PC2) * (-0.39531394456336794 - PC2)) + exp(-0.098 * (-

1.379030467220063 - PC3) * (-1.379030467220063 - PC3)) + exp(-0.098 * (0.5660197663995784 - PC4) * (0.5660197663995784 - PC4)) + exp(-0.098 * (-2.758443793767156 - PC5) * (-2.758443793767156 - PC5))), 2.7))+ 0.5022488253463897 * (pow((exp(-0.098 * (-1.3009135885636098 - PC1) * (-1.3009135885636098 - PC1)) + exp(-0.098 * (1.3658684277099082 - PC2) * (1.3658684277099082 - PC2)) + exp(-0.098 * (-1.4862301849376662 - PC3) * (-1.4862301849376662 - PC3)) + exp(-0.098 * (0.10091690262625563 - PC4) * (0.10091690262625563 - PC4)) + exp(-0.098 * (-0.3947624175669415 - PC5) * (-0.3947624175669415 - PC5))), 2.7))- 0.26272741386116844 * (pow((exp(-0.098 * (-0.1184007708504788 - PC1) * (-0.1184007708504788 - PC1)) + exp(-0.098 * (0.6201532886412443 - PC2) * (0.6201532886412443 - PC2)) + exp(-0.098 * (0.9909842848255979 - PC3) * (0.9909842848255979 - PC3)) + exp(-0.098 * (0.8583112544228613 - PC4) * (0.8583112544228613 - PC4)) + exp(-0.098 * (-0.8284293681455501 - PC5) * (-0.8284293681455501 - PC5))), 2.7))+ 0.03169968054575478 * (pow((exp(-0.098 * (1.3129876294043967 - PC1) * (1.3129876294043967 - PC1)) + exp(-0.098 * (-0.8962688862898223 - PC2) * (-0.8962688862898223 - PC2)) + exp(-0.098 * (-0.10600422317027504 - PC3) * (-0.10600422317027504 - PC3)) + exp(-0.098 * (0.3667835426134578 - PC4) * (0.3667835426134578 - PC4)) + exp(-0.098 * (-0.08883302611377125 - PC5) * (-0.08883302611377125 - PC5))), 2.7))- 0.04309011785537282 * (pow((exp(-0.098 * (2.1634194485190896 - PC1) * (2.1634194485190896 - PC1)) + exp(-0.098 * (1.0421549020577168 - PC2) * (1.0421549020577168 - PC2)) + exp(-0.098 * (1.6192180272388075 - PC3) * (1.6192180272388075 - PC3)) + exp(-0.098 * (-0.45890583489823056 - PC4) * (-0.45890583489823056 - PC4)) + exp(-0.098 * (1.2993368945886257 - PC5) * (1.2993368945886257 - PC5))), 2.7))+ 0.5305727061640492 * (pow((exp(-0.098 * (0.34337258311415225 - PC1) * (0.34337258311415225 - PC1)) + exp(-0.098 * (0.1134827859130497 - PC2) * (0.1134827859130497 - PC2)) + exp(-0.098 * (-0.5881286661144398 - PC3) * (-0.5881286661144398 - PC3)) + exp(-0.098 * (-1.481057555295546 - PC4) * (-1.481057555295546 - PC4)) + exp(-0.098 * (1.5049000450005416 - PC5) * (1.5049000450005416 - PC5))), 2.7))- 5.289261912898404E-4 * (pow((exp(-0.098 * (0.696170188233415 - PC1) * (0.696170188233415 - PC1)) + exp(-0.098 * (-0.11248589400295197 - PC2) * (-0.11248589400295197 - PC2)) + exp(-0.098 * (0.03168322118119695 - PC3) * (0.03168322118119695 - PC3)) + exp(-0.098 * (-0.10774679673462449 - PC4) * (-0.10774679673462449 - PC4)) + exp(-0.098 * (1.6434882738205092 - PC5) * (1.6434882738205092 - PC5))), 2.7))- 0.2274992826506121 * (pow((exp(-0.098 * (-0.37450406609755615 - PC1) * (-0.37450406609755615 - PC1)) + exp(-0.098 * (0.6092000145682899 - PC2) * (0.6092000145682899 - PC2)) + exp(-0.098 * (-0.0059955268446347946 - PC3) * (-0.0059955268446347946 - PC3)) + exp(-0.098 * (0.030624857914989385 - PC4) * (0.030624857914989385 - PC4)) + exp(-0.098 * (-0.09090203984913423 - PC5) * (-0.09090203984913423 - PC5))), 2.7))+ 0.13363000753449908 * (pow((exp(-0.098 * (0.9981627083262239 - PC1) * (0.9981627083262239 - PC1)) + exp(-0.098 * (-1.1155363905985134 - PC2) * (-1.1155363905985134 - PC2)) + exp(-0.098 * (-0.46549560268553825 - PC3) * (-0.46549560268553825 - PC3)) + exp(-0.098 * (-0.5562102844401146 - PC4) * (-0.5562102844401146 - PC4)) + exp(-0.098 * (-2.171345971896689 - PC5) * (-2.171345971896689 - PC5))), 2.7))- 0.00816795106717532 * (pow((exp(-0.098 * (0.6759011450079427 - PC1) * (0.6759011450079427 - PC1)) + exp(-0.098 * (-0.16706099888156786 - PC2) * (-0.16706099888156786 - PC2)) + exp(-0.098 * (-1.1875128250870457 - PC3) * (-1.1875128250870457 - PC3)) + exp(-0.098 * (3.397060828779145 - PC4) * (3.397060828779145 - PC4)) + exp(-0.098 * (3.136406538284857 - PC5) * (3.136406538284857 - PC5))), 2.7))+ 0.24256154445816777 * (pow((exp(-0.098 * (-1.0676721070445117 - PC1) * (-1.0676721070445117 - PC1)) + exp(-0.098 * (-2.9703843749042376 - PC2) * (-2.9703843749042376 - PC2)) + exp(-0.098 * (2.17641048828956 - PC3) * (2.17641048828956 - PC3)) + exp(-0.098 * (1.6989013200542686 - PC4) * (1.6989013200542686 - PC4)) + exp(-0.098 * (-0.8445123109601973 - PC5) * (-0.8445123109601973 - PC5))), 2.7))+ 0.31729189116563866 * (pow((exp(-0.098 * (1.0557318039873917 - PC1) * (1.0557318039873917 - PC1)) + exp(-0.098 * (-0.8536343436709896 - PC2) * (-0.8536343436709896 - PC2)) + exp(-0.098 * (-0.8100207344751506 - PC3) * (-0.8100207344751506 - PC3)) + exp(-0.098 * (-0.29138681515014625 - PC4) * (-0.29138681515014625 - PC4)) + exp(-0.098 * (0.5793598908437294 - PC5) *

(0.5793598908437294 - PC5))), 2.7))+ 0.061837330060731714 * (pow((exp(-0.098 * (-1.4002398657995658 - PC1) * (-1.4002398657995658 - PC1)) + exp(-0.098 * (-1.7732074232397323 - PC2) * (-1.7732074232397323 - PC2)) + exp(-0.098 * (1.4721316242794888 - PC3) * (1.4721316242794888 - PC3)) + exp(-0.098 * (0.32097584031188553 - PC4) * (0.32097584031188553 - PC4)) + exp(-0.098 * (-0.2229851597807384 - PC5) * (-0.2229851597807384 - PC5))), 2.7))- 0.41911764705882354 * (pow((exp(-0.098 * (0.7972238554274257 - PC1) * (0.7972238554274257 - PC1)) + exp(-0.098 * (-0.8111983052747952 - PC2) * (-0.8111983052747952 - PC2)) + exp(-0.098 * (-0.17336629251529476 - PC3) * (-0.17336629251529476 - PC3)) + exp(-0.098 * (-1.4287312922375148 - PC4) * (-1.4287312922375148 - PC4)) + exp(-0.098 * (0.12670527412866694 - PC5) * (0.12670527412866694 - PC5))), 2.7))- 0.4191176470588236 * (pow((exp(-0.098 * (-0.6999468936736798 - PC1) * (-0.6999468936736798 - PC1)) + exp(-0.098 * (0.4192116954138775 - PC2) * (0.4192116954138775 - PC2)) + exp(-0.098 * (-0.5798386845787662 - PC3) * (-0.5798386845787662 - PC3)) + exp(-0.098 * (-0.5703275718531158 - PC4) * (-0.5703275718531158 - PC4)) + exp(-0.098 * (0.34484059293246044 - PC5) * (0.34484059293246044 - PC5))), 2.7))+ 0.6195652173913043 * (pow((exp(-0.098 * (-0.9028254783978452 - PC1) * (-0.9028254783978452 - PC1)) + exp(-0.098 * (-0.08356100109477506 - PC2) * (-0.08356100109477506 - PC2)) + exp(-0.098 * (-0.013694857401745248 - PC3) * (-0.013694857401745248 - PC3)) + exp(-0.098 * (-0.6162303415771668 - PC4) * (-0.6162303415771668 - PC4)) + exp(-0.098 * (0.5319968789597246 - PC5) * (0.5319968789597246 - PC5))), 2.7))- 0.41911764705882354 * (pow((exp(-0.098 * (-0.08203391059556273 - PC1) * (-0.08203391059556273 - PC1)) + exp(-0.098 * (0.6960689449362187 - PC2) * (0.6960689449362187 - PC2)) + exp(-0.098 * (0.4126861271403178 - PC3) * (0.4126861271403178 - PC3)) + exp(-0.098 * (-0.02156770696807815 - PC4) * (-0.02156770696807815 - PC4)) + exp(-0.098 * (-1.0812206333142538 - PC5) * (-1.0812206333142538 - PC5))), 2.7))- 0.41911764705882354 * (pow((exp(-0.098 * (0.9945065606206327 - PC1) * (0.9945065606206327 - PC1)) + exp(-0.098 * (-0.10858553349491797 - PC2) * (-0.10858553349491797 - PC2)) + exp(-0.098 * (-0.04253126834561452 - PC3) * (-0.04253126834561452 - PC3)) + exp(-0.098 * (0.015594366634567413 - PC4) * (0.015594366634567413 - PC4)) + exp(-0.098 * (1.0483254232875934 - PC5) * (1.0483254232875934 - PC5))), 2.7))+ 0.6195652173913043 * (pow((exp(-0.098 * (-0.8741387427588942 - PC1) * (-0.8741387427588942 - PC1)) + exp(-0.098 * (0.44806653707707406 - PC2) * (0.44806653707707406 - PC2)) + exp(-0.098 * (-0.3631078146019596 - PC3) * (-0.3631078146019596 - PC3)) + exp(-0.098 * (-0.15543841647181908 - PC4) * (-0.15543841647181908 - PC4)) + exp(-0.098 * (-0.10989393872166346 - PC5) * (-0.10989393872166346 - PC5))), 2.7))+ 0.6195652173913043 * (pow((exp(-0.098 * (0.04392782681741657 - PC1) * (0.04392782681741657 - PC1)) + exp(-0.098 * (0.5594544083198002 - PC2) * (0.5594544083198002 - PC2)) + exp(-0.098 * (0.3885945605909807 - PC3) * (0.3885945605909807 - PC3)) + exp(-0.098 * (-0.12477126895534217 - PC4) * (-0.12477126895534217 - PC4)) + exp(-0.098 * (-0.4895894412636865 - PC5) * (-0.4895894412636865 - PC5))), 2.7))- 0.41911764705882354 * (pow((exp(-0.098 * (-0.20712231391588393 - PC1) * (-0.20712231391588393 - PC1)) + exp(-0.098 * (0.036880195125216154 - PC2) * (0.036880195125216154 - PC2)) + exp(-0.098 * (-0.343999682611987 - PC3) * (-0.343999682611987 - PC3)) + exp(-0.098 * (-1.1424327820413174 - PC4) * (-1.1424327820413174 - PC4)) + exp(-0.098 * (1.56218160356869 - PC5) * (1.56218160356869 - PC5))), 2.7)) - 0.4191176470588236 * (pow((exp(-0.098 * (-0.6274305934302018 - PC1) * (-0.6274305934302018 - PC1)) + exp(-0.098 * (0.15575613171141378 - PC2) * (0.15575613171141378 - PC2)) + exp(-0.098 * (0.3081566095876752 - PC3) * (0.3081566095876752 - PC3)) + exp(-0.098 * (0.354485086009518 - PC4) * (0.354485086009518 - PC4)) + exp(-0.098 * (-0.49551241846263816 - PC5) * (-0.49551241846263816 - PC5))), 2.7))- 0.41911764705882354 * (pow((exp(-0.098 * (-0.11789058240511749 - PC1) * (-0.11789058240511749 - PC1)) + exp(-0.098 * (0.7182796915744194 - PC2) * (0.7182796915744194 - PC2)) + exp(-0.098 * (0.31409362168139693 - PC3) * (0.31409362168139693 - PC3)) + exp(-0.098 * (-0.2095672890325668 - PC4) * (-0.2095672890325668 - PC4)) + exp(-0.098 * (-0.16838485951651105 - PC5) * (-0.16838485951651105 - PC5))), 2.7))+ 0.6195652173913043 * (pow((exp(-0.098 * (-0.448382732758669 - PC1) * (-0.448382732758669 - PC1)) + exp(-0.098 *

(0.2503309479376489 - PC2) * (0.2503309479376489 - PC2)) + exp(-0.098 * (-0.17753008063928702 - PC3) * (-0.17753008063928702 - PC3)) + exp(-0.098 * (1.0949097898648736 - PC4) * (1.0949097898648736 - PC4)) + exp(-0.098 * (-1.5379647141567385 - PC5) * (-1.5379647141567385 - PC5))), 2.7))- 0.4191176470588236 * (pow((exp(-0.098 * (-0.3439004663838459 - PC1) * (-0.3439004663838459 - PC1)) + exp(-0.098 * (0.19469978924813322 - PC2) * (0.19469978924813322 - PC2)) + exp(-0.098 * (-0.2439279944724757 - PC3) * (-0.2439279944724757 - PC3)) + exp(-0.098 * (0.6360550881011969 - PC4) * (0.6360550881011969 - PC4)) + exp(-0.098 * (-0.6912134022326267 - PC5) * (-0.6912134022326267 - PC5))), 2.7))- 0.4191176470588235 * (pow((exp(-0.098 * (0.30639683250173155 - PC1) * (0.30639683250173155 - PC1)) + exp(-0.098 * (0.7024369787194964 - PC2) * (0.7024369787194964 - PC2)) + exp(-0.098 * (0.4245924794757452 - PC3) * (0.4245924794757452 - PC3)) + exp(-0.098 * (-0.17060309477667818 - PC4) * (-0.17060309477667818 - PC4)) + exp(-0.098 * (-0.7373252278050881 - PC5) * (-0.7373252278050881 - PC5))), 2.7))+ 0.6195652173913043 * (pow((exp(-0.098 * (1.0153856936848906 - PC1) * (1.0153856936848906 - PC1)) + exp(-0.098 * (0.7474999520635012 - PC2) * (0.7474999520635012 - PC2)) + exp(-0.098 * (0.7650798204461697 - PC3) * (0.7650798204461697 - PC3)) + exp(-0.098 * (-0.44414889788276946 - PC4) * (-0.44414889788276946 - PC4)) + exp(-0.098 * (0.3896164247598161 - PC5) * (0.3896164247598161 - PC5))), 2.7))- 0.41911764705882354 * (pow((exp(-0.098 * (-1.222856629937907 - PC1) * (-1.222856629937907 - PC1)) + exp(-0.098 * (-1.915903938916697 - PC2) * (-1.915903938916697 - PC2)) + exp(-0.098 * (1.544924327305037 - PC3) * (1.544924327305037 - PC3)) + exp(-0.098 * (1.0844975763675515 - PC4) * (1.0844975763675515 - PC4)) + exp(-0.098 * (-0.28248195957500755 - PC5) * (-0.28248195957500755 - PC5))), 2.7))- 0.4191176470588236 * (pow((exp(-0.098 * (0.29232377548813415 - PC1) * (0.29232377548813415 - PC1)) + exp(-0.098 * (0.874961851074236 - PC2) * (0.874961851074236 - PC2)) + exp(-0.098 * (1.1842398092030713 - PC3) * (1.1842398092030713 - PC3)) + exp(-0.098 * (-0.42895071524957146 - PC4) * (-0.42895071524957146 - PC4)) + exp(-0.098 * (0.4746768671444466 - PC5) * (0.4746768671444466 - PC5))), 2.7))- 0.41911764705882354 * (pow((exp(-0.098 * (-0.8736229079997044 - PC1) * (-0.8736229079997044 - PC1)) + exp(-0.098 * (1.4991921221184028 - PC2) * (1.4991921221184028 - PC2)) + exp(-0.098 * (-0.9031900479780239 - PC3) * (-0.9031900479780239 - PC3)) + exp(-0.098 * (0.07806420519690872 - PC4) * (0.07806420519690872 - PC4)) + exp(-0.098 * (-0.07518844052057602 - PC5) * (-0.07518844052057602 - PC5))), 2.7))- 0.41911764705882354 * (pow((exp(-0.098 * (-1.3906848320055007 - PC1) * (-1.3906848320055007 - PC1)) + exp(-0.098 * (0.6077581876980701 - PC2) * (0.6077581876980701 - PC2)) + exp(-0.098 * (-1.0808626963877783 - PC3) * (-1.0808626963877783 - PC3)) + exp(-0.098 * (-0.22890751755183525 - PC4) * (-0.22890751755183525 - PC4)) + exp(-0.098 * (0.5041389449055688 - PC5) * (0.5041389449055688 - PC5))), 2.7))+ 1.3477749024115757

... (5.34)

where $\widehat{HT}$ refer to the normalized values of *HT*; the magnitudes of *PCs* in equation (5.34) are calculated using the equations (5.19) to (5.23).

5. F.4: SVR model pertaining to the *Fluid Temperature* (FT) prediction

$\widehat{FT} =$ 0.47777143209305456 * (pow((exp(-1.0 * (0.9732039175972192 - PC1) * (0.9732039175972192 - PC1)) + exp(-1.0 * (1.4308095728478971 - PC2) * (1.4308095728478971 - PC2)) + exp(-1.0 * (-1.0897309535344422 - PC3) * (-1.0897309535344422 - PC3)) + exp(-1.0 * (-0.7960126528289836 - PC4) * (-0.7960126528289836 - PC4)) + exp(-1.0 * (-0.19323503614199544 - PC5) * (-0.19323503614199544 - PC5))), 1.0))+ 0.8991213437276867 * (pow((exp(-1.0 * (1.1439406213837588 - PC1) * (1.1439406213837588 - PC1)) + exp(-1.0 * (1.6744736577523833 - PC2) * (1.6744736577523833 - PC2)) + exp(-1.0 * (1.1216799335077527 - PC3) * (1.1216799335077527 - PC3)) + exp(-1.0 * (-0.47707726144145307 - PC4) * (-0.47707726144145307 - PC4)) + exp(-1.0 * (0.1748961889068372 - PC5) * (0.1748961889068372 - PC5))), 1.0))+ 0.29466068399954837 * (pow((exp(-1.0 * (-0.0972149487917619 - PC1) * (-0.0972149487917619 - PC1)) + exp(-1.0 * (-1.4351985169077643 - PC2) * (-1.4351985169077643 - PC2)) + exp(-1.0 * (-0.38700516714156247 - PC3) * (-0.38700516714156247 - PC3)) + exp(-1.0 * (0.1918369656442798 - PC4) * (0.1918369656442798 - PC4)) + exp(-1.0 * (-1.0099306574662568 - PC5) * (-1.0099306574662568 - PC5))), 1.0))- 0.9188297013763417 * (pow((exp(-1.0 * (2.3976480753484486 - PC1) * (2.3976480753484486 - PC1)) + exp(-1.0 * (1.0368988953529834 - PC2) * (1.0368988953529834 - PC2)) + exp(-1.0 * (0.5767211858866957 - PC3) * (0.5767211858866957 - PC3)) + exp(-1.0 * (1.4881760682179535 - PC4) * (1.4881760682179535 - PC4)) + exp(-1.0 * (0.6745903835601345 - PC5) * (0.6745903835601345 - PC5))), 1.0))+ 1.1953149786392663 * (pow((exp(-1.0 * (0.9301387359865164 - PC1) * (0.9301387359865164 - PC1)) + exp(-1.0 * (1.3530518031362677 - PC2) * (1.3530518031362677 - PC2)) + exp(-1.0 * (-0.46727910032544245 - PC3) * (-0.46727910032544245 - PC3)) + exp(-1.0 * (2.6265674470450087 - PC4) * (2.6265674470450087 - PC4)) + exp(-1.0 * (-0.8284556635119927 - PC5) * (-0.8284556635119927 - PC5))), 1.0))- 0.7796051739444981 * (pow((exp(-1.0 * (0.15632456719337343 - PC1) * (0.15632456719337343 - PC1)) + exp(-1.0 * (1.4238653482576358 - PC2) * (1.4238653482576358 - PC2)) + exp(-1.0 * (0.5838400159040255 - PC3) * (0.5838400159040255 - PC3)) + exp(-1.0 * (1.3495247646140212 - PC4) * (1.3495247646140212 - PC4)) + exp(-1.0 * (-1.0380806809955998 - PC5) * (-1.0380806809955998 - PC5))), 1.0))+ 0.04033249960996687 * (pow((exp(-1.0 * (-0.9303307887236191 - PC1) * (-0.9303307887236191 - PC1)) + exp(-1.0 * (0.1509440880408775 - PC2) * (0.1509440880408775 - PC2)) + exp(-1.0 * (1.6845832921948807 - PC3) * (1.6845832921948807 - PC3)) + exp(-1.0 * (0.3409767470554385 - PC4) * (0.3409767470554385 - PC4)) + exp(-1.0 * (-0.2746611489694111 - PC5) * (-0.2746611489694111 - PC5))), 1.0))+ 0.3766429626998204 * (pow((exp(-1.0 * (0.6296979508671355 - PC1) * (0.6296979508671355 - PC1)) + exp(-1.0 * (-1.4537391410275213 - PC2) * (-1.4537391410275213 - PC2)) + exp(-1.0 * (1.1994501407759919 - PC3) * (1.1994501407759919 - PC3)) + exp(-1.0 * (-1.1432252745409386 - PC4) * (-1.1432252745409386 - PC4)) + exp(-1.0 * (-0.27850056336738077 - PC5) * (-0.27850056336738077 - PC5))), 1.0))- 0.43522932111417845 * (pow((exp(-1.0 * (0.44447838534296524 - PC1) * (0.44447838534296524 - PC1)) + exp(-1.0 * (-0.31150843846245224 - PC2) * (-0.31150843846245224 - PC2)) + exp(-1.0 * (-0.6061541985548496 - PC3) * (-0.6061541985548496 - PC3)) + exp(-1.0 * (-0.528799454157723 - PC4) * (-0.528799454157723 - PC4)) + exp(-1.0 * (-0.3176245384439266 - PC5) * (-0.3176245384439266 - PC5))), 1.0))+ 0.3460082126935699 * (pow((exp(-1.0 * (1.151901767903374 - PC1) * (1.151901767903374 - PC1)) + exp(-1.0 * (0.21605533047971254 - PC2) * (0.21605533047971254 - PC2)) + exp(-1.0 * (-1.5691715283407133 - PC3) * (-1.5691715283407133 - PC3)) + exp(-1.0 * (1.0558785991020487 - PC4) * (1.0558785991020487 - PC4)) + exp(-1.0 * (-

```
0.22467071206968584 - PC5) * (-0.22467071206968584 - PC5))), 1.0))-
0.9342904702479011 * (pow((exp(-1.0 * (-1.2834876911139623 - PC1) * (-
1.2834876911139623 - PC1)) + exp(-1.0 * (0.7813920910236144 - PC2) *
(0.7813920910236144 - PC2)) + exp(-1.0 * (-1.1212168639216762 - PC3) * (-
1.1212168639216762 - PC3)) + exp(-1.0 * (-1.4958537484911743 - PC4) * (-
1.4958537484911743 - PC4)) + exp(-1.0 * (-1.3723075318680424 - PC5) * (-
1.3723075318680424 - PC5))), 1.0))- 0.16435582977665916 * (pow((exp(-1.0 * (-
1.4074420372061258 - PC1) * (-1.4074420372061258 - PC1)) + exp(-1.0 * (-
0.5073500971615174 - PC2) * (-0.5073500971615174 - PC2)) + exp(-1.0 *
(0.5408279051656457 - PC3) * (0.5408279051656457 - PC3)) + exp(-1.0 *
(1.6518031682679775 - PC4) * (1.6518031682679775 - PC4)) + exp(-1.0 * (-
0.6727624080493384 - PC5) * (-0.6727624080493384 - PC5))), 1.0))- 0.9552631578947369
* (pow((exp(-1.0 * (-1.38552781261835 - PC1) * (-1.38552781261835 - PC1)) + exp(-1.0 * (-
1.7039118055874325 - PC2) * (-1.7039118055874325 - PC2)) + exp(-1.0 * (-
0.959580874439717 - PC3) * (-0.959580874439717 - PC3)) + exp(-1.0 *
(2.5140365049467843 - PC4) * (2.5140365049467843 - PC4)) + exp(-1.0 *
(0.5867478831990992 - PC5) * (0.5867478831990992 - PC5))), 1.0))+ 0.6787628675964762
* (pow((exp(-1.0 * (-1.1999024112742769 - PC1) * (-1.1999024112742769 - PC1)) + exp(-
1.0 * (0.4389567584553705 - PC2) * (0.4389567584553705 - PC2)) + exp(-1.0 *
(1.049554380843094 - PC3) * (1.049554380843094 - PC3)) + exp(-1.0 *
(0.6693681452375662 - PC4) * (0.6693681452375662 - PC4)) + exp(-1.0 * (-
0.329639545035936 - PC5) * (-0.329639545035936 - PC5))), 1.0)) - 0.5256029674169872 *
(pow((exp(-1.0 * (-0.9552482113306603 - PC1) * (-0.9552482113306603 - PC1)) + exp(-1.0
* (1.364503680267093 - PC2) * (1.364503680267093 - PC2)) + exp(-1.0 * (-
0.9776653490119238 - PC3) * (-0.9776653490119238 - PC3)) + exp(-1.0 * (-
2.565708036396036 - PC4) * (-2.565708036396036 - PC4)) + exp(-1.0 * (-
1.3318809159066576 - PC5) * (-1.3318809159066576 - PC5))), 1.0))- 0.9552631578947369
* (pow((exp(-1.0 * (0.3282918736187309 - PC1) * (0.3282918736187309 - PC1)) + exp(-1.0
* (-0.5250602578973824 - PC2) * (-0.5250602578973824 - PC2)) + exp(-1.0 * (-
0.3407074971760928 - PC3) * (-0.3407074971760928 - PC3)) + exp(-1.0 *
(0.871994709500329 - PC4) * (0.871994709500329 - PC4)) + exp(-1.0 * (-
0.35722101318336463 - PC5) * (-0.35722101318336463 - PC5))), 1.0))-
0.5230542321102395 * (pow((exp(-1.0 * (-1.0366900501389753 - PC1) * (-
1.0366900501389753 - PC1)) + exp(-1.0 * (0.149558715116471 - PC2) *
(0.149558715116471 - PC2)) + exp(-1.0 * (-0.1750646001518295 - PC3) * (-
0.1750646001518295 - PC3)) + exp(-1.0 * (-0.5760705344658554 - PC4) * (-
0.5760705344658554 - PC4)) + exp(-1.0 * (0.8322818939902702 - PC5) *
(0.8322818939902702 - PC5))), 1.0)) - 0.3174425454442844 * (pow((exp(-1.0 * (-
0.292920863564135 - PC1) * (-0.292920863564135 - PC1)) + exp(-1.0 * (-
0.10537266301463469 - PC2) * (-0.10537266301463469 - PC2)) + exp(-1.0 * (-
0.5178778770027747 - PC3) * (-0.5178778770027747 - PC3)) + exp(-1.0 * (-
0.6910816529965188 - PC4) * (-0.6910816529965188 - PC4)) + exp(-1.0 * (-
0.6409945384902461 - PC5) * (-0.6409945384902461 - PC5))), 1.0))+
0.10969713374958272 * (pow((exp(-1.0 * (0.8652159911255344 - PC1) *
(0.8652159911255344 - PC1)) + exp(-1.0 * (-1.8749654987794884 - PC2) * (-
1.8749654987794884 - PC2)) + exp(-1.0 * (-0.45164580541178856 - PC3) * (-
0.45164580541178856 - PC3)) + exp(-1.0 * (0.13284208476456977 - PC4) *
(0.13284208476456977 - PC4)) + exp(-1.0 * (-0.38017374270113063 - PC5) * (-
0.38017374270113063 - PC5))), 1.0))+ 0.17929311380708635 * (pow((exp(-1.0 * (-
1.832608841080855 - PC1) * (-1.832608841080855 - PC1)) + exp(-1.0 *
0.34656580207609333 - PC2) * (0.34656580207609333 - PC2)) + exp(-1.0 * (-
0.19743898511416302 - PC3) * (-0.19743898511416302 - PC3)) + exp(-1.0 *
(1.5277571080144192 - PC4) * (1.5277571080144192 - PC4)) + exp(-1.0 * (-
0.9477343926224424 - PC5) * (-0.9477343926224424 - PC5))), 1.0))- 0.31320329674879366
* (pow((exp(-1.0 * (0.8369515780542807 - PC1) * (0.8369515780542807 - PC1)) + exp(-1.0
* (1.6233114391238614 - PC2) * (1.6233114391238614 - PC2)) + exp(-1.0 *
(2.1111688142720313 - PC3) * (2.1111688142720313 - PC3)) + exp(-1.0 *
(0.2117951594490802 - PC4) * (0.2117951594490802 - PC4)) + exp(-1.0 *
(2.079319098317639 - PC5) * (2.079319098317639 - PC5))), 1.0))+ 0.05095463080501894 *
(pow((exp(-1.0 * (-1.9245530353164513 - PC1) * (-1.9245530353164513 - PC1)) + exp(-1.0
```

* (-0.5020852242900583 - PC2) * (-0.5020852242900583 - PC2)) + exp(-1.0 * (1.135912919610537 - PC3) * (1.135912919610537 - PC3)) + exp(-1.0 * (0.6426010467719301 - PC4) * (0.6426010467719301 - PC4)) + exp(-1.0 * (2.3316951932266003 - PC5) * (2.3316951932266003 - PC5))), 1.0)) + 0.3788221486028322 * (pow((exp(-1.0 * (1.6324638855944151 - PC1) * (1.6324638855944151 - PC1)) + exp(-1.0 * (1.124812688799096 - PC2) * (1.124812688799096 - PC2)) + exp(-1.0 * (0.705652474941605 - PC3) * (0.705652474941605 - PC3)) + exp(-1.0 * (-0.35186984650975045 - PC4) * (-0.35186984650975045 - PC4)) + exp(-1.0 * (0.6635831928415263 - PC5) * (0.6635831928415263 - PC5))), 1.0))- 0.5021657480084027 * (pow((exp(-1.0 * (-0.43042193921889327 - PC1) * (-0.43042193921889327 - PC1)) + exp(-1.0 * (0.1605245640692602 - PC2) * (0.1605245640692602 - PC2)) + exp(-1.0 * (-0.029736423542470285 - PC3) * (-0.029736423542470285 - PC3)) + exp(-1.0 * (-1.57267661114413 - PC4) * (-1.57267661114413 - PC4)) + exp(-1.0 * (2.482091280189985 - PC5) * (2.482091280189985 - PC5))), 1.0))+ 0.11486580531304186 * (pow((exp(-1.0 * (1.4543567266402726 - PC1) * (1.4543567266402726 - PC1)) + exp(-1.0 * (1.896852527616768 - PC2) * (1.896852527616768 - PC2)) + exp(-1.0 * (-2.573862825974282 - PC3) * (-2.573862825974282 - PC3)) + exp(-1.0 * (-0.5737638021917664 - PC4) * (-0.5737638021917664 - PC4)) + exp(-1.0 * (1.1787506029022967 - PC5) * (1.1787506029022967 - PC5))), 1.0))- 0.5742527624164043 * (pow((exp(-1.0 * (-1.301709367516012 - PC1) * (-1.301709367516012 - PC1)) + exp(-1.0 * (0.3941673487626199 - PC2) * (0.3941673487626199 - PC2)) + exp(-1.0 * (0.5271269262122992 - PC3) * (0.5271269262122992 - PC3)) + exp(-1.0 * (-0.6644947899428104 - PC4) * (-0.6644947899428104 - PC4)) + exp(-1.0 * (-0.8576655465124544 - PC5) * (-0.8576655465124544 - PC5))), 1.0)) - 0.3853229117066643 * (pow((exp(-1.0 * (0.23863424153349827 - PC1) * (0.23863424153349827 - PC1)) + exp(-1.0 * (1.1595686012221882 - PC2) * (1.1595686012221882 - PC2)) + exp(-1.0 * (1.2949231639099072 - PC3) * (1.2949231639099072 - PC3)) + exp(-1.0 * (0.7256832508864659 - PC4) * (0.7256832508864659 - PC4)) + exp(-1.0 * (-1.1298787900674192 - PC5) * (-1.1298787900674192 - PC5))), 1.0))- 0.9552631578947369 * (pow((exp(-1.0 * (-1.1701959824420312 - PC1) * (-1.1701959824420312 - PC1)) + exp(-1.0 * (0.49607514793081675 - PC2) * (0.49607514793081675 - PC2)) + exp(-1.0 * (-0.15580141465039807 - PC3) * (-0.15580141465039807 - PC3)) + exp(-1.0 * (-0.8544271448601325 - PC4) * (-0.8544271448601325 - PC4)) + exp(-1.0 * (-0.2993688169160095 - PC5) * (-0.2993688169160095 - PC5))), 1.0))+ 1.2964285714285715 * (pow((exp(-1.0 * (-0.29425333365471273 - PC1) * (-0.29425333365471273 - PC1)) + exp(-1.0 * (-0.7488441651924657 - PC2) * (-0.7488441651924657 - PC2)) + exp(-1.0 * (0.14099610327489845 - PC3) * (0.14099610327489845 - PC3)) + exp(-1.0 * (0.6496346416098849 - PC4) * (0.6496346416098849 - PC4)) + exp(-1.0 * (-0.1382882121338554 - PC5) * (-0.1382882121338554 - PC5))), 1.0))- 0.9552631578947369 * (pow((exp(-1.0 * (-0.2851745656656295 - PC1) * (-0.2851745656656295 - PC1)) + exp(-1.0 * (0.3763749989194041 - PC2) * (0.3763749989194041 - PC2)) + exp(-1.0 * (0.8094528950374702 - PC3) * (0.8094528950374702 - PC3)) + exp(-1.0 * (0.039483764281568266 - PC4) * (0.039483764281568266 - PC4)) + exp(-1.0 * (0.4706106356598398 - PC5) * (0.4706106356598398 - PC5))), 1.0))- 0.9552631578947368 * (pow((exp(-1.0 * (0.1079375013777528 - PC1) * (0.1079375013777528 - PC1)) + exp(-1.0 * (1.0758838163494822 - PC2) * (1.0758838163494822 - PC2)) + exp(-1.0 * (-1.1857959615458784 - PC3) * (-1.1857959615458784 - PC3)) + exp(-1.0 * (-0.1673426540852874 - PC4) * (-0.1673426540852874 - PC4)) + exp(-1.0 * (0.026132426995135613 - PC5) * (0.026132426995135613 - PC5))), 1.0))- 0.9552631578947369 * (pow((exp(-1.0 * (-0.8567603189085968 - PC1) * (-0.8567603189085968 - PC1)) + exp(-1.0 * (-0.2065525506551554 - PC2) * (-0.2065525506551554 - PC2)) + exp(-1.0 * (-0.6789980433962217 - PC3) * (-0.6789980433962217 - PC3)) + exp(-1.0 * (0.7805257432069781 - PC4) * (0.7805257432069781 - PC4)) + exp(-1.0 * (0.58137923077997 - PC5) * (0.58137923077997 - PC5))), 1.0)) + 1.2964285714285715 * (pow((exp(-1.0 * (-1.5652907673752117 - PC1) * (-1.5652907673752117 - PC1)) + exp(-1.0 * (-0.8359016418916223 - PC2) * (-0.8359016418916223 - PC2)) + exp(-1.0 * (-1.017777993456468 - PC3) * (-1.017777993456468 - PC3)) + exp(-1.0 * (1.9509327153021803 - PC4) * (1.9509327153021803 - PC4)) + exp(-1.0 * (-0.13252562222272146 - PC5) * (-0.13252562222272146 - PC5))), 1.0))- 0.06532142445417022 * (pow((exp(-1.0 *

(1.1385948292031338 - PC1) * (1.1385948292031338 - PC1)) + exp(-1.0 * (0.27622614275326085 - PC2) * (0.27622614275326085 - PC2)) + exp(-1.0 * (-1.0513987281049189 - PC3) * (-1.0513987281049189 - PC3)) + exp(-1.0 * (-0.4469489196549603 - PC4) * (-0.4469489196549603 - PC4)) + exp(-1.0 * (0.5538115528247464 - PC5) * (0.5538115528247464 - PC5))), 1.0))- 0.9552631578947369 * (pow((exp(-1.0 * (0.6465439694433665 - PC1) * (0.6465439694433665 - PC1)) + exp(-1.0 * (-0.9929036586973015 - PC2) * (-0.9929036586973015 - PC2)) + exp(-1.0 * (-0.5882032312576566 - PC3) * (-0.5882032312576566 - PC3)) + exp(-1.0 * (-0.9928659158903361 - PC4) * (-0.9928659158903361 - PC4)) + exp(-1.0 * (-0.48783455954679106 - PC5) * (-0.48783455954679106 - PC5))), 1.0))- 0.9552631578947369 * (pow((exp(-1.0 * (1.4517825283256496 - PC1) * (1.4517825283256496 - PC1)) + exp(-1.0 * (0.5418292645660471 - PC2) * (0.5418292645660471 - PC2)) + exp(-1.0 * (1.5007008415361978 - PC3) * (1.5007008415361978 - PC3)) + exp(-1.0 * (0.03930823178506545 - PC4) * (0.03930823178506545 - PC4)) + exp(-1.0 * (-0.008683332008600053 - PC5) * (-0.008683332008600053 - PC5))), 1.0))+ 1.2964285714285715 * (pow((exp(-1.0 * (0.22574824254622178 - PC1) * (0.22574824254622178 - PC1)) + exp(-1.0 * (-0.6585662026761424 - PC2) * (-0.6585662026761424 - PC2)) + exp(-1.0 * (-0.6929471524300306 - PC3) * (-0.6929471524300306 - PC3)) + exp(-1.0 * (-0.8958199537945392 - PC4) * (-0.8958199537945392 - PC4)) + exp(-1.0 * (-0.9816645255815629 - PC5) * (-0.9816645255815629 - PC5))), 1.0)) - 0.9552631578947369 * (pow((exp(-1.0 * (0.6537523248178476 - PC1) * (0.6537523248178476 - PC1)) + exp(-1.0 * (-1.9950314458590697 - PC2) * (-1.9950314458590697 - PC2)) + exp(-1.0 * (0.6160899306812634 - PC3) * (0.6160899306812634 - PC3)) + exp(-1.0 * (-0.6546021447882548 - PC4) * (-0.6546021447882548 - PC4)) + exp(-1.0 * (-0.9870014981890274 - PC5) * (-0.9870014981890274 - PC5))), 1.0))+ 1.2964285714285715 * (pow((exp(-1.0 * (-8.561881225052373E-4 - PC1) * (-8.561881225052373E-4 - PC1)) + exp(-1.0 * (0.6714910463407581 - PC2) * (0.6714910463407581 - PC2)) + exp(-1.0 * (1.85596515410265 - PC3) * (1.85596515410265 - PC3)) + exp(-1.0 * (-0.8323829416023468 - PC4) * (-0.8323829416023468 - PC4)) + exp(-1.0 * (-0.2642297217346937 - PC5) * (-0.2642297217346937 - PC5))), 1.0))- 0.9552631578947369 * (pow((exp(-1.0 * (1.2870872327957699 - PC1) * (1.2870872327957699 - PC1)) + exp(-1.0 * (-0.7176298204510407 - PC2) * (-0.7176298204510407 - PC2)) + exp(-1.0 * (1.5637092131071746 - PC3) * (1.5637092131071746 - PC3)) + exp(-1.0 * (0.41918135951579966 - PC4) * (0.41918135951579966 - PC4)) + exp(-1.0 * (-0.2171541238408937 - PC5) * (-0.2171541238408937 - PC5))), 1.0))+ 1.2964285714285715 * (pow((exp(-1.0 * (0.2879626400641941 - PC1) * (0.2879626400641941 - PC1)) + exp(-1.0 * (0.7973900475155014 - PC2) * (0.7973900475155014 - PC2)) + exp(-1.0 * (-1.3887646120942885 - PC3) * (-1.3887646120942885 - PC3)) + exp(-1.0 * (-0.11168527347382246 - PC4) * (-0.11168527347382246 - PC4)) + exp(-1.0 * (1.052020672337765 - PC5) * (1.052020672337765 - PC5))), 1.0)) + 1.2964285714285715 * (pow((exp(-1.0 * (-1.261983898361134 - PC1) * (-1.261983898361134 - PC1)) + exp(-1.0 * (-0.10055088975537771 - PC2) * (-0.10055088975537771 - PC2)) + exp(-1.0 * (0.9126521869654228 - PC3) * (0.9126521869654228 - PC3)) + exp(-1.0 * (0.7409711073927905 - PC4) * (0.7409711073927905 - PC4)) + exp(-1.0 * (0.1215848945006455 - PC5) * (0.1215848945006455 - PC5))), 1.0))+ 1.2964285714285715 * (pow((exp(-1.0 * (-0.4755842330651412 - PC1) * (-0.4755842330651412 - PC1)) + exp(-1.0 * (0.2837031722132356 - PC2) * (0.2837031722132356 - PC2)) + exp(-1.0 * (-0.19808748742718618 - PC3) * (-0.19808748742718618 - PC3)) + exp(-1.0 * (-2.236245974343832 - PC4) * (-2.236245974343832 - PC4)) + exp(-1.0 * (2.4376731690228883 - PC5) * (2.4376731690228883 - PC5))), 1.0))+ 1.2964285714285715 * (pow((exp(-1.0 * (-0.21719880581736847 - PC1) * (-0.21719880581736847 - PC1)) + exp(-1.0 * (-1.2210608448644131 - PC2) * (-1.2210608448644131 - PC2)) + exp(-1.0 * (-0.168235268320581 - PC3) * (-0.168235268320581 - PC3)) + exp(-1.0 * (-0.1507108922402918 - PC4) * (-0.1507108922402918 - PC4)) + exp(-1.0 * (-1.3529930054749602 - PC5) * (-1.3529930054749602 - PC5))), 1.0))+ 1.2964285714285715 * (pow((exp(-1.0 * (-0.7186718684550352 - PC1) * (-0.7186718684550352 - PC1)) + exp(-1.0 * (0.29873222577612685 - PC2) * (0.29873222577612685 - PC2)) + exp(-1.0 * (0.4250013915371853 - PC3) * (0.4250013915371853 - PC3)) + exp(-1.0 * (-0.48018033109457026 - PC4) * (-0.48018033109457026 - PC4)) + exp(-1.0 * (-

0.5815213892470266 - PC5) * (-0.5815213892470266 - PC5))), 1.0))- 0.9552631578947368 * (pow((exp(-1.0 * (0.11269198465026975 - PC1) * (0.11269198465026975 - PC1)) + exp(-1.0 * (-0.7492854650339129 - PC2) * (-0.7492854650339129 - PC2)) + exp(-1.0 * (-0.3687733141027855 - PC3) * (-0.3687733141027855 - PC3)) + exp(-1.0 * (-0.8773968082598453 - PC4) * (-0.8773968082598453 - PC4)) + exp(-1.0 * (-0.8299320700509456 - PC5) * (-0.8299320700509456 - PC5))), 1.0))- 0.9552631578947369 * (pow((exp(-1.0 * (-0.3854889203468838 - PC1) * (-0.3854889203468838 - PC1)) + exp(-1.0 * (-0.29246045524449454 - PC2) * (-0.29246045524449454 - PC2)) + exp(-1.0 * (0.3979113173111659 - PC3) * (0.3979113173111659 - PC3)) + exp(-1.0 * (-0.24221861556546842 - PC4) * (-0.24221861556546842 - PC4)) + exp(-1.0 * (-0.6793362223390602 - PC5) * (-0.6793362223390602 - PC5))), 1.0)) - 0.9552631578947368 * (pow((exp(-1.0 * (1.1502216857915553 - PC1) * (1.1502216857915553 - PC1)) + exp(-1.0 * (-1.123369499172101 - PC2) * (-1.123369499172101 - PC2)) + exp(-1.0 * (1.6577688769653032 - PC3) * (1.6577688769653032 - PC3)) + exp(-1.0 * (0.2036884726443394 - PC4) * (0.2036884726443394 - PC4)) + exp(-1.0 * (0.03702708129704269 - PC5) * (0.03702708129704269 - PC5))), 1.0))+ 1.2964285714285715 * (pow((exp(-1.0 * (-0.7843921175302312 - PC1) * (-0.7843921175302312 - PC1)) + exp(-1.0 * (0.7602670946915685 - PC2) * (0.7602670946915685 - PC2)) + exp(-1.0 * (0.3255110693086585 - PC3) * (0.3255110693086585 - PC3)) + exp(-1.0 * (-0.9899756944433088 - PC4) * (-0.9899756944433088 - PC4)) + exp(-1.0 * (-1.2348087489791395 - PC5) * (-1.2348087489791395 - PC5))), 1.0))+ 1.2964285714285715 * (pow((exp(-1.0 * (1.0558078819753354 - PC1) * (1.0558078819753354 - PC1)) + exp(-1.0 * (-0.006072845102713058 - PC2) * (-0.006072845102713058 - PC2)) + exp(-1.0 * (-1.332130116304932 - PC3) * (-1.332130116304932 - PC3)) + exp(-1.0 * (-0.391961813842165 - PC4) * (-0.391961813842165 - PC4)) + exp(-1.0 * (0.6166769436184407 - PC5) * (0.6166769436184407 - PC5))), 1.0)) + 1.2964285714285715 * (pow((exp(-1.0 * (0.8967485009384014 - PC1) * (0.8967485009384014 - PC1)) + exp(-1.0 * (-1.9284444394893696 - PC2) * (-1.9284444394893696 - PC2)) + exp(-1.0 * (-0.7927479236630741 - PC3) * (-0.7927479236630741 - PC3)) + exp(-1.0 * (0.21281727018931376 - PC4) * (0.21281727018931376 - PC4)) + exp(-1.0 * (0.14135561619964995 - PC5) * (0.14135561619964995 - PC5))), 1.0))- 0.9552631578947369 * (pow((exp(-1.0 * (-0.8467452185956604 - PC1) * (-0.8467452185956604 - PC1)) + exp(-1.0 * (0.34306599045574926 - PC2) * (0.34306599045574926 - PC2)) + exp(-1.0 * (-1.3830724263112386 - PC3) * (-1.3830724263112386 - PC3)) + exp(-1.0 * (-0.43273932948922533 - PC4) * (-0.43273932948922533 - PC4)) + exp(-1.0 * (1.7277632662418494 - PC5) * (1.7277632662418494 - PC5))), 1.0))- 0.9552631578947368 * (pow((exp(-1.0 * (0.4059731238672119 - PC1) * (0.4059731238672119 - PC1)) + exp(-1.0 * (0.02210200021595946 - PC2) * (0.02210200021595946 - PC2)) + exp(-1.0 * (-0.6752623687662617 - PC3) * (-0.6752623687662617 - PC3)) + exp(-1.0 * (-0.34113147739545585 - PC4) * (-0.34113147739545585 - PC4)) + exp(-1.0 * (0.16758500361295633 - PC5) * (0.16758500361295633 - PC5))), 1.0))- 0.9552631578947369 * (pow((exp(-1.0 * (-0.2725624660983987 - PC1) * (-0.2725624660983987 - PC1)) + exp(-1.0 * (-0.10189209507332755 - PC2) * (-0.10189209507332755 - PC2)) + exp(-1.0 * (0.30548600812497667 - PC3) * (0.30548600812497667 - PC3)) + exp(-1.0 * (-0.720093493669338 - PC4) * (-0.720093493669338 - PC4)) + exp(-1.0 * (-0.6462028801143773 - PC5) * (-0.6462028801143773 - PC5))), 1.0))- 0.9552631578947369 * (pow((exp(-1.0 * (0.796519083523312 - PC1) * (0.796519083523312 - PC1)) + exp(-1.0 * (-0.9686272829261681 - PC2) * (-0.9686272829261681 - PC2)) + exp(-1.0 * (0.712105329065506 - PC3) * (0.712105329065506 - PC3)) + exp(-1.0 * (-0.8194462607006237 - PC4) * (-0.8194462607006237 - PC4)) + exp(-1.0 * (-0.09973937764419954 - PC5) * (-0.09973937764419954 - PC5))), 1.0))+ 1.2964285714285715 * (pow((exp(-1.0 * (-1.1944556185512665 - PC1) * (-1.1944556185512665 - PC1)) + exp(-1.0 * (0.4971107722000004 - PC2) * (0.4971107722000004 - PC2)) + exp(-1.0 * (1.1515798166202458 - PC3) * (1.1515798166202458 - PC3)) + exp(-1.0 * (-0.4286931846517899 - PC4) * (-0.4286931846517899 - PC4)) + exp(-1.0 * (-0.7723715565084187 - PC5) * (-0.7723715565084187 - PC5))), 1.0)) - 0.9552631578947368 * (pow((exp(-1.0 *

(0.9029139909544912 - PC1) * (0.9029139909544912 - PC1)) + exp(-1.0 * (-1.276470307387063 - PC2) * (-1.276470307387063 - PC2)) + exp(-1.0 * (1.136718953620475 - PC3) * (1.136718953620475 - PC3)) + exp(-1.0 * (-0.2711877354552425 - PC4) * (-0.2711877354552425 - PC4)) + exp(-1.0 * (-0.32797667493727317 - PC5) * (-0.32797667493727317 - PC5))), 1.0))+ 1.2964285714285715 * (pow((exp(-1.0 * (0.24891494823572066 - PC1) * (0.24891494823572066 - PC1)) + exp(-1.0 * (-0.7969793468633649 - PC2) * (-0.7969793468633649 - PC2)) + exp(-1.0 * (-0.4758609622796306 - PC3) * (-0.4758609622796306 - PC3)) + exp(-1.0 * (-0.6257646906283095 - PC4) * (-0.6257646906283095 - PC4)) + exp(-1.0 * (-0.16972609196640884 - PC5) * (-0.16972609196640884 - PC5))), 1.0))- 0.9552631578947369 * (pow((exp(-1.0 * (-0.7697872501578359 - PC1) * (-0.7697872501578359 - PC1)) + exp(-1.0 * (1.0373149842379308 - PC2) * (1.0373149842379308 - PC2)) + exp(-1.0 * (-0.9743468612773878 - PC3) * (-0.9743468612773878 - PC3)) + exp(-1.0 * (0.38426637071967534 - PC4) * (0.38426637071967534 - PC4)) + exp(-1.0 * (1.0961520584522266 - PC5) * (1.0961520584522266 - PC5))), 1.0))+ 1.2964285714285715 * (pow((exp(-1.0 * (-1.5515915629082544 - PC1) * (-1.5515915629082544 - PC1)) + exp(-1.0 * (-0.5979067829675345 - PC2) * (-0.5979067829675345 - PC2)) + exp(-1.0 * (0.8427765701029908 - PC3) * (0.8427765701029908 - PC3)) + exp(-1.0 * (0.4540689877224843 - PC4) * (0.4540689877224843 - PC4)) + exp(-1.0 * (0.9481086675737915 - PC5) * (0.9481086675737915 - PC5))), 1.0))+ 0.3585930005491796

... (5.35)

where $\widehat{FT}$ refer to the normalized values of *FT*; the magnitudes of *PCs* in equation (5.35) are calculated using the equations (5.25) to (5.29).

REFERENCES

Akiyama K., Pak H., Ueki Y., Yoshiie R., Naruse I. (2011). Effect of MgO based addition to upgraded brown coal on the ash deposition behavior during combustion, *Fuel*, **90(11)**, pp.3230-3236.

Bahrin D., Faizal M., Ibrahim E., Adiarso, Priambodo T.B., Darmawan, Iman (2009). Predicting coal ash fusion temperature of South Sumatera’s blended coal with coal blending simulation and laboratory analysis, SISEST 2009 - RUSNAS PEBT A IX – 1.

Bryant G.W., Browning G.J., Emanuel H., Gupta S. K., Gupta R. P., Lucas J. A., Wall T. F. (2000). The fusibility of blended coal ash, *Energy Fuels*, **14(2)**, pp.316-325.

CEA Report of the group for studying range of blending of imported coal with domestic coal (2012). *Central Electricity Authority (CEA)*, April 2012. http://cea.nic.in/reports/others/thermal/tetd/blending_coal.pdf

Hanxu L., Ninomiya Y., Zhongbing D., Mingxu Z. (2006). Application of the FactSage to predict the ash melting behavior in reducing conditions, *Chinese J. Chem. Eng.*, **14(6)**, pp. 784–789.

Kim H.T., Chung S.W., Lee S.H. (1994). Slagging behavior of coal ash under gasifier environment (1994), *American Chemical Society (ACS) national meeting*, Washington, DC (United States), 21-26 Aug 1994, pp.894-898.
https://web.anl.gov/PCS/acsfuel/preprint%20archive/Files/39_3_WASHINGTON%20DC_08-94_0894.pdf

Kim M.R., Jang J.G., Lee S.K., Hwang B.Y., Lee J.K. (2010). Correlation between the ash composition and melting temperature of waste incineration residue, *Korean J. Chem. Eng.*, **27(3)**, pp. 1028-1034.

Lolja S.A., Haxhi H., Dhimitri R., Drushku S., Malja A. (2002). Correlation between ash fusion temperatures and chemical composition in Albanian coal ashes, *Fuel*, **81(17)**, pp. 2257–2261.

Matuszewski M. (2012). *Quality guidelines for energy system studies-Detailed coal specifications*, DOE/NETL-401/012111, Final Report, January 2012.

Moreno R.X. (2008). *Development of a LIBS system for prediction of ash fusion temperature in coal-fired power plants*, Theses and Dissertations (2008), Paper 1019. https://preserve.lehigh.edu/etd/1019.

Qiu J.R., Li F., Zheng Y., Zheng C.G., Zhou H.C. (1999). The influences of mineral behaviour on blended coal ash fusion characteristics, *Fuel*, **78(8)**, pp. 963–969.

Rees O.W. (1964). *Composition of the Ash of Illinois Coals*, Illinois State Geological Survey, Urbana, Circular 365, URL: http://library.isgs.illinois.edu/Pubs/pdfs/circulars/c365.pdf.

Steyn M, Minnitt RCA. (2010) Thermal coal products in South Africa. Journal of The Southern African Institute of Mining and Metallurgy VOLUME 110, OCTOBER 2010.

U.S.DOE Award Number DE-FC 26-00NT- 40693 (2001) Quarterly Technical Progress Report, p.2-7. https://www.osti.gov/servlets/purl/788765

Vorres K.S. (1979). Chemistry of coal ash melting in gasification and combustion, *Structure of Coal Symposium*, Fuel Chemistry Division, ACS/CSJ Congress, Honolulu, April 1979, pp.247-254, Institute of Gas Technology, Chicago. https://web.anl.gov/PCS/acsfuel/preprint%20archive/Files/24_2_HONOLULU_04-79_0247.pdf

CHAPTER 6

PREDICTION OF INITIAL DEFORMATION TEMPERATURE OF INDIAN COALS USING COMPUTATIONAL INTELLIGENCE BASED MODELS

Abstract

The property that governs the deposition of ash in various coal-utilizing process equipment is termed *ash fusion temperature* (AFT). The physico-chemical phenomena underlying the said ash deposition that adversely affects the efficiency of coal-based processes are termed *slagging* and *fouling*. One of the four commonly used AFTs in practice is termed *Initial deformation temperature* (*IDT*). It is that temperature at which the rounding of the tip of an ash cone is noted. It's of particular concern since it represents the temperature where the ash first softens and therefore becomes sticky. In India, the currently mined coals have a high ash content (25-45%) and therefore are of low quality. These coals are predominantly used in the thermal power stations located widely India. Owing to the its importance in plant operation, it is necessary that a model with a good quality *IDT* prediction accuracy and generalization ability is available for design engineers and plant personnel. The best *IDT* prediction model developed for in Chapter 5 is found to be unsuitable for high ash Indian coals. Accordingly, computational intelligence (CI) based *IDT* prediction models have been developed in this study wherein the contents of the seven principal oxides found in coal ashes (i.e. SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO, TiO_2, Na_2O+K_2O), were used as the predictors. The experimental data on Indian coals were sourced from *Central Institute of Mining and Fuel Research* (CIMFR), Dhanbad, India. Specifically, three CI-based modeling formalisms, namely MLPNN, SVR and GP were employed to develop the *IDT* prediction models. Among these models, the *IDT* prediction accuracy and generalization performance of the SVR-based model was found to be superior to the corresponding GP and MLPNN-based models.

6.1 INTRODUCTION

Coal is an important source for energy generation. As coal is combusted or gasified its mineral matter, typically in the form of oxides, is left as an incombustible deposit (residue), called *ash*. The nature and quantity of ash formed during combustion and gasification of coal is a significant factor influencing its utility, removal and storage, heat transfer processes, process efficiency, and operating and maintenance costs (Lolja et al., 2002).

Ash fusion temperatures (AFTs) represent important properties indicating the suitability of a coal source for combustion, gasification and liquefaction purposes. The melting characteristics and fusion temperatures of the coal ash depends on the elemental and mineral compositions (Liu et al., 2006; Thompson and Argent, 1999). There are two major phenomena namely *slagging* or *fouling* arising essentially from the deposition of ash in the operation of coal-based processes. These are responsible for the occurrence of ash clinkering and ash agglomeration on the heat absorbing surfaces of the exposed equipment (such as heat exchangers) of the coal utilizing processes. It is well-established that ash clinkering may result into an unstable gasifier operation along with channel burning and pressure drop (Van Dyk et al., 2001). The major undesired effects of the slagging and fouling are frequent shut-downs and reduced process efficiency. A detailed discussion of slagging and fouling is already provided in Chapter 5 (section 5.1). AFTs are good cross-line parameters (indicators) of the occurrence of slagging and fouling (Dyk et al., 2001; Seggiani, 1999) in the process operation. Therefore, it becomes important to predict the AFTs for designing, operating and optimizing coal-based processes.

Among the four AFTs, namely *Initial Deformation Temperature* (IDT), *Hemispherical Temperature* (HT), *Softening Temperature* (ST) and *Fluid Temperature* (FT), the first one is of special significance since it decides the type of coal and fusion range (difference between *IDT* and *FT*) (Lolja et al., 2002; Ministry of coal, 2008). The *IDT* refers to the temperature at which the rounding of the tip of an ash cone is noted. It’s of particular concern since it represents the temperature where the ash first softens and therefore becomes sticky (Gupta et al. 1998).

The knowledge of *IDT* is useful in case of gasifiers using non-coking coals (Sharma et al., 2014). These coals have a tendency to form *Char* (Sengar, 2015); it is a high ash containing residue that remains after light gases (for example, coal gas), and tar have been driven out or released from a carbonaceous material during the initial stage of combustion, which is known as *carbonization* (Kumar and Gupta, 1994; Ministry of coal, 2008). The non-coking coals, when subjected to gasification causes an uneven temperature distribution in reactor, and thereby transient pressure drop levels, resulting in an unstable gasifier operation and hence variable efficiencies in each operation (Dyk et al., 2001). Due to the high ash content in a non-coking coal, it does not possess any caking properties and as a result the *IDT* measurement are easier as compared to other ash fusion temperatures (AFTs) (Sengar, 2015). Another significant feature of *IDT* is that it shows a definite increasing or decreasing trend in coal carbonization processes. These trends are often used as predictive indicators of the occurrence of slagging, fouling, and abrasion during coal combustion. The increase or decrease of *IDT* is influenced by the variations in the weight percentages of metal oxides such as Fe_2O_3, Al_2O_3, SiO_2, and CaO (Sharma A., 2014). Also, in the studies of blending of lignite coal ashes with metal oxides, *IDT* has been observed as the prime testing temperature among all AFTs (Acma et al., 2010; Mamoru et al., 1977). The reason of its prime importance is that the measurement of *IDT* takes lesser time, cost and energy as compared to all other AFTs.

In India, coal-based thermal power stations generate close to 76.5% of the total energy produced in the country (https://en.wikipedia.org/wiki/Electricity_sector_in _India). The currently mined coals in India are non-coking (Ghosh and Chatterjee, 2012) and contain a high percentage of ash that varies between 25 and 45%. Thus, their heat capacity is low. These coals are also used in industries such as cement, fertilizer, glass, ceramic, paper, chemical and brick manufacturing (Kumar and Patel, 2008; Ministry of coal, 2008, Singh, 1997). The comparatively higher ash content in Indian coals is due to the *drift* theory of formation of coal deposits (Singh, 1997). The coal seams formed according to the drift theory possess higher ash content than those which follow the *in-situ* theory of formation. The usage of high-ash coals—since it affects the efficiency of the thermal power stations and therefore has tremendous cost implications for the manufacturer and the users—necessitates that a model possessing a good prediction ability and generalization performance is available for the prediction of their *IDT*.

The Indian coals have been studied by several researchers (Mishra and Ghosh, 1996; Singh and Singh, 2000; Singh et al., 2013). These studies shows the lack of information on the fusibility of coals ash during combustion. The *IDT* regression models have been developed by Sharma et al., 2014 for limited 37 coals samples collected from Meghalaya basin. Recently, the thermodynamic models of phase equilibria have been developed by Chakravarty et al., 2015. The limitations in the applicability of these models are that these models are less parsimonious, limited to a small sample range, and therefore difficult to deploy in practice.

In Chapter 5, CI-based models have been developed for the prediction of all four AFTs, namely *Initial Deformation Temperature* (IDT), *Hemispherical Temperature* (HT), *Softening Temperature* (ST) and *Fluid Temperature* (FT) using data of coals from several countries. An exercise was performed to assess the applicability of the best performing MLPNN model for the *IDT* prediction developed in Chapter 5 to the high ash Indian coals. This test was conducted using the oxides and the corresponding *IDT* data of 91 number of high ash Indian coals listed in Table 6.A APPENDIX 6. For constructing and examining the generalization ability of these models, the experimental data set for *IDT* was randomly partitioned in 63:27:10 ratio into *training*, *test*, and *validation* sets. The results of the stated test are given in **Table 6.1**

Table 6.1: Comparison of MLPNN-model predicted *IDT* developed for Global coal with high ash Indian coals

Dataset#	**Comparison values of MLPNN-model predicted *IDT* with high ash Indian coals**			**MLPNN-model predicted *IDT* for Global coal (Chapter 5)**		
	CC	*RMSE*	*MAPE* (%)	*CC*	*RMSE*	*MAPE* (%)
Training	0.59	36.83	2.39	0.87	53.22	3.39
Test	0.70	34.16	2.24	0.82	77.93	4.73
Validation	0.83	28.97	2.01	0.76	38.78	2.66

# The number of patterns in the *training, test* and *validation* sets have been 57,25, and 9, respectively.

As can be seen from the above table, the MLPNN model is making poor *IDT* predictions for the Indian high ash coals. The "not-so-good" *IDT* prediction results obtained using the MLPNN model trained on the global coal data has prompted development of *IDT* prediction models exclusively for the high ash Indian coals. Accordingly, in this Chapter presents results of the development *IDT* models using three computational intelligence (CI) data-driven modeling formalisms viz. *Artificial Neural Networks* (ANN), *Support Vector Regression* (SVR) and *Genetic Programming* (GP). The reasons for choosing these computational intelligence methods are already explained in Chapter 5 (section 5.4) and the details of their origin, implementation procedures and applications are provided in sections 2.3.1, 2.2.2 and 2.3.3 in Chapter 2. Owing to the strong dependence of *IDT* on the oxide content in the coal ashes (Liu et al., 2006; Yin et al., 1998), the weight percentages (wt%) of eight oxides, namely, SiO_2, Al_2O_3, Fe_2O_3, CaO, MgO, TiO_2 and K_2O+Na_2O were used as the inputs of the three types of models developed in this work.

6.2 DATA

The experimental data of the mineral composition of 91 number of Indian coal ash samples along with the corresponding *IDT* [^{0}C] values were sourced from *Central Institute of Mining and Fuel Research* (CIMFR), Dhanbad, India. These are tabulated in Appendix 6 Table 6.A. The average wt% of ash content in the Indian coals vary in between the range, ~15 to ~55 (Choudhury and Bhaktavatsalam, 1995). The wt% of the ash covered in the present data ranges from ~0.06 to ~70.55.

6.3 RESULTS AND DISCUSSION

6.3.1 Principal Component Analysis (PCA)

To get rid of the correlated inputs (predictors) and thereby the data redundancy, and an unnecessary raise in the computational load, the input space of the CI-based models consisting of the weight percentages of the seven oxides ($x_1 - x_7$) listed in the Table in Appendix 6, were subjected to the *principal component analysis* (PCA) (Geladi and Kowalski, 1986). The purpose of PCA is to convert a set of linearly correlated variables into a set of linearly uncorrelated new variables (termed principal components) using orthogonal transformations. Next, only first few principle components (*PC*s)—which are lesser than the number of original variables—capturing maximum variance in the data are chosen as predictors for developing the modeling thus bringing about a reduction in the dimensionality of the model's input space. In this work, seven *PC*s were extracted from the weight percentage values of the seven oxides listed in Table 6.A in Appendix 6. Prior to the PCA, the wt% values of the seven metal oxides namely SiO_2 (x_1), Al_2O_3 (x_2), Fe_2O_3 (x_3), CaO (x_4), MgO (x_5), TiO_2 (x_6), and K_2O+Na_2O (x_7) were normalized using "*Z-score*" technique using equation (5.5). Similarly, the *IDT* values were normalized using equation (5.6). The mean and standard deviation values in conducting the normalization of the seven inputs (x_1 to x_7) and the *IDT* values are listed in Table 6.B in APPENDIX 6. The magnitude of the variance captured by each PC is given in **Table 6.2**. It is seen in Table 6.2 that a large percentage (≈ 92%) of variance in the oxide data pertaining to the high ash Indian coals has been captured by only the first four *PC*s. This result shows that in CI-based modeling only first four PCs can be considered in place of the original seven inputs.

Table 6.2: Percentage of variance (%) in the *IDT* data captured by seven PCs

Data Set	Variance (%)						
	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6	PC_7
High Ash Indian coals	34.7	27.6	18.6	11.1	6.9	1.0	0

The expressions of the first four PCs are given below

$$PC_1 = 0.457\hat{x}_1 + 0.134\hat{x}_2 - 0.403\hat{x}_3 - 0.152\hat{x}_4 - 0.539\hat{x}_5 - 0.538\hat{x}_6 + 0.088\hat{x}_7 \tag{6.1}$$

$$PC_2 = -0.405\hat{x}_1 + 0.515\hat{x}_2 + 0.168\hat{x}_3 + 0.491\hat{x}_4 - 0.289\hat{x}_5 - 0.254\hat{x}_6 - 0.391\hat{x}_7 \tag{6.2}$$

$$PC_3 = 0.071\hat{x}_1 - 0.526\hat{x}_2 + 0.55\hat{x}_3 + 0.393\hat{x}_4 - 0.23\hat{x}_5 - 0.309\hat{x}_6 + 0.336\hat{x}_7 \tag{6.3}$$

$$PC_4 = 0.402\hat{x}_1 - 0.314\hat{x}_2 + 0.209\hat{x}_3 - 0.077\hat{x}_4 - 0.014\hat{x}_5 + 0.008\hat{x}_6 - 0.831\hat{x}_7 \tag{6.4}$$

where $\hat{x}_r$; r = 1,2,...,7 denote the normalized inputs (evaluated using Eq. 5.5 given in Chapter 5) and the mean and standard deviation values given in Table 6.B in Appendix 6.

The above-stated four PCs were used as the inputs in developing the GP-, MLPNN-, and SVR-based *IDT* prediction models. For constructing and examining the generalization ability of these models, the experimental data set consisting of the PCA-transformed inputs and the related *IDT* values was arbitrarily partitioned in 63:27:10 ratio into *training*, *test*, and *validation* sets. Here the first set was used for training the CI-based models; the test and the validation sets were used for testing and validating the generalization ability of the models.

6.3.2 GP-based modeling

The GP-based *IDT* prediction model was built using *Eureqa Formulize* software package (Schmidt and Lipson, 2009). The GP procedure followed by this package is adapted to search and optimize models possessing low complexity and, thus, endowed with the much-desired generalization ability. The best converged solution provided by the GP is influenced by multiple factors that includes the size of the training set, tree depth, choice/selection of the mathematical operators, and input normalization schemes. To obtain a model with a good quality *IDT* prediction accuracy and generalization ability, a number of GP runs were performed by varying the stated influential attributes rigorously and systematically. The best converged solution (i.e., the one possessing the maximum fitness in the converged population of candidate solutions) obtained in each run was recorded and the overall best solution was selected on the basis of following criteria (Sharma and Tambe, 2014): (a) high and comparable values of correlation coefficients (CCs) with respect to the *IDT* predictions of the training, test and validation set data, and (b) small and comparable values of *RMSE,* and *MAPE*, pertaining to the model predictions in respect of the training, test, and validation set data. The GP-based overall best model satisfying the stated conditions for the prediction of *IDT* of the high ash Indian coals is given below wherein $\widehat{IDT}$, refers to the normalized value of *IDT*.

$$\widehat{IDT} = 0.270PC_1 + \ 0.228PC_2 + \left(\frac{0.049PC_2PC_3 - 0.025}{PC_4 - 0.143 - 0.242PC_3 - 0.244PC_1^3}\right)$$

$$-0.321PC_3 - 0.030PC_2PC_3 \qquad (6.5)$$

As can be seen, the GP-based model (Eq. 6.5) has a nonlinear form. This is similar to the generalized *IDT* model for coals from multiple countries developed in Chapter 5 (Eq. 5.7). It may also be noted that Eq. 6.5 contains all the for PCs although the GP strategy has the ability to ignore the inputs that are not influential in correctly predicting the output (i.e. *IDT*). The weight percentages of all the principal oxides in the ashes of Indian coals are represented in the four PCs (see Eqs. 6.1 to 6.4); these results suggest that nonlinear relationships are better suited than the linear relationships for predicting *IDT* magnitudes of the high ash Indian coals using the weight percentages of metal oxides present in ash. The *CC*, *RMSE* and *MAPE* values in respect of the *IDT* predictions made by the GP-based model for the training, test, and validation datasets are listed in **Table 6.3**.

Table 6.3: Results of the statistical analysis pertaining to the prediction and generalization performance of the GP-, MLP-, and SVR-based models predicting magnitudes of *IDT*

Dataset[#]	**Models**								
	GP			**MLPNN**			**SVR**		
	CC	***RMSE***	***MAPE (%)***	***CC***	***RMSE***	***MAPE (%)***	***CC***	***RMSE***	***MAPE (%)***
Training	0.812	11.31	1.75	0.894	10.37	1.62	0.997	1.47	0.24
Test	0.835	10.63	1.74	0.810	12.64	2.01	0.913	7.92	0.76
Validation	0.950	8.20	1.38	0.898	14.10	2.14	0.959	8.43	1.22

# The number of patterns in the *training*, *test* and *validation* sets have been 57, 25, and 9, respectively.

6.3.3 MLPNN- based models

The MLPNN-based *IDT* model were trained using *error-back-propagation* (EBP) algorithm (Rumelhart et al., 1986) available in *RapidMiner* data-mining suite (Mierswa et al., 2006; RapidMiner, 2007). The inputs for the MLPNN were the four principal components defined in Eqs. 6.1 to 6.4. Following structural parameters of the MLPNN and the EBP algorithm-specific parameters were changed methodically to attain the best model having a high *IDT* prediction accuracy and generalization and validation performance: number of hidden layers, number of nodes in each hidden layer, *learning rate* (η), *momentum coefficient* (μ), and initial distribution of MLPNN weights. **Table 6.4** gives the details of the MLPNN structure, type of the transfer functions used at the hidden and output layer, and EBP-specific parameter values (η, μ), that resulted in the optimal model. The *CC*, *RMSE* and *MAPE* values in respect of the *IDT* predictions made by the optimal MLPNN model are given in Table 6.3.

Table 6.4: Details of the MLPNN-based optimal model for the prediction of *IDT*

Number of nodes/neurons in input layer (*L*)	**Number of nodes/neurons in the hidden layer (*M*)**	**Error Back Propagation Algorithm Parameter**		**Transfer function for hidden nodes**	**Transfer function at output node**
		Learning rate (η)	**Momentum (μ)**		
7	6	0.333	0.08105	Logistic Sigmoid	Identity

6.3.4 SVR-based model

Likewise, the MLPNN, the SVR-based *IDT* prediction model was constructed using *RapidMiner* data-mining suite (Mierswa et al., 2006; RapidMiner, 2007) wherein the widely used ε-SVR algorithm was employed; the kernel function employed was *ANOVA*. In SVR algorithm, the inputs (predictors) are first mapped nonlinearly in to a high-dimensional feature input space, wherein they are mapped linearly with the outputs. A detailed account of the origin, theory and applications of SVR are provided in Chapter 2, section 2.3.2. The ε-SVR training algorithm (Vapnik, 1995) and ANOVA kernel function employ four parameters namely *regularization constant* (*C*), *kernel gamma* (γ), *kernel degree* and, *radius of the tube* (ε), which were varied systematically and rigorously to reach an optimal SVR model having high *IDT* prediction and generalization performance. **Table 6.5** gives the values of the above mentioned four ε-SVR and ANOVA parameters as also the number of support vectors that yielded the optimal SVR model. The *CC*, *RMSE,* and *MAPE* magnitudes pertaining to the SVR model predictions of *IDT* for training, test, and validation set data are listed in Table 6.3. The SVR model pertaining to the *IDT* prediction is presented in APPENDIX 6, SVR Model 6. C.

Table 6.5: Details of the SVR-based model predicting *IDT*

Kernel gamma (γ)	**Tube radius (ε)**	**Regularization Parameter (C)**	**Kernel Degree (δ)**	**Number of support vectors**	**Kernel function**
0.836	0.00025	1	4.5	53	ANOVA

6.3.5 Comparison of CI-based models for *IDT* prediction

In Table 6.3, it is noticed that the correlation coefficient (*CC*) magnitudes for the *IDT* predictions prepared by the SVR-based model for the training, test and validation data are high and fall in the (~0.91, ~0.99) range. The stated values are also higher than the corresponding *CC* values in the case of MLPNN (range: 0.81 - 0.89) and GP (range: 0.81 - 0.95) models. Moreover, the *RMSE* (range: 1.47 – 8.43) and *MAPE* values (range: 0.24 – 1.22) regarding the predictions by the SVR-based model are also lower than those for the MLPNN and GP based models. Thus, all the three statistical measures (*CC*, *RMSE*, and *MAPE*) regarding the *IDT* predictions clearly signify that the SVR model have an excellent prediction and generalization performance and it has outperformed the MLPNN and GP-based models.

A graphical comparison of the performance of the three CI-based models has also been made. Figure 6.1 depicts three parity plots wherein the GP-, MLPNN-, and SVR-model predicted *IDT* values are plotted against the corresponding experimental *IDT* values. In panels (a) and (b) of the figure it is seen that there exists a reasonably good match between the experimental and model-predicted *IDT* values. However, in panel (c) of the figure, which corresponds to the predictions by the SVR model, it is noticed that the match between the model-predicted and experimental *IDT* values is the closest. This observation further supports the earlier inference (drawn on the basis of the *CC*, *RMSE* and *MAPE* values) that SVR model has outperformed the MLPNN and GP models.

In general, GP based models are simple to know and deploy in a practical setting. It can be seen in Table 6.3 that the *CC* values corresponding to the *IDT* predictions by the GP model are adequately high (range: ~0.81 to ~0.95) although these are not as high as that for the SVR model. Similarly, the corresponding *RMSE* and *MAPE* magnitudes are reasonably low but still are higher than that for the SVR model predictions. These results suggest that the GP-based model though not the best CI-based model for predicting *IDT* of high ash Indian coals, can still be a potential candidate in situations where the ease of usage along with a reasonable prediction and generalization performance is a primary criterion for the model deployment.

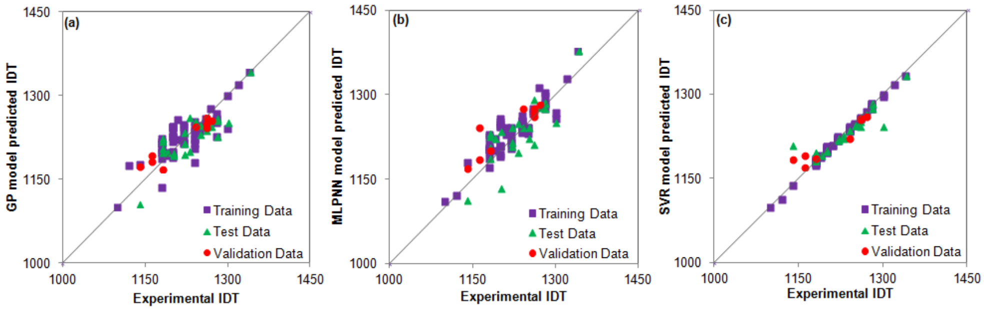


Figure 6.1: Plots pertaining to the experimental *IDT* values vis-à-vis those predicted by the GP (panel *a*), MLP (panel *b*), and SVR based (panel *c*) models.

6.4 CONCLUSION

The coals presently mined In India, contain high amount (25 – 45%) of ash. These low-quality coals are used predominantly in thermal power stations which produce close to 76 % energy generated in the country. In India coals are also used in cement, paper, steel, brick and other industries. The use of high ash coals aggravates the commonly experienced problem of slagging and fouling caused by the deposition of coal ash in the equipment (such as boilers and heat exchangers) of coal-utilizing industries. The occurrence of these phenomena adversely affects the process efficiency and thereby economics. *Initial deformation of temperature* (*IDT*) is one of the member of a group of temperatures termed *ash fusion temperature* (AFTs), which govern the deposition of ash in coal-based process equipment. AFT represents the temperature when the ash first softens and therefore becomes sticky. As one of the important indicators of slagging and fouling, *IDT* has certain advantages namely (a) its measurement is easier as compared to the AFTs, and (b) it exhibits a definite increasing or decreasing trend which greatly assists in predicting the occurrence of slagging, fouling, and abrasion during the process operation. Owing to the importance of *IDT* in the design and operation of coal-based processes, it was necessary that a mathematical model with a good quality *IDT* prediction accuracy and generalization ability is available for design engineers and plant personnel. In Chapter 5, such a model has been developed using multi-layer neural networks (MLPNN) using the ash composition data of coals from various countries. However, this model made poor *IDT* predictions for the high ash Indian coals. To address the above-stated difficulty, in this study models have been developed using three computational intelligence (CI) based data-driven formalisms, namely, genetic programming (GP), multi-layer perceptron neural network (MLPNN), and support vector regression (SVR) for the prediction of *IDT* of high ash Indian coals. A rigorous comparison of performance of the three competing models shows that the SVR model possesses the best *IDT* prediction accuracy and generalization potential. The noteworthy feature of the SVR model presented here is that it can be used by the power stations and other industries using high ash Indian coals for accurately predicting the *IDT* magnitudes. The knowledge of the *IDT* greatly assists in getting forewarned about the occurrence of slagging and fouling in the process equipment, which result in the reduced process efficiency.

6.5 REFERENCES

Acma H.H., Yaman S., Kucukbayrak S. (2010). Effect of biomass on temperatures of sintering and initial deformation of lignite ash, *Fuel*, **80(10)**, pp. 3063- 3068.

Chakravarty S., Mohanty A., Banerjee A., Tripathy R., Mandal G.K., Basariya M.R., Sharma M., (2015). Composition, mineral matter characteristics and ash fusion behavior of some Indian coals, *Fuel*, **150**, pp. 96–101.

Choudhury R., Bhaktavatsalam A.K. (1995). Benification of Indian Coal by Chemical Technique, *Energy Converse anagement*, **38(2)**, pp. 173-178.

Dyk J.C., Keyser M.J., Zyl J.W. (2001). Suitability of feedstocks for the Sasol–Lurgi fixed bed dry bottom gasification process, *GTC Conference*, San Francisco, USA.

Geladi P., Kowalski B.R. (1986). Partial least squares regression (PLS): A tutorial, *Anal Chim Acta*, **85**, pp. 1–17.

Ghosh A., Chatterjee A. (2012). *Iron making and steel making theory and practice*, New Delhi, PHI Learning Private Limited.

Gupta, S.K., Wall, T.F., Creelman, R.A., Gupta, R.P. (1998) Ash fusion temperatures and the transformations of coal ash particles to slag, *Fuel Processing Technology*, 56(1-2), pp. 33-43.

Kumar M., Gupta R.C. (1994). Carbonization study of Dhanbad non-coking coal, *Trans. Indian Inst.Met.*, **47**, pp.103-109.

Kumar M., Patel S.K. (2008). Characteristics of Indian non-coking coals and iron ore reduction by their Chars for directly reduced iron production, *Mineral processing and extractive metallurgy review*, **29**, pp. 258-273.

Liu Y.P., Wu M.G., Qian J.X. (2006). Predicting coal ash fusion temperature based on its chemical composition using ACO-BP neural network, *Thermochim. Acta*, **454(1)**, pp. 64–68.

Lolja S.A., Haxhi H., Dhimitri R., Drushku S., Malja A. (2002). Correlation between ash fusion temperatures and chemical composition in Albanian coal ashes, *Fuel*, **81(17)**, pp. 2257–2261.

Mamoru K., Mahajan O.P, Walker P.L. (1977). Effect of carbon deposition on porosity and reactivity of a lignite char, *Fuel*, **56**, pp. 444-450.

Mierswa I., Wurst M., Klinkenberg R., Scholz M., Euler T. (2006). Rapid prototyping for complex data mining tasks, In: *Proceeding of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*, pp.935–940. Doi:10.1145/1150402.1150531.

Ministry of Coal – Government of India (2008), *COAL DIRECTORY OF INDIA 2007 – 08 (Part-I: Coal Statistics),* Kolkata. http://www.coalcontroller.gov.in/writereaddata/files/download/coaldirectory/Coal%20Directory%20of%20India%202007-08.pdf

Mishra H. K., Ghosh R. K. (1996). Geology, petrology and utilization potential of some tertiary coals of northeastern region of India, *Int. J. Coal Geol.*, **30**, pp. 65–100.

RapidMiner, 2007. https://rapidminer.com/products/studio.

Rumelhart D. E., Hinton G. E., Williams R.J. (1986). Learning representations by back-propagating error, *Nature*, **323(6)**, pp. 533- 536.

Schmidt M., Lipson H. (2009). Distilling free-form natural laws from experimental data, *Science*, **324(5923)**, pp. 81–85.

Seggiani M. (1999). Empirical correlations of the ash fusion temperatures and temperature of critical viscosity for coal and biomass ashes, *Fuel*, **78 (9)**, pp. 1121–1125.

Sengar V.S. (2015). *Carbonization study of non-coking coals and characterization of their properties for application in DRI production*, Thesis, Department of Metallurgical and Materials Engineering, National Institute of Technology, Rourkela.

Sharma A., Saikia A., Khare P., Dutta D.K., Baruah B. P. (2014). The chemical composition of tertiary Indian coal ash and its combustion behaviour – a statistical approach- Part 2, *Journal of earth system science*, **123**, pp.1439-1449.

Sharma S., Tambe S.S. (2014). Soft-sensor development for biochemical systems using genetic programming, *Biochemical Engineering Journal*, **85**, pp. 80-100.

Singh R. D. (1997). *Principles and Practices of Modern Coal Mining (1997),* New Age International, New Delhi.

Singh M. P., Singh A. K. (2000). Petrographic characteristics and depositional conditions of Eocene coals of platform basins, Meghalaya, India, *Int. J. Coal Geol.,* **42(4),** pp. 315–356.

Singh A. K., Singh M. P., Singh P. K. (2013). Petrological investigations of Oligocene coals from foreland basin of northeast India, *Energy, Exploration and Exploitation*, **31(6)**, pp. 909–936.

Thompson D., Argent B.B. (1999). Prediction of coal ash composition as a function of feedstock composition, *Fuel*, **78(5)**, pp. 539–548.

Van Dyk J.C., Keyser M.J., Van Zyl J.W. (2001). Suitability of feed-stocks for the Sasol–Lurgi fixed bed dry bottom gasification process, In: *Gasification Technology Conference*, Gasification Technologies Council, Arlington, VA, USA, pages 11.

Vapnik V.N. (1995). *The Nature of Statistical Learning Theory,* Springer Verlag: New York. ISBN 9781475732641.

Yin C., Luo Z., Ni M., Cen K. (1998). Predicting coal ash fusion temperature with a back-propagation neural network model, *Fuel*, **77(15)**, pp. 1777-1782.

APPENDIX 6

Table 6.A: Experimental data consisting of the oxide composition (wt %) and the corresponding magnitudes of *Initial Deformation Temperature* (IDT)

S. No.	Model inputs (wt%)							Model Output
	SiO_2	Al_2O_3	Fe_2O_3	TiO_2	CaO	MgO	K_2O+Na_2O	IDT
	(x_1)	(x_2)	(x_3)	(x_4)	(x_5)	(x_6)	(x_7)	
1	59.02	31.58	4.63	2.02	0.87	0.45	1.38	1270
2	59.57	28.74	6.42	1.64	1.50	0.44	1.64	1250[$]
3	58.72	23.65	8.92	1.13	2.93	1.28	3.28	1120[^]
4	62.03	24.24	6.45	1.13	2.62	0.80	2.64	1200
5	61.68	26.18	5.63	1.61	1.33	0.45	3.07	1220
6	61.67	27.04	4.87	1.46	1.82	0.72	2.34	1200
7	55.75	26.83	9.38	1.96	2.89	1.15	1.94	1180[^]
8	66.07	24.92	4.02	1.12	1.63	0.57	1.61	1220
9	58.87	26.89	7.62	1.77	1.86	0.65	2.31	1180[^]
10	51.47	34.20	5.62	1.60	4.23	1.32	1.50	1240
11	63.49	28.23	4.01	1.44	0.87	0.31	1.57	1210
12	58.52	31.12	6.01	2.00	0.69	0.20	1.42	1280
13	62.26	28.65	2.41	1.61	1.44	0.40	3.15	1220
14	60.88	27.81	6.83	2.09	0.23	0.12	1.99	1260
15	62.17	27.27	5.49	2.52	0.70	0.32	1.44	1220
16	59.87	22.68	10.2	2.08	2.40	1.22	1.44	1240[^]
17	62.21	27.06	6.47	1.70	1.16	0.25	1.09	1240[$]
18	66.75	26.76	2.22	1.53	0.43	0.32	1.94	1280
19	61.14	29.56	4.83	1.21	1.46	0.35	1.39	1340
20	66.45	22.79	6.35	1.18	0.92	0.16	1.71	1320[$]
21	60.34	25.63	9.63	1.68	0.96	0.48	1.21	1220
22	59.16	24.15	12.8	2.08	0.40	0.11	1.27	1300[^]
23	51.99	27.69	15.0	3.17	0.64	0.10	1.29	1180[^]
24	61.77	27.76	6.80	2.28	0.23	0.10	1.04	1220
25	59.09	29.70	6.66	2.34	0.69	0.12	1.35	1300
26	61.37	24.59	8.48	2.75	1.16	0.45	1.13	1260[^]
27	61.37	28.18	6.89	1.62	0.60	0.26	0.99	1200
28	60.64	29.89	4.82	1.45	1.48	0.65	1.01	1240[^]
29	58.20	26.38	11.0	1.85	0.86	0.74	0.89	1180[^]
30	59.70	26.73	9.45	1.89	0.66	0.47	1.03	1220[^]
31	64.37	26.41	6.39	1.44	0.17	0.08	1.11	1260
32	62.42	25.38	8.07	1.86	0.99	0.22	1.00	1200[^]
33	64.25	27.44	4.43	1.93	0.69	0.15	1.05	1240
34	62.41	27.34	6.01	1.84	0.71	0.48	1.17	1200

35	64.78	27.84	4.01	1.52	0.23	0.18	1.38	1230
36	62.40	24.38	8.40	1.52	1.10	0.38	1.78	1230[$]
37	60.81	26.80	8.03	1.61	0.63	0.27	1.82	1250[$]
38	65.08	26.58	3.04	1.71	1.17	0.38	1.95	1260[^]
39	61.84	27.24	6.63	2.09	0.11	0.10	1.94	1250
40	62.85	24.82	8.00	1.52	0.11	0.11	2.54	1210[$]
41	63.14	24.47	6.06	1.61	1.34	0.55	2.76	1240
42	60.75	24.51	8.08	1.49	2.01	0.63	2.46	1240
43	65.46	24.12	4.80	1.68	1.67	0.44	1.80	1200
44	59.04	23.81	12.8	1.93	0.98	0.42	0.87	1180
45	59.62	19.07	9.10	1.82	7.26	1.95	0.99	1180
46	64.44	22.99	5.66	1.29	2.37	0.79	2.39	1100
47	60.63	21.90	8.62	1.64	3.51	1.13	2.43	1140
48	61.65	24.33	4.90	1.80	3.91	1.23	2.06	1200
49	64.14	22.79	6.87	1.62	1.89	0.88	1.74	1200[^]
50	64.97	24.71	6.43	1.61	0.49	0.21	1.54	1220[$]
51	65.83	26.13	2.03	1.46	1.77	0.69	2.02	1220
52	63.64	23.28	5.28	1.14	4.17	1.15	1.26	1200[^]
53	63.05	26.20	5.47	1.86	0.14	0.06	3.13	1200
54	62.69	26.03	4.67	1.79	0.17	0.09	4.47	1240[^]
55	68.28	21.80	6.09	1.62	0.32	0.14	1.65	1180
56	66.55	22.88	6.51	1.63	0.19	0.06	2.09	1180
57	59.79	14.66	11.7	1.84	6.78	1.54	3.34	1140
58	64.17	20.76	9.26	1.61	0.49	0.21	3.45	1220
59	66.54	20.51	8.03	1.28	1.17	0.42	1.98	1190
60	60.55	28.21	6.02	1.77	1.15	0.58	1.67	1200[^]
61	70.55	19.24	5.67	1.62	0.84	0.35	1.65	1160
62	59.05	25.00	7.05	1.99	4.78	1.41	0.50	1190
63	61.35	21.74	6.94	1.96	2.96	0.96	3.99	1140
64	63.84	23.60	4.46	1.62	0.99	0.23	5.17	1180
65	64.17	17.36	10.1	1.95	1.70	0.43	4.11	1180
66	67.29	22.02	4.02	1.29	0.60	0.22	4.52	1220
67	68.40	19.04	4.84	1.45	1.30	0.30	4.62	1200
68	66.08	25.94	4.03	1.37	0.78	0.23	1.51	1200
69	66.71	19.12	10.0	1.45	1.15	0.36	1.08	1200
70	67.06	25.02	4.39	1.60	0.80	0.17	0.91	1220[^]
71	63.66	27.37	5.63	1.93	0.34	0.08	0.95	1280[^]
72	66.15	25.26	4.80	1.76	0.33	0.13	1.52	1300
73	61.50	26.21	6.83	1.85	1.91	0.22	1.44	1260
74	65.10	27.38	3.21	1.57	0.69	0.19	1.82	1270
75	66.40	23.87	5.11	1.31	1.26	0.88	1.08	1240[$]
76	66.86	25.03	3.63	1.49	1.08	0.53	1.30	1250
77	59.20	23.12	10.1	1.18	3.35	1.01	1.87	1160[^]
78	59.25	25.62	8.60	1.27	2.24	0.70	2.18	1260
79	66.67	26.63	3.20	1.44	0.52	0.25	1.22	1280

80	60.51	25.33	9.31	1.54	1.52	0.63	1.09	1220
81	62.30	28.68	4.04	1.62	1.39	0.75	1.15	1270
82	62.53	26.21	4.71	1.80	2.78	0.67	1.17	1280^
83	63.78	25.45	5.42	1.20	0.75	0.33	3.03	1220
84	64.29	24.64	5.02	1.44	1.15	0.35	3.05	1230^
85	64.69	26.05	4.19	1.36	0.29	0.23	3.14	1240
86	63.42	24.38	6.89	1.62	1.89	0.70	1.03	1220^
87	63.19	23.99	7.67	1.78	1.68	0.56	1.06	1180$
88	60.10	28.00	5.83	1.37	2.61	1.00	1.02	1250
89	61.34	28.98	4.22	1.53	1.82	0.85	1.19	1240^
90	64.13	25.36	6.45	1.45	0.92	0.34	1.27	1230^
91	61.47	28.72	4.46	1.30	2.20	0.77	1.00	1240^

^ Test data; $ Validation data

APPENDIX 6

Table 6.B: Mean and Standard deviation of oxide weight percentages (wt %) used in the normalization of the oxides and *IDT* data

	SiO_2 wt (%) (x_1)	**Al_2O_3 wt (%) (x_2)**	**Fe_2O_3 wt (%) (x_3)**	**CaO wt (%) (x_4)**	**MgO wt (%) (x_5)**	**TiO_2 wt (%) (x_6)**	**K_2O+Na_2O wt (%) (x_7)**	***IDT* (^{O}C)$^{\#}$**
Mean	62.52	25.37	6.51	1.67	1.47	0.51	1.87	1222.19
Std. dev.	3.23	3.09	2.46	0.34	1.31	0.38	0.97	42.68

APPENDIX 6

SVR Model 6.C: SVR model pertaining to the *Initial Deformation Temperature* (IDT) prediction

$\widehat{IDT} =$ - 0.0017934099824415061 * (pow((exp(-0.836 * (-0.3161524239882646 - pc1) * (-0.3161524239882646 - pc1)) + exp(-0.836 * (-0.8576841902400439 - pc2) * (-0.8576841902400439 - pc2)) + exp(-0.836 * (0.11584523230265795 - pc3) * (0.11584523230265795 - pc3)) + exp(-0.836 * (0.6356919885958912 - pc4) * (0.6356919885958912 - pc4))), 4.5)) + 3.7465783459379634E-4 * (pow((exp(-0.836 * (1.239600373940927 - pc1) * (1.239600373940927 - pc1)) + exp(-0.836 * (-0.7518600341488825 - pc2) * (-0.7518600341488825 - pc2)) + exp(-0.836 * (-0.021295991055272884 - pc3) * (-0.021295991055272884 - pc3)) + exp(-0.836 * (-1.0178584003771907 - pc4) * (-1.0178584003771907 - pc4))), 4.5)) - 0.001527369739231294 * (pow((exp(-0.836 * (0.3100795317173647 - pc1) * (0.3100795317173647 - pc1)) + exp(-0.836 * (-1.0552087176196738 - pc2) * (-1.0552087176196738 - pc2)) + exp(-0.836 * (-0.797499203412895 - pc3) * (-0.797499203412895 - pc3)) + exp(-0.836 * (1.3057712459323405 - pc4) * (1.3057712459323405 - pc4))), 4.5)) - 0.002180496343803871 * (pow((exp(-0.836 * (0.9235172960656939 - pc1) * (0.9235172960656939 - pc1)) + exp(-0.836 * (-1.4147496663864565 - pc2) * (-1.4147496663864565 - pc2)) + exp(-0.836 * (1.201098907308331 - pc3) * (1.201098907308331 - pc3)) + exp(-0.836 * (-2.6846512404088143 - pc4) * (-2.6846512404088143 - pc4))), 4.5)) - 0.01993569529858484 * (pow((exp(-0.836 * (0.8739472300903176 - pc1) * (0.8739472300903176 - pc1)) + exp(-0.836 * (0.053868553966073054 - pc2) * (0.053868553966073054 - pc2)) + exp(-0.836 * (-0.9801417229266889 - pc3) * (-0.9801417229266889 - pc3)) + exp(-0.836 * (-0.0749253797744924 - pc4) * (-0.0749253797744924 - pc4))), 4.5)) - 0.002621721674837597 * (pow((exp(-0.836 * (0.6851259773920778 - pc1) * (0.6851259773920778 - pc1)) + exp(-0.836 * (-0.9881642164455552 - pc2) * (-0.9881642164455552 - pc2)) + exp(-0.836 * (-1.2780362321403975 - pc3) * (-1.2780362321403975 - pc3)) + exp(-0.836 * (-0.12199736973927593 - pc4) * (-0.12199736973927593 - pc4))), 4.5)) - 0.0024665431523053284 * (pow((exp(-0.836 * (-0.9047664607781067 - pc1) * (-0.9047664607781067 - pc1)) + exp(-0.836 * (1.1077829706160858 - pc2) * (1.1077829706160858 - pc2)) + exp(-0.836 * (0.5262878291205736 - pc3) * (0.5262878291205736 - pc3)) + exp(-0.836 * (0.564067654770551 - pc4) * (0.564067654770551 - pc4))), 4.5)) + 0.0038412798009185454 * (pow((exp(-0.836 * (-0.22131756574657543 - pc1) * (-0.22131756574657543 - pc1)) + exp(-0.836 * (-0.28383757355362976 - pc2) * (-0.28383757355362976 - pc2)) + exp(-0.836 * (-0.23306132650262723 - pc3) * (-0.23306132650262723 - pc3)) + exp(-0.836 * (0.9914091272866141 - pc4) * (0.9914091272866141 - pc4))), 4.5)) - 0.004782440462263265 * (pow((exp(-0.836 * (0.48177020141199706 - pc1) * (0.48177020141199706 - pc1)) + exp(-0.836 * (-0.6185150795792027 - pc2) * (-0.6185150795792027 - pc2)) + exp(-0.836 * (-0.04686476101064932 - pc3) * (-0.04686476101064932 - pc3)) + exp(-0.836 * (0.4109810338728228 - pc4) * (0.4109810338728228 - pc4))), 4.5)) - 0.004365579359484319 * (pow((exp(-0.836 * (1.1272544532900697 - pc1) * (1.1272544532900697 - pc1)) + exp(-0.836 * (-0.21597589483580978 - pc2) * (-0.21597589483580978 - pc2)) + exp(-0.836 * (-0.2659833440704244 - pc3) * (-0.2659833440704244 - pc3)) + exp(-0.836 * (1.2849056548226425 - pc4) * (1.2849056548226425 - pc4))), 4.5)) - 0.0030426512441456737 * (pow((exp(-0.836 * (0.5095007284178883 - pc1) * (0.5095007284178883 - pc1)) + exp(-0.836 * (1.5668518943263279 - pc2) * (1.5668518943263279 - pc2)) + exp(-0.836 * (0.5845571259454397 - pc3) * (0.5845571259454397 - pc3)) + exp(-0.836 * (0.27775858788674274 - pc4) * (0.27775858788674274 - pc4))), 4.5)) - 0.003610214032883117 * (pow((exp(-0.836 * (-0.865169687997901 - pc1) * (-0.865169687997901 - pc1)) + exp(-0.836 * (1.0352502473244263 - pc2) * (1.0352502473244263 - pc2)) + exp(-0.836 * (1.6157151038780542 - pc3) * (1.6157151038780542 - pc3)) + exp(-0.836 *

(1.0833745362960874 - pc4) * (1.0833745362960874 - pc4))), 4.5)) - 0.004167523310420265 * (pow((exp(-0.836 * (-2.120942835800243 - pc1) * (-2.120942835800243 - pc1)) + exp(-0.836 * (-0.05959713421007199 - pc2) * (-0.05959713421007199 - pc2)) + exp(-0.836 * (-1.1459111483568185 - pc3) * (-1.1459111483568185 - pc3)) + exp(-0.836 * (0.7425527818572352 - pc4) * (0.7425527818572352 - pc4))), 4.5)) + 0.00602453518955581 * (pow((exp(-0.836 * (0.2110174042197664 - pc1) * (0.2110174042197664 - pc1)) + exp(-0.836 * (-0.653264227003972 - pc2) * (-0.653264227003972 - pc2)) + exp(-0.836 * (0.3124548800308624 - pc3) * (0.3124548800308624 - pc3)) + exp(-0.836 * (-0.6215869761755202 - pc4) * (-0.6215869761755202 - pc4))), 4.5)) - 0.0010237680816353408 * (pow((exp(-0.836 * (-1.7985328695209404 - pc1) * (-1.7985328695209404 - pc1)) + exp(-0.836 * (0.5729156878712146 - pc2) * (0.5729156878712146 - pc2)) + exp(-0.836 * (-0.1097875536318003 - pc3) * (-0.1097875536318003 - pc3)) + exp(-0.836 * (-0.8851442182289885 - pc4) * (-0.8851442182289885 - pc4))), 4.5)) - 0.007681321821171909 * (pow((exp(-0.836 * (0.4134070246587206 - pc1) * (0.4134070246587206 - pc1)) + exp(-0.836 * (0.829211435113403 - pc2) * (0.829211435113403 - pc2)) + exp(-0.836 * (-0.35004964263465016 - pc3) * (-0.35004964263465016 - pc3)) + exp(-0.836 * (0.3795477254809649 - pc4) * (0.3795477254809649 - pc4))), 4.5)) + 0.016305260446174318 * (pow((exp(-0.836 * (1.322735710452216 - pc1) * (1.322735710452216 - pc1)) + exp(-0.836 * (-0.2931758190900727 - pc2) * (-0.2931758190900727 - pc2)) + exp(-0.836 * (-0.8438387214524109 - pc3) * (-0.8438387214524109 - pc3)) + exp(-0.836 * (0.7326144092748482 - pc4) * (0.7326144092748482 - pc4))), 4.5)) + 0.01687823701662612 * (pow((exp(-0.836 * (0.038432100329678254 - pc1) * (0.038432100329678254 - pc1)) + exp(-0.836 * (0.47612889161300437 - pc2) * (0.47612889161300437 - pc2)) + exp(-0.836 * (0.14844742323405097 - pc3) * (0.14844742323405097 - pc3)) + exp(-0.836 * (0.13325871045394966 - pc4) * (0.13325871045394966 - pc4))), 4.5)) - 0.005101004522095592 * (pow((exp(-0.836 * (-0.07765034712582977 - pc1) * (-0.07765034712582977 - pc1)) + exp(-0.836 * (-1.4999901117048753 - pc2) * (-1.4999901117048753 - pc2)) + exp(-0.836 * (-0.29730509877668926 - pc3) * (-0.29730509877668926 - pc3)) + exp(-0.836 * (0.0488806848030746 - pc4) * (0.0488806848030746 - pc4))), 4.5)) - 0.006662596185437997 * (pow((exp(-0.836 * (-0.20413803371730396 - pc1) * (-0.20413803371730396 - pc1)) + exp(-0.836 * (-0.008156923853899817 - pc2) * (-0.008156923853899817 - pc2)) + exp(-0.836 * (0.2915640023367995 - pc3) * (0.2915640023367995 - pc3)) + exp(-0.836 * (1.010260295597532 - pc4) * (1.010260295597532 - pc4))), 4.5)) + 0.0020687874668215583 * (pow((exp(-0.836 * (-0.465937710451296 - pc1) * (-0.465937710451296 - pc1)) + exp(-0.836 * (1.4378044123766707 - pc2) * (1.4378044123766707 - pc2)) + exp(-0.836 * (1.521604582969593 - pc3) * (1.521604582969593 - pc3)) + exp(-0.836 * (0.5100698683822472 - pc4) * (0.5100698683822472 - pc4))), 4.5)) - 9.687074536036598E-4 * (pow((exp(-0.836 * (0.4107369108416689 - pc1) * (0.4107369108416689 - pc1)) + exp(-0.836 * (-1.3931024894511097 - pc2) * (-1.3931024894511097 - pc2)) + exp(-0.836 * (0.9352991541145571 - pc3) * (0.9352991541145571 - pc3)) + exp(-0.836 * (1.1397801430731473 - pc4) * (1.1397801430731473 - pc4))), 4.5)) + 0.011038614666537237 * (pow((exp(-0.836 * (-0.6735316983620093 - pc1) * (-0.6735316983620093 - pc1)) + exp(-0.836 * (-0.43306149359186824 - pc2) * (-0.43306149359186824 - pc2)) + exp(-0.836 * (-0.22121481995348824 - pc3) * (-0.22121481995348824 - pc3)) + exp(-0.836 * (-0.4420737246405684 - pc4) * (-0.4420737246405684 - pc4))), 4.5)) - 0.018899056663940324 * (pow((exp(-0.836 * (0.09000149425015841 - pc1) * (0.09000149425015841 - pc1)) + exp(-0.836 * (0.4749521636535863 - pc2) * (0.4749521636535863 - pc2)) + exp(-0.836 * (-1.653785302251064 - pc3) * (-1.653785302251064 - pc3)) + exp(-0.836 * (-0.04991100770321792 - pc4) * (-0.04991100770321792 - pc4))), 4.5)) - 0.010129808573292369 * (pow((exp(-0.836 * (0.02217783961206232 - pc1) * (0.02217783961206232 - pc1)) + exp(-0.836 * (-0.45231816249093965 - pc2) * (-0.45231816249093965 - pc2)) + exp(-0.836 * (-0.848732634165855 - pc3) * (-0.848732634165855 - pc3)) + exp(-0.836 * (-0.7753627261311334 - pc4) * (-0.7753627261311334 - pc4))), 4.5)) - 0.03534590725008338 * (pow((exp(-0.836 * (1.1501850043564068 - pc1) * (1.1501850043564068 - pc1)) + exp(-0.836 * (-

0.4826482880188189 - pc2) * (-0.4826482880188189 - pc2)) + exp(-0.836 * (-0.5952492993257168 - pc3) * (-0.5952492993257168 - pc3)) + exp(-0.836 * (0.5623246574375055 - pc4) * (0.5623246574375055 - pc4))), 4.5)) + 0.012228895385219504 * (pow((exp(-0.836 * (0.35070826547623146 - pc1) * (0.35070826547623146 - pc1)) + exp(-0.836 * (0.5528757894385362 - pc2) * (0.5528757894385362 - pc2)) + exp(-0.836 * (-0.21481282460048534 - pc3) * (-0.21481282460048534 - pc3)) + exp(-0.836 * (0.45595507629074544 - pc4) * (0.45595507629074544 - pc4))), 4.5)) - 0.0022027216856989607 * (pow((exp(-0.836 * (-0.9152480465963996 - pc1) * (-0.9152480465963996 - pc1)) + exp(-0.836 * (-1.2565996585962753 - pc2) * (-1.2565996585962753 - pc2)) + exp(-0.836 * (1.074946062481113 - pc3) * (1.074946062481113 - pc3)) + exp(-0.836 * (-1.650134345135338 - pc4) * (-1.650134345135338 - pc4))), 4.5)) + 0.014795212490864462 * (pow((exp(-0.836 * (0.8455618137235085 - pc1) * (0.8455618137235085 - pc1)) + exp(-0.836 * (-0.6937651852125313 - pc2) * (-0.6937651852125313 - pc2)) + exp(-0.836 * (-0.7232633646739206 - pc3) * (-0.7232633646739206 - pc3)) + exp(-0.836 * (0.877636669634845 - pc4) * (0.877636669634845 - pc4))), 4.5)) - 0.0036458665756994363 * (pow((exp(-0.836 * (-0.4396164029561461 - pc1) * (-0.4396164029561461 - pc1)) + exp(-0.836 * (-0.5083928572563903 - pc2) * (-0.5083928572563903 - pc2)) + exp(-0.836 * (0.28370700993188114 - pc3) * (0.28370700993188114 - pc3)) + exp(-0.836 * (-0.4701040678518993 - pc4) * (-0.4701040678518993 - pc4))), 4.5)) - 0.0031627707823654104 * (pow((exp(-0.836 * (0.7196385058840126 - pc1) * (0.7196385058840126 - pc1)) + exp(-0.836 * (-0.09282652861974415 - pc2) * (-0.09282652861974415 - pc2)) + exp(-0.836 * (1.0088713011238495 - pc3) * (1.0088713011238495 - pc3)) + exp(-0.836 * (-0.31407547609800535 - pc4) * (-0.31407547609800535 - pc4))), 4.5)) + 8.494710687717626E-4 * (pow((exp(-0.836 * (0.7579805256977026 - pc1) * (0.7579805256977026 - pc1)) + exp(-0.836 * (-0.3053888541838141 - pc2) * (-0.3053888541838141 - pc2)) + exp(-0.836 * (-0.8883598304618445 - pc3) * (-0.8883598304618445 - pc3)) + exp(-0.836 * (-1.8256794916089014 - pc4) * (-1.8256794916089014 - pc4))), 4.5)) - 0.007326340682299035 * (pow((exp(-0.836 * (1.1569050891518575 - pc1) * (1.1569050891518575 - pc1)) + exp(-0.836 * (-0.7533147214081971 - pc2) * (-0.7533147214081971 - pc2)) + exp(-0.836 * (0.9639655197089163 - pc3) * (0.9639655197089163 - pc3)) + exp(-0.836 * (1.2682846845823617 - pc4) * (1.2682846845823617 - pc4))), 4.5)) - 0.007307349469651596 * (pow((exp(-0.836 * (0.20566630068167815 - pc1) * (0.20566630068167815 - pc1)) + exp(-0.836 * (1.5258054664268181 - pc2) * (1.5258054664268181 - pc2)) + exp(-0.836 * (-1.035468746339954 - pc3) * (-1.035468746339954 - pc3)) + exp(-0.836 * (-0.8895421870969775 - pc4) * (-0.8895421870969775 - pc4))), 4.5)) - 0.005507824775975481 * (pow((exp(-0.836 * (-0.5529830410817818 - pc1) * (-0.5529830410817818 - pc1)) + exp(-0.836 * (0.2961679692676845 - pc2) * (0.2961679692676845 - pc2)) + exp(-0.836 * (0.032004167038682974 - pc3) * (0.032004167038682974 - pc3)) + exp(-0.836 * (-0.9247014577465271 - pc4) * (-0.9247014577465271 - pc4))), 4.5)) + 0.03134475986504505 * (pow((exp(-0.836 * (0.47968405385751955 - pc1) * (0.47968405385751955 - pc1)) + exp(-0.836 * (0.16881041514616996 - pc2) * (0.16881041514616996 - pc2)) + exp(-0.836 * (-1.495013801436867 - pc3) * (-1.495013801436867 - pc3)) + exp(-0.836 * (-0.23133071773050498 - pc4) * (-0.23133071773050498 - pc4))), 4.5)) + 0.007813431108739827 * (pow((exp(-0.836 * (-0.2806561644899401 - pc1) * (-0.2806561644899401 - pc1)) + exp(-0.836 * (-0.06301940890843201 - pc2) * (-0.06301940890843201 - pc2)) + exp(-0.836 * (-0.8703463836423779 - pc3) * (-0.8703463836423779 - pc3)) + exp(-0.836 * (0.3322462902349431 - pc4) * (0.3322462902349431 - pc4))), 4.5)) - 0.006682377895246377 * (pow((exp(-0.836 * (0.1807663662493949 - pc1) * (0.1807663662493949 - pc1)) + exp(-0.836 * (0.6472201278218913 - pc2) * (0.6472201278218913 - pc2)) + exp(-0.836 * (0.5642229028694529 - pc3) * (0.5642229028694529 - pc3)) + exp(-0.836 * (0.8398537002927383 - pc4) * (0.8398537002927383 - pc4))), 4.5)) + 0.018780191706766557 * (pow((exp(-0.836 * (1.412226405135036 - pc1) * (1.412226405135036 - pc1)) + exp(-0.836 * (-0.48224855358548846 - pc2) * (-0.48224855358548846 - pc2)) + exp(-0.836 * (-0.7819322265142984 - pc3) * (-0.7819322265142984 - pc3)) + exp(-0.836 * (-1.0336783785448423E-4 - pc4) * (-

1.0336783785448423E-4 - pc4))), 4.5)) + 0.00557986814482944 * (pow((exp(-0.836 * (-0.4285202799745919 - pc1) * (-0.4285202799745919 - pc1)) + exp(-0.836 * (1.3545850254228406 - pc2) * (1.3545850254228406 - pc2)) + exp(-0.836 * (2.150584048925362 - pc3) * (2.150584048925362 - pc3)) + exp(-0.836 * (0.6786403243068435 - pc4) * (0.6786403243068435 - pc4))), 4.5)) + 0.011892194114677677 * (pow((exp(-0.836 * (0.9399676019033102 - pc1) * (0.9399676019033102 - pc1)) + exp(-0.836 * (-1.0449904578157827 - pc2) * (-1.0449904578157827 - pc2)) + exp(-0.836 * (0.272704500429088 - pc3) * (0.272704500429088 - pc3)) + exp(-0.836 * (0.9983529914838942 - pc4) * (0.9983529914838942 - pc4))), 4.5)) + 0.002876536713560682 * (pow((exp(-0.836 * (0.869162832924278 - pc1) * (0.869162832924278 - pc1)) + exp(-0.836 * (0.9236167147431259 - pc2) * (0.9236167147431259 - pc2)) + exp(-0.836 * (0.04883372540896253 - pc3) * (0.04883372540896253 - pc3)) + exp(-0.836 * (0.6066933291841611 - pc4) * (0.6066933291841611 - pc4))), 4.5)) - 0.001439644382434863 * (pow((exp(-0.836 * (-3.495654594560473 - pc1) * (-3.495654594560473 - pc1)) + exp(-0.836 * (-1.7641386765260283 - pc2) * (-1.7641386765260283 - pc2)) + exp(-0.836 * (-0.6147987936883903 - pc3) * (-0.6147987936883903 - pc3)) + exp(-0.836 * (1.2031566583085616 - pc4) * (1.2031566583085616 - pc4))), 4.5)) - 0.0012681801726392686 * (pow((exp(-0.836 * (-1.5628514800904572 - pc1) * (-1.5628514800904572 - pc1)) + exp(-0.836 * (3.7155213535143488 - pc2) * (3.7155213535143488 - pc2)) + exp(-0.836 * (3.0290836441613935 - pc3) * (3.0290836441613935 - pc3)) + exp(-0.836 * (-0.6681784901705634 - pc4) * (-0.6681784901705634 - pc4))), 4.5)) - 0.001375428601705236 * (pow((exp(-0.836 * (1.010735956417599 - pc1) * (1.010735956417599 - pc1)) + exp(-0.836 * (0.1390403377435575 - pc2) * (0.1390403377435575 - pc2)) + exp(-0.836 * (0.854322732781934 - pc3) * (0.854322732781934 - pc3)) + exp(-0.836 * (-1.2405469548155705 - pc4) * (-1.2405469548155705 - pc4))), 4.5)) + 0.08298037760916481 * (pow((exp(-0.836 * (-0.04357233967172938 - pc1) * (-0.04357233967172938 - pc1)) + exp(-0.836 * (0.6346315603636001 - pc2) * (0.6346315603636001 - pc2)) + exp(-0.836 * (-0.6112040094805213 - pc3) * (-0.6112040094805213 - pc3)) + exp(-0.836 * (-0.5263951476153189 - pc4) * (-0.5263951476153189 - pc4))), 4.5)) - 0.002462723323728153 * (pow((exp(-0.836 * (-0.5284655536146238 - pc1) * (-0.5284655536146238 - pc1)) + exp(-0.836 * (0.22129986611639127 - pc2) * (0.22129986611639127 - pc2)) + exp(-0.836 * (0.09412021223473827 - pc3) * (0.09412021223473827 - pc3)) + exp(-0.836 * (0.7021663266577084 - pc4) * (0.7021663266577084 - pc4))), 4.5)) + 0.0012831592988290625 * (pow((exp(-0.836 * (-2.01573016169638 - pc1) * (-2.01573016169638 - pc1)) + exp(-0.836 * (1.1515948224198211 - pc2) * (1.1515948224198211 - pc2)) + exp(-0.836 * (-2.957369103395309 - pc3) * (-2.957369103395309 - pc3)) + exp(-0.836 * (-2.067150115231318 - pc4) * (-2.067150115231318 - pc4))), 4.5)) + 8.181537833923074E-4 * (pow((exp(-0.836 * (0.13537667535004783 - pc1) * (0.13537667535004783 - pc1)) + exp(-0.836 * (-0.9935223697434423 - pc2) * (-0.9935223697434423 - pc2)) + exp(-0.836 * (1.5064664780456531 - pc3) * (1.5064664780456531 - pc3)) + exp(-0.836 * (2.2260302777000196 - pc4) * (2.2260302777000196 - pc4))), 4.5)) + 0.007605623822766256 * (pow((exp(-0.836 * (1.1575851732438371 - pc1) * (1.1575851732438371 - pc1)) + exp(-0.836 * (0.007897211969133613 - pc2) * (0.007897211969133613 - pc2)) + exp(-0.836 * (0.22905310404834275 - pc3) * (0.22905310404834275 - pc3)) + exp(-0.836 * (0.6142264448463629 - pc4) * (0.6142264448463629 - pc4))), 4.5)) - 0.0022011690676431044 * (pow((exp(-0.836 * (-1.1632836339084873 - pc1) * (-1.1632836339084873 - pc1)) + exp(-0.836 * (-0.9472874004376449 - pc2) * (-0.9472874004376449 - pc2)) + exp(-0.836 * (-0.8697222040662933 - pc3) * (-0.8697222040662933 - pc3)) + exp(-0.836 * (-0.3494909761509669 - pc4) * (-0.3494909761509669 - pc4))), 4.5)) + 0.0029493005785105756 * (pow((exp(-0.836 * (-1.6265941595278162 - pc1) * (-1.6265941595278162 - pc1)) + exp(-0.836 * (0.03603663127401795 - pc2) * (0.03603663127401795 - pc2)) + exp(-0.836 * (0.7812804982136997 - pc3) * (0.7812804982136997 - pc3)) + exp(-0.836 * (0.5494021732038121 - pc4) * (0.5494021732038121 - pc4))), 4.5)) + 0.0032683007145808793 * (pow((exp(-0.836 * (1.1191585237531723 - pc1) * (1.1191585237531723 - pc1)) + exp(-0.836 * (-

0.3950825637570728 - pc2) * (-0.3950825637570728 - pc2)) + exp(-0.836 * (1.026893255204098 - pc3) * (1.026893255204098 - pc3)) + exp(-0.836 * (-2.4775304990330724 - pc4) * (-2.4775304990330724 - pc4))), 4.5)) - 0.08703573385487168 * (pow((exp(-0.836 * (0.005377189559608766 - pc1) * (0.005377189559608766 - pc1)) + exp(-0.836 * (0.5684290737649728 - pc2) * (0.5684290737649728 - pc2)) + exp(-0.836 * (-0.5030394827995469 - pc3) * (-0.5030394827995469 - pc3)) + exp(-0.836 * (-0.4217309432939644 - pc4) * (-0.4217309432939644 - pc4))), 4.5)) - 0.00349153229606815 * (pow((exp(-0.836 * (-1.3904646154826095 - pc1) * (-1.3904646154826095 - pc1)) + exp(-0.836 * (-1.4855040475579773 - pc2) * (-1.4855040475579773 - pc2)) + exp(-0.836 * (-0.1811346039165674 - pc3) * (-0.1811346039165674 - pc3)) + exp(-0.836 * (-1.1983341137190233 - pc4) * (-1.1983341137190233 - pc4))), 4.5)) + 0.007420581151774166 * (pow((exp(-0.836 * (0.28238816803785177 - pc1) * (0.28238816803785177 - pc1)) + exp(-0.836 * (1.7173060303641612 - pc2) * (1.7173060303641612 - pc2)) + exp(-0.836 * (-0.47835304032328246 - pc3) * (-0.47835304032328246 - pc3)) + exp(-0.836 * (-0.8238681008394209 - pc4) * (-0.8238681008394209 - pc4))), 4.5)) + 0.0023980487389680077 * (pow((exp(-0.836 * (0.6534013790462456 - pc1) * (0.6534013790462456 - pc1)) + exp(-0.836 * (1.0317866531758357 - pc2) * (1.0317866531758357 - pc2)) + exp(-0.836 * (0.7396418131590212 - pc3) * (0.7396418131590212 - pc3)) + exp(-0.836 * (-0.4134865573967652 - pc4) * (-0.4134865573967652 - pc4))), 4.5))
- 0.20657144623082332

... (6.6)

where $\widehat{IDT}$ refer to the normalized values of *IDT*; the magnitudes of *PCs* in equation (6.6) are calculated using the equations (6.1) to (6.4).

CHAPTER 7

THESIS CONCLUSION

7.1 INTRODUCTION

This chapter concludes the thesis by presenting (a) the importance of modeling and optimization in chemical and biochemical processes using the state-of-the-art computational intelligence (CI) based methodologies, (b) the principal results obtained in the studies described in Chapters 3 to 6 and the conclusions reached thereof, and (c) the future scope of the research area dealt with in this thesis.

Chemical processes of various kinds transform raw materials/chemicals into useful and valuable forms or products. The chemical industry encompasses numerous areas such as commodity chemicals, wood chemicals, petroleum, petrochemicals, specialty chemicals, surfactants, pigments, dyes, dyestuff, paints, plastics, polymers, resins, synthetic rubber, herbicides, fungicides, insecticides and pesticides, fertilizers, bio-chemicals, biotechnology and pharmaceuticals, food processing, electronics/ fuel cells, energy generation, etc. Some of the important tasks conducted by the chemists and chemical engineers/technologists are as given below.

- Envisage, devise, and develop new chemical processes and products,
- Design and fabricate/procure process equipment, appliances and services for safe, economical, and eco-friendly process operation.
- Arrange and manage services, for example, inventory control in continuous running mode process plant, and related logistics.
- Optimize the process operation for efficient, cost-effective and profitable use of the resources.

Availability of a mathematical process model greatly assists in performing many of the above listed tasks. An accurate model provides vital information about the steady-state/dynamic process behavior and thus saves the time and cost of experimentation incurred in studying the influence of various process parameters and operating conditions. Models are also useful in process monitoring for control and fault detection and diagnosis as also monitoring the product quality. Modeling for process identification is necessary for instrumentation design and development of *Programmable*

Logic Controllers (PLC) or *Distributed Control System* (DCS). In dynamic process systems such as high temperature operations, pyrolysis of biomass and reactors/reactions operating under supercritical conditions, the soft-sensor models serve a critical purpose of providing information about difficult to monitor variables. The model-based control strategies have been found to yield better performance than the conventional proportional-integral-derivative (PID) control especially for processes exhibiting nonlinear dynamics such as nonisothermal continuous stirred tank reactor (CSTR). Process models are of significant utility in the detection and diagnosis of faults in chemical processes wherein they are used to learn the correct associations between the faults and the operating conditions.

Among various approaches that are available for modeling, the phenomenological (also termed "first principles" or "mechanistic") and empirical are widely used in process engineering practice. The phenomenological models are constructed from the knowledge of the mass, momentum, and energy balances, as well as other chemical engineering and thermodynamics principles. In general, obtaining the stated information is time-consuming, costly and tedious, and frequently requires extensive experimentation. The other approach namely empirical modeling that uses linear/nonlinear regression techniques requires that the model structure (form) is specified a *priori*. In general, satisfying this condition, especially for nonlinear chemical processes that are encountered frequently in practice, is a time-consuming and cumbersome task. The reason being it necessitates choosing by trial and error an appropriate mathematical form of a data-fitting model from numerous competing forms. Due to the above-stated difficulties in respect of the phenomenological and empirical modeling, there existed a need for modeling paradigms that do not require (a) a detailed knowledge of the first principles governing the process, and (b) pre-specification of the data fitting function.

For any chemical process it is at most necessary that its operation is raw-material and energy efficient, safe, economical, and environment-friendly. This requires that the process operation is optimized. It may be noted that for optimizing a process its model (phenomenological or empirical) is necessary.

In process engineering, the objectives of optimization are, for instance, maximization of process yield/selectivity/turn-over number, minimization of operational cost/ total product cost (TPC), and maximization of conversion/profit. Commonly, deterministic gradient based algorithms are used in process optimization. These need computation of the derivative of the objective function (to be minimized/maximized) at each step towards securing an optimal solution, which is computationally expensive. Also, they require the objective function to be smooth, continuous and differentiable; these conditions are often difficult to meet simultaneously in real practice. Moreover, gradient descent algorithms have a tendency to get stuck in to a local optimum instead of the global optimum on the objective function surface. This results in a sub-optimal solution. The above-stated deficiencies of the conventional deterministic optimization methods necessitated usage of alternative strategies that are not strict about the form of the objective function and at the same time provide a globally optimal solution.

In recent years, a number of state-of-the-art computational intelligence (CI) based methods have offered themselves as attractive alternatives to the conventional modeling and optimization methods. These methods have several attractive properties that overcome a number of deficiencies of the conventional phenomenological and empirical modeling techniques as also the commonly used deterministic gradient based optimization methods. Accordingly, in the present thesis three CI-based methodologies, namely, *Artificial Neural Networks* (ANNs), *Support Vector Regression* (SVR), and *Genetic Programming* (GP) have been employed for the modeling of selected chemical and biochemical systems. For optimization of a process, the widely used artificial intelligence (AI) based stochastic optimization method, namely *Genetic Algorithm* (GA), which overcomes the above-stated limitations of the gradient based methods has been employed. The studies presented in this thesis show the diversity of applications of the CI-based methods for modeling and optimization of chemical and biochemical systems.

7.2 SUMMARY AND PRINCIPAL RESULTS OF STUDIES PRESENTED IN THE THESIS

This thesis is organized in seven chapters. Chapters 1 and 2 respectively provide the rationale and the salient features of the thesis and a detailed exposition of the origin, implementation and applications of the CI-based modeling and optimization methods employed in the thesis. The second chapter also details a well-known dimensionality reduction method, namely *principal component analysis* (PCA) that has been employed in this thesis for reducing the dimensionality of the input space of the CI-based models.

Chapter 3 presents the development of *Multilayer Perceptron* (MLP) *Neural Network*- and GP- based models for predicting the rate constants of the photo-catalytic degradation (PCD) reaction for treating the water contaminated by the pharmaceutical pollutants. In this study, the reaction conditions and molecular structural property attributes have been used as the model inputs. The models were developed for three pharmaceutical pollutants, namely, Ciprofloxacin, Naproxen, and Paracetamol. In the models' input space, the three pharmaceutical pollutants were differentiated using a number of attributes related to their molecular structures. In this work, PCA was performed on the fourteen-dimensional input space consisting of four experimental parameters and ten influential structural parameters of each pharmaceutical molecule. The MLP- and GP-based models predicting the rate constant of the PCD, proposed in this study, exhibit an excellent prediction accuracy and generalization performance. In addition to the modeling, the chapter also reports results of the reaction optimization conducted using the genetic algorithm (GA) strategy. Here, the objective of optimization was to obtain maximum possible degradation of pharmaceutical pollutants in a contaminated water under milder conditions. The noteworthy features of this study are as follows.

- For the first time, state-of-the-art CI-based exclusively data-driven models have been developed for computing the rate constant (K_c) for the photocatalytic degradation (PCD) of pharmaceutical pollutants namely Ciprofloxacin (CFX), Naproxen (NPX) and Paracetamol (PARA).

- The Genetic Programming (GP) based models predicting the rate constant of the photocatalytic degradation, proposed in this study exhibit an excellent prediction accuracy [*correlation coefficient* (*CC*) > 0.9] and generalization performance than the corresponding MLPNN-based models.
- The optimum PCD reaction conditions given by the hybrid genetic programming - genetic algorithm (GP-GA) modeling-optimization method were validated successfully (less than ± 3 % deviation) by conducting fresh experiments.
- The modeling and optimization strategy utilized in this study can be fruitfully used in the development of models and optimization of a wide variety of other pharmaceutical degradation reactions.

In Chapter 4, the artificial intelligence-based models have been developed for the prediction of cetane number of biofuels. It is well-known that the fuel properties of biodiesel, namely, *cetane number* (*CN*), *kinematic viscosity* (*KV*), *density* (*D*) and *higher heating value* (*HHV*), play a significant role in its utilization for the combustion process. Accordingly, Chapter 4 presents the development of exclusively data-driven models predicting the *CN* of biofuels. The models are constructed using two AI-based formalisms, namely, Genetic Programming (GP) and Multi-layer Perceptron (MLP) neural networks that employ the fatty acid methyl ester (FAME) composition and fuel properties as inputs (predictors). The carbon chain length and degree of unsaturation are two significant factors of FAME that affect the *CN* magnitudes of biodiesel and thus contribute to the ignition quality of the biodiesel. Based on the relevant data, two types of *CN* prediction models have been developed in this study using FAME composition of biodiesels (Type I models) and properties of biodiesels (Type II models). Experimental data corresponding to a large number of biodiesel samples were used in developing the two types of AI-based models. The *Principal Component Analysis* (PCA) was performed on the twelve-dimensional input space of the FAME composition-based GP- and MLP-based models with a view to reduce the input dimensionality of the models. In the case of fuel property-based models the input dimensionality was not high and therefore PCA was not performed on the input space of the corresponding models. The noteworthy results of this study are given below.

- FAME composition-based GP and MLP models exhibit an excellent *CN* prediction accuracy [correlation coefficient (*CC*) > 0.9 and root mean squared error (*RMSE*) < 4] and generalization performance for both training and test set data. The biodiesel property-based models exhibit a reasonably good prediction accuracy and generalization performance [*CC* > 0.8 and *RMSE* < 4], albeit inferior than the FAME based models, for both training and test set data.
- The novel GP method has been employed for the first time to develop models for the *CN* prediction of biodiesels.

The *ash fusion temperature* (AFT) is an important property of coals, the knowledge of which is significantly useful in predicting the occurrence of the undesired fouling and slagging phenomena in the equipment of the coal-based combustion, gasification and liquefaction processes. There exist AFT prediction models, which are based on the weight percentages of the chemical and mineral constituents (i.e., oxides) of the ashes of coals belonging to mostly a single geographical region although the coal ash properties diverge widely based on the geographical origin of coals. Most of these models are linear even though a close search of the corresponding data suggests that the relationships between the AFTs and the weight percentages of some of the chemical and mineral constituents could be nonlinear. This indicates that nonlinear models could be better suited for the AFT prediction. Accordingly, in Chapter 5, CI-based models have been proposed to predict the four components of AFT, namely, *Initial Deformation Temperature* (IDT), *Softening Temperature* (ST), *Hemispherical Temperature* (HT), and *Flow Temperature* (FT), of the coal ashes. The three CI-based exclusively data-driven formalisms used in the AFT modeling are Genetic programming (GP), Multi-layer Perceptron (MLP) Neural network, and Support Vector Regression (SVR). The noteworthy characteristics of this study are as given below.

- The models developed have a wider applicability since the chemical and mineral composition and the related AFT data used in their construction pertain to coal ash samples from a large number of geographical regions globally.

- All the four best fitting GP-based models for the prediction of four ash phase temperatures have nonlinear forms. Note that the GP method is capable of fitting either linear or nonlinear models. The fact that GP has fitted nonlinear models clearly shows that the relationship between the seven oxides in the coal ashes and the related AFTs is nonlinear and therefore linear models are not suitable for the prediction of AFT of coal ashes.

- Most of the CI-based models possess better AFT prediction accuracy and generalization performance than the currently available models.

- Since they are based upon lower number of inputs/predictors (weight percentages of oxides), the CI-based models are of reduced complexity when compared with the existing models that use more number of predictors. Models with reduced complexity are known to possess better generalization capability than their more complex counterparts.

- Among the three types of CI-based models, GP and MLP-based models have yielded overall better performance in predicting all four AFTs.

Owing to their high AFT prediction accuracy and generalization performance with wider applicability, the CI-based models developed in this work have a potential to be the preferred ones for predicting the AFT values of coal ashes from different geographies in the world.

In India, the currently mined coals have a high ash content and therefore are low in quality. These coals are predominantly used in the thermal power stations across the country. Since AFT is a good indicator of slagging and fouling that take place in the equipment of coal-based thermal power stations, it is important to develop the AFT prediction models specific to the Indian coals with intrinsically high ash content. Accordingly, Chapter 6 reports a study wherein CI-based models have been developed for the prediction of the *initial deformation temperature* (IDT) of Indian coals. The three CI-paradigms used in the modeling are GP, MLPNN and SVR. The remarkable features of this IDT modeling study are:

- For the first time, CI-based models having high prediction accuracy and generalization capability have been developed for the prediction of IDT of high ash Indian coals.
- Among GP-, MLPNN-, and SVR-based models, the SVR-based model [CC > 0.9, *RMSE* (range: 1.47 – 8.43) and *MAPE* values (range: 0.24 – 1.22)] has been found to possess better IDT prediction and generalization performance.
- The models can be significantly useful in Indian power generating stations for predicting the occurrence of the undesired and detrimental fouling and slagging phenomena in the process equipment. The mathematical models presented here can also be gainfully used by the design engineers and plant personnel of coal-based combustion, gasification and liquefaction processes.

7.3 FUTURE RESEARCH SCOPE

A large number of artificial/computational intelligence/machine learning applications in use presently or in the development arise from the exponentially increasing demand for the automation and intelligent decision making in all kinds of businesses and industries, including chemical processes. The benefits of automating critical tasks and decision making are, for example, enhanced efficiency, improved product quality or service, reduction in staff, removal of human error, and ultimately safe, energy efficient, economical, and environment friendly process operation. Artificial/computational intelligence (AI/CI) and machine learning (ML) will have tremendous influence on the automation and decision making in many different ways.

- The traditional chemical process operation is not fully automated or necessarily intelligent and thus AI/CI based modeling and optimization strategies will be increasingly used in recognizing, and learning patterns from the process related data and applying the knowledge gained thereby to, for example, implementing better process control, detecting any process malfunction and timely intervention to avoid any related undesired events, devising conditions for energy and cost efficient, optimized and safe process operation, and raw material procurement and inventory control. Traditionally, all these are complex tasks that can be effectively automated using computers/machines equipped with AI/CI capability while ensuring intelligent decision making and corresponding actions.

- The latest development in AI are deep artificial neural networks. These systems can have hundreds of layers in their architecture and possess tremendous ability to recognize and learn patterns in datasets containing millions/billions of observations. The deep ANNs have a huge potential, for example, in recognizing patterns in fluid flows, equipment noise, classifying products and quality control, detecting speedily off-spec products, and recognizing any unusual movement, radiation, odor, and noise in the vicinity of the process equipment.

- In industries such as on-shore and off-shore hydrocarbon exploration and production that pose significant risk to workers, the AI-powered robots can be used to improve productivity, mitigate the risk and cost-effectiveness.

- The molecular structure - property/activity data can be used by AI systems to design and synthesis molecules/catalysts/enzymes with the desired properties.

- Artificial intelligence can be used in biochemistry for discovering different reaction pathways to synthesize a specific bio-molecule.

- In recent years, a technology known as *Internet of Things* (IOT) is gaining wide acceptance in all types of industries. It represents use of computers, sensors, electronics and wireless connections to interconnect physical devices. In chemical industry, AI can be used to recognize patterns in the data from an IOT system for improving/optimizing process performance, improve product quality and yield rates, lower per batch costs, etc.

- BASF has developed an AI-based state-of-the-art color matching system using artificial neural networks to prepare a perfect match of the car color selected by a customer. In future, AI based similar systems can be developed to prepare specific shades of paints for homes, perfumes, etc. demanded by clients.
- AI-based smart manufacturing systems can be developed for predicting the demand for the commodities such as oil, gas and bulk chemicals, and also shortages in their supply.

To summarize, the application potential AI/CI and ML paradigms in chemical engineering is limitless and bounded only by the human imagination. The stated technologies and the more recent ones such as Big data, cloud computing, and IOT are still in infancy and their impact on the chemical industry and R&D in Chemical sciences and engineering is difficult to fathom. What is however certain is that in future their applications will grow exponentially and almost certainly result in better, effective and safe-to-use chemical products including medicines, polymers, and bio-chemicals, and efficient, safe, and environment friendly chemical processes.

LIST OF PUBLICATIONS

Publications Received from the Work Presented in the Thesis

1. Tiwary S., Naniwadekar M. Y., Sonolikar R., Bapat S., Yerudkar A., Kamble S.P., Tambe S.S. (2017). Prediction of Rate Constant of Photocatalytic Degradation of Pharmaceutical Pollutants by utilizing Artificial Intelligence based Genetic Programming Formalism, *International Conference on Sustainable Development for Energy and Environment* (ICSDEE-2017), pp.58 (pages 1-11). ISBN: 978-93-24457-19-0.

2. Tiwary S., Naniwadekar M.Y., Tambe S.S. (2017). Prediction of cetane number of biofuels using artificial intelligence-based models, *NexGen Technologies for Mining and Fuel Industries* (NxGnMiFu -2017), **1**, pp. 617-626. ISBN: 978-93-85926-40-2.

3. Tiwary S., Naniwadekar M. Y., Sonolikar R., Bapat S., Yerudkar A., Kamble S.P., Kothari G., Tambe S.S. (2018). Prediction of Rate Constants of Photocatalytic Degradation of Pharmaceutical Pollutants by Artificial Intelligence based Genetic Programming Formalism, *Current Environmental Engineering,* **5 (1)**, pp. 58-67.

4. Tambe S.S., Naniwadekar M.Y., Tiwary S., Mukherjee A., Das T.B. (2018). Prediction of Coal Ash Fusion Temperatures using Computational Intelligence based models, *International Journal of Coal Science & Technology* (A Springer Journal), pp. 1-22, DOI: 10.1007/s40789-018-0213-6 (Online).