Design of Soft Sensors for Monitoring and Control of Cement Manufacturing Processes

THESIS

Submitted in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

AJAYA KUMAR PANI

Under the Supervision of

Dr. Hare Krishna Mohanta

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
PILANI – 333 031 (RAJASTHAN) INDIA

2015

Design of Soft Sensors for Monitoring and Control of Cement Manufacturing Processes

THESIS

Submitted in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

AJAYA KUMAR PANI
(2007PHXF405P)

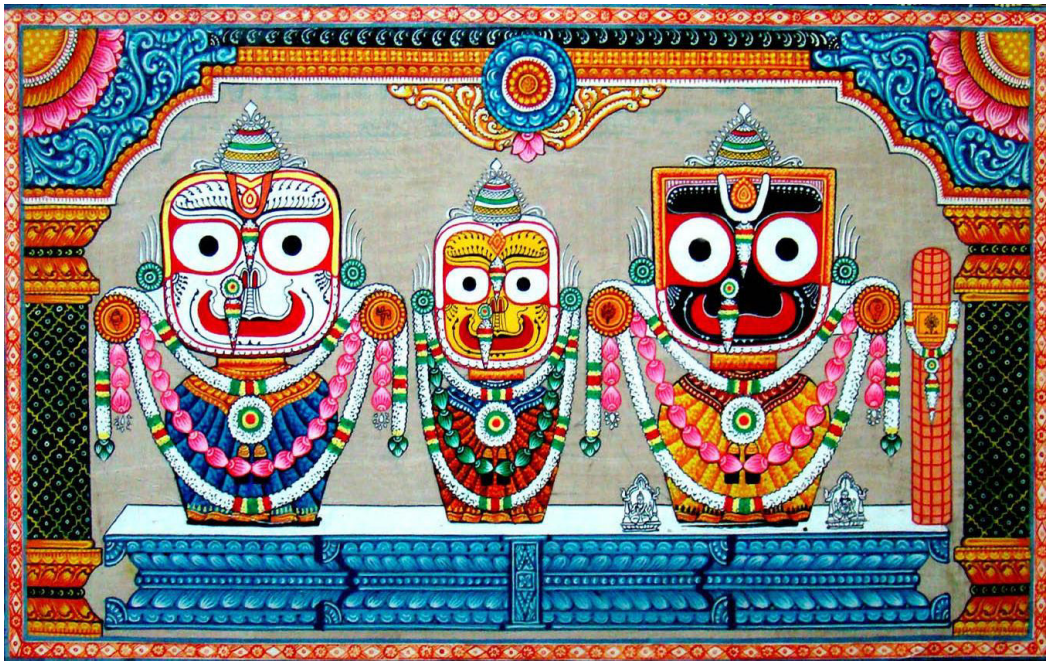
Under the Supervision of

Dr. Hare Krishna Mohanta

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
PILANI – 333 031 (RAJASTHAN) INDIA
2015

I Dedicate This Thesis To The Almighty



LORD JAGANNATH

Who Gave Me The Strength And Patience
To Complete This Study

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE

PILANI - 333031 (RAJASTHAN) INDIA

CERTIFICATE

This is to certify that the thesis entitled **"Design of Soft Sensors for Monitoring and Control of Cement Manufacturing Processes"** and submitted by **Ajaya Kumar Pani**, ID. No. **2007PHXF405P** for the award of PhD Degree of the Institute embodies the original work done by him under my supervision.

Signature in full of the Supervisor _____

DR. HARE KRISHNA MOHANTA

Assistant Professor
Department of Chemical Engineering
BITS-Pilani, Pilani Campus

Date: _____

Acknowledgements

In the first instance I would like to express my wholehearted gratitude to my supervisor **Dr. Hare Krishna Mohanta**, Assistant Professor, Department of Chemical Engineering for his constant encouragement, constructive and valuable suggestions, and moral support throughout the period of this research work starting from the initial stage of proposal writing to the final stage of thesis submission. Be it academic or personal, he has always been there with me as a mentor-cum-elder brother during all the difficult times I faced during the course of my doctoral work. It has been a privilege for me to work under his valuable guidance.

I would like to thank the members of the Doctoral Advisory Committee, Dr Surekha Bhanot, Professor, Department of Electrical and Electronics Engineering and Dr Suresh Gupta, Associate Professor & Head, Department of Chemical Engineering for their support and suggestions to carry out this work effectively.

My sincere thank goes to Prof. B. N. Jain, Vice-Chancellor, BITS-Pilani for giving me the opportunity to carry out the PhD work in BITS. I am thankful to Prof. G. Raghurama, Director (Pilani Campus), Prof. G. Sundar, Director (Off - Campus Programmes and Industry Engagement), Prof. R. K. Mittal, Director (Special Projects), Prof. S. K. Verma, Dean, Academic Research Division (PhD Program), Prof. Shan Balasubramanian, Dean, Academic and Resource Planning, Prof. R. Mahesh, Dean, Faculty Affairs, Prof. Niranjan Swain, Dean, Practice School, Prof. S. Gurunarayanan, Dean, Work Integrated Learning Programs, Prof. N. N. Sharma, Dean, Academic Registration and Counseling, Prof. Ajit Pratap Singh, Dean, Instruction Division, Prof. S. C. Sivasubramanian, Dean, Administration, Prof. H.R. Jadhav, Associate Dean, Academic Research Division (PhD Program), Prof. B. K. Rout, Associate Dean, Academic Registration and Counseling, for providing the necessary facility and infrastructure to carry out this work.

I sincerely thank Prof. B. V. Babu and Prof. G. Sundar for assigning me the task of acting as a Practice School instructor for Ultratech Cements, Kotputli Cement Works, Rajasthan, India which subsequently helped me in carrying forward my research work.

This research would not have been possible without the cooperation of the Management, Ultratech Cements, Kotputli Cement Works who generously provided the required data for carrying out this research work. I sincerely thank Mr V. K. Jain (Unit Chief), Mr Ashish Banerjee, Mr S. N. Sahu (HR), Mr Rakesh Gattani, Mr Prakhar Shrivastava (Quality control), Mr Pawan Mathur, Mr Suraj Sharma (Instrumentation), Mr P. K. Shrivastava, Mr Sandeep Jalori (Process) for their help at various stages for data collection and process understanding.

I extend my special thanks to Dr. Arvind Kumar Sharma, Associate Professor, Department of Chemical Engineering for giving valuable suggestions for improvement of the thesis.

I thank my colleagues Dr. Pradipta Chattopadhyay, Dr. Pratik N Sheth, Dr. Smita Raghuvanshi, Dr. Sushil Kumar, Dr. Nikhil Prakash, Dr. Ashish Gujrathi, Dr. Dipaloy Datta, Dr. Banasri Roy, Dr. Sonal Mazumdar, Dr. Raman Sharma, Mr Amit Jain, Ms Priya C Sande, Mr Subhajit Majumder, Mr Utkarsh Maheshwari, Dr. Bharti Khungar,

Mr Parikshit Kishore Singh and Mr Murali Monohar Pandey, research scholars Neha, Somesh, Saswat and Tapas for their constant support throughout the work.

I also thank my students Vamsi and Bhargavi who were partially involved in this work during their tenure as ME students at BITS Pilani.

Thanks are also due to Mr Babu Lal Saini, Mr Jangvir, Mr Jeevan, Mr Ashok and Mr Subodh for their cooperation during my PhD work.

I extend my special thanks to Dr. Shivi Agrawal, Assistant Professor, Department of Mathematics BITS Pilani and Dr. Kapil, Assistant Professor, Masters of Computer Application Department, NIT, Kurukshetra, for their valuable conceptual inputs regarding fuzzy inference and support vector regression techniques.

I sincerely thank my teacher Prof. Radha Kant Jha (Former Professor, Department of Chemical Engineering, IT BHU) who is solely responsible for my interest in the field of modeling and process control.

I am deeply indebted to my family. There are not enough words in the world to express gratitude to my parents who have taken a lot of pain in my upbringing and made me the person I am today. My father Sri Antarjyami Pani and my mother Late Smt Anusuya Pani have showered their love, care, concern towards me all throughout their life. Mummy, you never let us know the seriousness of your illness and left us all forever in September 2013. But despite not being there physically with me, you will continue to rule over my heart till my last breath.

My deepest sense of gratitude goes to my elder brother Mr Amiya Kumar Pani, sister in law Sasmita Pani, nieces Aishwarya and Akankhya.

Above all I would like to thank my better half Munmun for her love and constant support. She has taken a lot of pain in taking care of the house and two small kids all alone and letting me work freely at the Institute. I also take this opportunity to thank my two kids daughter Adyasha and son Aditya who have brought all the joys in our life. During the later stage of research work, they were kind enough to let their Baba go to the Institute and concentrate on the research work and thesis writing during extended hours on working days and during holidays.

Last but not the least, this acknowledgement will not be complete without thanking the Almighty God for his unparalleled grace, superior protection and guidance throughout the lows and highs of my PhD journey.

AJAYA KUMAR PANI

Abstract

In most of the process industries the end product quality is not measurable online. Unavailability of reliable hardware sensors for continuous quality monitoring at times results in production of low quality products. Soft sensors or inferential sensors are process models which use the information of easily measurable process variables and produce as output, the estimated/predicted values of difficult to measure process variables. Most of the present day approaches for soft sensor development are data based because of the various difficulties associated with development of mathematical models for complex processes.

Clinker composition and cement particle size are two of the most important quality parameters in cement manufacturing process. Unfortunately, there are no hardware sensors available for continuous monitoring of these quality parameters. Therefore, the focus of this research is to develop data-driven soft sensors for online monitoring of cement clinker quality and cement fineness.

The two processes focused upon are, (1) a chemical process where the raw mix is converted to cement clinker at high temperature in a rotary cement kiln and (2) a physical process where the clinker is ground in a vertical roller mill for cement production. The required data for these two processes were collected from a cement plant.

The grinding process data set comprised of data for three input variables and one output variable. The three input variables are measured continuously by installed hardware sensors and therefore contain some outlying observations. These outliers were detected and removed by the robust Hampel's method of outlier detection. The processed data set after outlier removal, consisted of 158 samples of input-output data. This data set was equally divided into a training set and a validation set consisting of 79 observations each. The data division was performed using Kennard-Stone maximal intra-distance criterion. The training set was used for development of different kinds of data-driven soft sensors. The different soft sensors developed include, linear and support vector regression, artificial neural network, fuzzy inference and neuro-fuzzy models of

the grinding process. The performances of the developed models were assessed with the validation data set by measuring six different statistical model performance indicators. The neuro-fuzzy and the back propagation neural network models showed better performances than the other models. The accuracy of both these models are quite acceptable as per the model acceptability criteria reported in the literature.

The clinkerization process taking place in the rotary kiln, involves nine inputs and eight outputs. Out of the nine inputs, 4 are raw meal quality parameters (measured in the laboratory) and five are kiln operating parameters. The operating parameters are measured continuously by installed hardware sensors and therefore contain outliers. After performing a comparison of different multivariate outlier detection techniques, the outliers present in the kiln input data were removed by the technique of closest distance to center method. The processed data set consisted of 223 pairs of input-output data. Using Kennard-Stone algorithm, the total data set was divided into a training set consisting of 112 samples and a validation set containing 111 samples. The training set was used to develop three kinds of feed forward neural network models and two types of fuzzy inference models. The performances of the developed soft sensors were analyzed with the validation data set by evaluating six statistical model evaluation parameters. The analysis showed that the soft sensor based on Takagi-Sugeno fuzzy inference modeling technique, has the highest accuracy in estimating eight cement clinker quality parameters.

Key words: A*rtificial neural network, ANFIS, cement fineness, clinker quality, fuzzy inference system, multivariate outlier detection, soft sensor, subset selection, support vector regression, vertical roller mill*.

Table of Contents

List of Figures

List of Tables

Notations

Abbreviations

AlM	Alumina modulus
ANFIS	Adaptive neuro fuzzy inference system
ANN	Artificial neural network
ARMAX	Autoregressive moving average model with exogenous inputs
ASTM	American society for testing materials
BIS	Bureau of Indian standards
BPNN	Back propagation neural network
CCN	Cement chemistry notation
CDC	Closest distance to center
ED	Euclidean distance
GDM	Gradient descent with momentum
FCC	Fluid catalytic cracking
FIS	Fuzzy inference system
FISM	Mamdani fuzzy inference system
FISS	Sugeno fuzzy inference system
GA	Genetic algorithm
GDM	Gradient descent with momentum
GRNN	Generalized regression neural network
ICA	Independent component analysis
IMC	Internal model control
K-S	Kennard-Stone
LM	Levenberg–Marquardt
LSF	Lime saturation factor
LS-SVR	Least square support vector regression
MAD	Median absolute deviation
MAE	Mean absolute error
MCD	Minimum covariance determinant
MD	Mahalanobis distance
MIMO	Multi input multi output
MLR	Multiple linear regression
MF	Membership function

MPC	Model predictive controller
MVE	Minimum volume ellipsoid
MVT	Multi variate trimming
NARMAX	Non-linear autoregressive moving average model with exogenous inputs
NSE	Nash-Sutcliffe efficiency
NSV	Number of support vectors
PCA	Principal component analysis
PLS	Partial least square/Projection to latent structure
RBF	Radial basis function
RBFNN	Radial basis function neural network
RHM	Resampling by half means
RMSE	Root mean squared error
RP	Resilient back propagation
RPM	Revolutions per minute
SiM	Silica modulus
SHV	Smallest half volume
SOM	Self organizing map
SRU	Sulphur recovery unit
SVM	Support vector machine
SVR	Support vector regression
TIC	Thiel's inequality coefficient
VAF	Variance account for
VRM	Vertical roller mill

Mathematical Notations

b	Bias value in SVR model
C	Covariance matrix for the N×M input data/Regularization parameter in SVR model
C'	Eigen vector matrix
c_i	Center associated with i^{th} RBF neuron
D	Distance matrix in SHV method
D_{ij}	Euclidean distance between i^{th} and j^{th} sample
ED_i	Euclidean distance of i^{th} observation from the centroid
H	Hat matrix
J	SVR model optimization function
K	Kernel function in SVR model
L	LS-SVR model optimization function
M	Total number of input variables/Control horizon in MPC
MD_i	Mahalanobis distance of ith observation from the centroid

N	Total number of samples or observations
P	Loading matrix in PCA/ Number of principal components/Prediction horizon in MPC
$Q_{0.25}$	First quartile (25% quantile) of a dataset
$Q_{0.75}$	Third quartile (75% quantile) of a dataset
R	Correlation coefficient/Rule base of FIS
T	Score matrix in PCA
X	$N \times M$ input data matrix
$\overline{X}$	Mean values of columns of X ($1 \times M$ row vector)
w	Weight vector of SVR/RBFNN/GRNN model
x_i	i^{th} value of an input variable
$x_{0.5}$	Median or 50% quantile value of a dataset
$\bar{x}$	Mean value of a dataset
x_{max}	Maximum value in the column x
x_{min}	Minimum value in the column x
y_i	Actual value of output for i^{th} observation/sample
$\widehat{y_i}$	Model predicted/estimated output
$\overline{y}$	Mean value of output

Greek Symbols

σ	Standard deviation of a dataset/RBF kernel width/kernel width in SVR model
σ_y	Standard deviation of the output dataset in SVR modeling
β	Regression coefficients of linear model
ε	Error bound/loss function parameter in SVR model
$\varphi(x)$	Non-linear function in SVR model/Basis function in RBF Neural network model
ξ_i^*, ξ	Positive slack variables in SVR model
μ	Membership function of fuzzy inference model
μ_{out}	Membership function of the output y
γ	Skewness of a dataset
κ	Kurtosis of a dataset
χ^2	Chi square distribution

Chapter - 1

Introduction

1.1 Motivation

Industrial processing plants are usually heavily instrumented with a large number of sensors. Instruments are essential in a process industry for continuous monitoring and control of different process variables which are required for different purposes (e.g. meeting stringent effluent norms, operating within safety limit, achieving desired product quality etc). In most cases, meeting these criteria demands accurate on-line continuous measurement of quality variables**.** For measurement of physical process variables such as temperature, pressure, flow rate etc., in an industry, a large variety of accurate hardware sensors and transducers are available which are quite affordable and reliable. However, for continuous online monitoring of chemical or biochemical variables related to composition (for assessment of product quality), reliable hardware sensors are not available. In most of the cases the analysis is performed in the laboratory by collecting samples and the response time can be several minutes or even hours. Even if an online analyzer is available, the cost, precision and reliability of such instruments are quite often unsatisfactory. Moreover, some common problems associated with hardware sensors for online quality monitoring are, time consuming maintenance, regular calibration, aged deterioration, insufficient accuracy, long dead time & slow dynamics, large noise and low reproducibility (Kano and Nakagawa, 2008). To maintain the quality specifications, knowledge of instantaneous composition is necessary for implementation of an efficient control system. Failure to accurately estimate important process outputs may result in

product loss, energy loss, undesirable byproduct formation and safety hazards (Zhao, 2003).

In the aforementioned context, when online sensing of quality parameters by hardware sensors is not possible, there is a need of estimating the parameter using other available informations. Therefore, soft sensors must be considered as possible alternatives for continuous online sensing of such variables for which hardware sensors are not available or have less reliability and significant time delay. Soft sensors allow online estimation of variables (mostly relating to product quality) which are otherwise difficult or impossible to monitor online, using easily measurable variables.

1.2 Soft Sensor Basics

Soft sensor is a combination of two terms: “software”, because the models are mostly computer programs, and “sensors”, because the models provide similar information as their hardware counterparts. Other synonyms for soft sensor are inferential sensors, virtual on-line analyzer or observer-based sensors.

There are a wide range of process industry problems that a soft sensor can address (Fortuna *et al*., 2007; Kadlec *et al*., 2009). Prediction of an unmeasured process variable from the available data of measured variables (process monitoring) is perhaps the most important application of a soft sensor. By doing so, process monitoring soft sensors make way for faster and more accurate decision making with regard to process operations. Besides this, other application includes process fault detection. This refers to detection of the state of the process and in the case of a deviation from the normal conditions, identification of the cause. A process industry employs large number of various sensors. Therefore, occasional failure of a sensor is always a possibility. Detection of this failure is the next application area of soft sensors which is described as sensor fault detection. Once a faulty sensor is detected and identified, it can be either reconstructed or the hardware

sensor can be replaced by another hardware sensor or soft sensor, which is trained to act as a back-up sensor of the hardware measuring device. If the soft sensor is accurate and reliable enough then this can be used in place of the hardware sensor in normal operating conditions which can lead to significant revenue saving by avoiding large investment, installation and maintenance cost associated with hardware sensors. Moreover, soft sensors also overcome the time delay problem associated with most hardware sensors thereby resulting in more effective control of the process output variable. The software tool is easier to maintain and is not subject to mechanical failures and therefore such a substitution can result in financial benefits for the process owner.

Soft sensor design is an emerging area attracting huge research interests. The heart of the soft-sensor constitutes the model of a process which takes values of the easily measurable process variables and predicts the output which is difficult to measure process variable, thereby replacing or assisting a real physical sensor. This plant model may be a first principle model (mechanistic model), black-box model (empirical model) or gray box model (hybrid model). First principle models describe the physical and chemical background of the process and hence require detailed process understanding. Black-box and gray box models are data-driven which are developed from the actual process input-output data.

While first principle models provide good process understanding and can be applied over a wide range of process data, there are difficulties in development of first principle models because: most of the processes are very complex and are not fully understood, parameters involved in the model equations are difficult to obtain and moreover, very often the model equations developed become so complex that it becomes difficult to solve. On the other hand availability of large amount of plant historical data has shifted the focus towards design of data-driven soft sensors. These, in comparison to the first principle

model based soft sensors, are more reality related and describe the true conditions of the process in a better way.

1.3 Data-Driven Soft Sensor Design Procedure

The basic steps in any data-driven soft sensor design method are: data collection, data preprocessing, model selection, parameter identification and finally model validation (Fortuna *et al*., 2007; Kadlec *et al*., 2009) which are shown in the schematic diagram below:

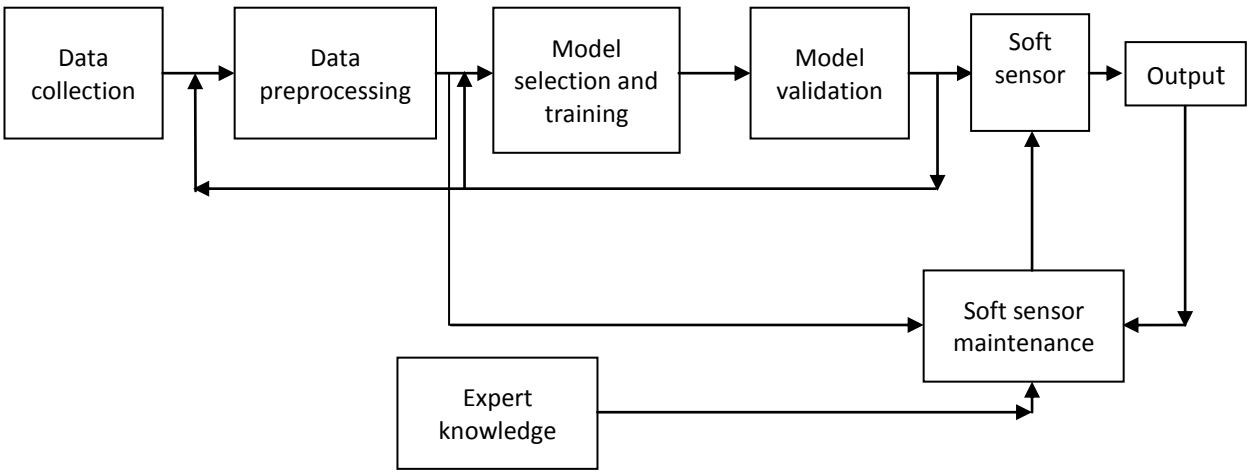


Figure 1.1: Soft Sensor Design Steps

1.3.1 Data Collection and Preprocessing

The first activity in the data-driven soft sensor design is collection of actual process data. There are sensors which record values for a process variable every few seconds or minutes. So, a dataset for a period of even a few months or some days comprises a vast amount of data which often makes the analysis quite complicated. The raw data extracted from the plant database may often suffer from one or more of these drawbacks (Fortuna *et al*., 2007; Ljung, 1999): multidimensionality, measurement noise and low accuracy, redundant and incorrect values, non-uniform sampling rates, presence of outliers, data co-linearity & missing values, drift & offset and high & low frequency disturbances. When unprocessed flawed data is used for model development, the resulting control system may

lead to suboptimal operation, unsafe process operation, production of off-specification products or high operating cost (Morad *et al*., 2005). Therefore, the collected raw data is subjected to data preprocessing.

Data preprocessing and cleaning is done in order to avoid confusion resulting from availability of huge amount of data pertaining to many variables and better process understanding. Preprocessing of the process data extracted from the plant data base can become the key to the success or failure of the final application. The analysis of available data leads to better process understanding, process diagnostics and process improvement. For effective modeling the data must contain all the relevant information over the process operating range. Data preprocessing task is a complex procedure involving variable selection followed by data selection.

An industrial database provides data of all the variables that are recorded. However, all the available variable data may not be relevant to the process variable to be estimated. Presence of irrelevant (or less relevant) variable data in the input dataset leads to noise which may result in deterioration of the model. An inferential sensor works satisfactorily if only those secondary variables that are most sensitive to the primary variables are employed. The inappropriate selection of soft sensor inputs may lead to numerical problems, such as singularity and over-parameterization, or may markedly reduce the estimation accuracy (Zamprogna *et al*., 2005). For neural network modeling a reduction in the input data dimension leads to simplified neural architecture and reduced training time (Gonzaga *et al*., 2009).

Therefore, for better modeling results, after collecting the data pertaining to all the associated process variables from the plant database, the irrelevant variables should be deleted from the dataset which will otherwise lead to increase in noise level, thereby deteriorating the modeling results. Prior process knowledge is used to screen out first

totally irrelevant variables. Then statistical or heuristic techniques are performed to further eliminate irrelevant and redundant variables. Principal component analysis and clustering analysis are two typical methods involving statistical and heuristic technique, respectively (Papadokonstantakis *et al*., 2005). Another method of variable selection is Kohonen maps (Kohonen, 1990). They belong to the set of self organized maps which are used to project subsets of input variables along with the output variables onto network output space. A dissimilarity method is used to determine the relevance of each combination.

The task of variable selection is followed by data selection. This is a crucial step deciding the success of the soft sensor design. This essentially involves identification and removal of data outliers from the collected raw data. Outliers are sensor values which deviate from the typical or sometimes also meaningful ranges of the measured values. In other words outliers are observations that do not follow the statistical distribution of the bulk of the data. In the context of process industry, outliers in dataset may arise due to (Liu *et al*., 2004, Lin *et al*., 2007, Fortuna *et al*., 2007, Hodge & Austin 2004): hardware failure, process disturbances or changes in operating conditions, instrument degradation, transmission problems and/or human error.

Outliers can be distinguished into two types: obvious and non obvious. Obvious outliers are those whose values do not satisfy the physical and technological limitations (e.g a negative absolute pressure). Non obvious outliers even though satisfy the technological limitations, have values outside the typical range and hence are not true reflection of the correct variable state. From knowledge of process and hardware limitations, obvious outliers can be detected and removed with ease. The real difficulty arises in accurate detection of non obvious outliers where one may face the problems of masking and/or swamping. Masking refers to the condition when outliers are incorrectly identified as actual process values and swamping is the situation when a normal process

value is incorrectly identified as outliers (Lin *et al*., 2007). Outliers may lead to model misspecification, biased parameter estimation and incorrect analysis results (Liu *et al*., 2004). Moreover, bad data or outlier characterized by low signal to noise ratio leads to reduced resolution and may produce false alarms during process monitoring (Sharmin *et al*., 2006). Common approaches for outlier detection are based on statistical methods using the historical data. The methods can be of univariate approach or multivariate approach.

The commonly used univariate outlier detection techniques are the conventional methods such as three sigma edit rule, box plot method and robust methods such as mean minimum distance algorithm or the Hampel's identifier technique (Pearson, 2001; Lu *et al*., 2009). Univariate outlier detection techniques should not be applied to multivariate data because of the possibility of the existence of correlations between the variables. The samples of a single variable which may be detected as outliers univariately may not appear as outliers when the variability of other variables is considered (Hoo *et al*., 2002).

The common approaches of multivariate outlier detection can be classified as either distance based techniques or projection pursuit methods (Filzmoser *et al*., 2008). Projection pursuit technique involves suitable projection of the raw data in which outliers are likely to appear more prominently. Distance based techniques are based on computation of distance of a particular observation from the centroid of the data cloud. Subsequently, based on certain cut off criteria a particular observation is either flagged as an outlying observation or retained as a normal observation. Both techniques are in use and are areas of open research. While there is no universally accepted single technique for a particular problem, in this work, attention is focused on the application of distance based techniques for outlier detection. The various univariate and distance based multivariate outlier detection techniques used in this work are described in Chapter 3.

The outliers may appear as a block (e.g. if the plant is shut for some period of time, the process values appear close to zero) or single instances (occurs mostly due to sensor malfunction). If outliers are detected as a block continuously over a period of time all those values must be deleted from the dataset. However if the outliers are detected at particular discontinous time instances instead of over a time period, then some reasonable value should be placed in place of that outlying value. This is called as missing value imputation. While there are many imputation techniques available in the literature (Lakshminarayan *et al*., 1999), in this work the commonly used interpolation technique has been adopted. This involves linear interpolation between neighboring values if the outlier is detected inside a dataset or linear extrapolation with the preceding or following values if the outlier is detected at the end or beginning of a dataset respectively.

1.3.2 Model Development and Validation

The processed dataset is subsequently used for soft sensor model development. Prior to modeling, the available data is divided into a training set which is used for model development and a validation set. The model accuracy is assessed by investigating the model's prediction accuracy with the validation set data which is not used for model development. Proper training set selection can also play a vital role in successful model development. It has to be ensured that the training subset formed from the total dataset must be a proper representative of the entire dataset. For small datasets it may be possible to capture the characteristics of the entire set in the training set by manually picking appropriate samples. But manual selection is not possible for large sets. Therefore, the values are selected either randomly or some statistical algorithm can be used for sample selection. Most modeling works in engineering applications have used random selection method for obtaining the training set from the total available data and there have been only very limited industrial application of structured method of data splitting. In this work, in

addition to random selection, two statistical methods, the Kennard-Stone algorithm and the Duplex algorithm have been used for division of data into training and validation subsets. The detail description of these methods are presented in Chapter 3.

After obtaining the training and validation data, one is ready for soft sensor model development. This involves model structure selection followed by model identification. Model structure is a set of candidate models among which the required model is searched for. The model structure stage is concerned with finding a form for the soft-sensor model. These may be first principle models, regressor models, black box or gray box models. Whatever is the model structure, it always has a set of dependent variables (process outputs) which are the consequence of a set of independent variables (inputs).

The enormous complexity of the industrial process makes it difficult for development of models from first principles. However, the availability of great amount of historical data in the plant database encourages the use of gray or blackbox models. Selection is mainly decided by the purpose for which the soft sensor is designed. The model structure can range from simple statistical models like ordinary least square regression to more complicated soft computing based models such as neural network, fuzzy inference systems or combinations of more than one technique which is popularly called as gray box or hybrid modeling approach.

Model identification consists in determining a set of parameters e.g., the regression coefficients in a multiple linear regression model or weights and biases in a neural network model which will identify a particular selected model structure based on the available data. The set of parameters are determined using some optimization techniques so as to achieve the least modeling error. The modeling error is measured using some statistical parameters between the actual process output values and the model predicted output values. The model accuracy can be expressed in terms of correlation coefficient,

root mean squared error, mean absolute error etc. The details of different statistical performances used for model evaluation are presented in Chapter 5.

A model that fits the data used for model identification may give poor results on a new dataset. So the generalization capability of the designed model is evaluated using a different set of input – output data which is known as the model validation process. If validation fails then one or more of the previous steps are repeated. Model validation is a complex process and becomes more difficult in case of non linear models which is often the case with a soft sensor. The design constraints considered depend on the objective for which the soft sensor is intended to be used.

Finally, the model that gives good prediction accuracies for both the training dataset as well as the validation dataset is chosen to be used as a soft sensor for the industrial process concerned.

1.4 Research Objectives

The objectives of the present work are mentioned below:

1. Design of data-driven soft sensors for cement manufacturing processes using statistical (linear and non-linear regression) and artificial intelligence (neural network, fuzzy inference and adaptive neuro-fuzzy inference) based modeling techniques.
2. Comparison of the performances of the different data-driven soft sensors.
3. Development of control algorithms for cement manufacturing processes using the designed soft sensors.

1.5 Organization of the Thesis

An exhaustive literature survey on the different types of soft sensors reported till date for different industrial processes along with scope of the present work is presented in Chapter 2. Theoretical details of different methods used in this work starting from outlier

detection techniques to different statistical, black box and hybrid modeling techniques are presented in Chapter 3. Two industrial processes are taken up in this work for development of soft sensor: the pyroprocess in the cement industry involving the rotary cement kiln, for estimation of cement clinker qualities and the clinker grinding process for estimation of cement particle size coming from the cement mill. Chapter 4 gives the details of these two processes along with the associated process variables and the soft sensor development activity for these two processes. The results and discussion involving application of different univariate and multivariate outlier detection techniques, training set design algorithms and the prediction accuracies of different statistical, neural network, fuzzy inference and neuro-fuzzy based soft sensors are presented in Chapter 5. The concluding remarks and future scope are presented in Chapter 6.

Chapter - 2

Literature Review

This chapter begins with a detailed literature survey of various types of soft sensors reported till date in Section 2.1. Subsequently the gaps in existing research are presented in Section 2.2 followed by the scope of the present work in Section 2.3.

2.1 Soft Sensor Development in Various industries

In this section the various types of soft sensors reported in the literature have been presented. The soft sensors have been classified according to the types of industrial processes concerned. While bulk of the reported soft sensors have been developed from actual process input - output data, a few models are also based on mathematical model of the process.

It may be noted that the focus of the present work is on the design of data-driven soft sensors. Therefore, only data-driven soft sensor models based on black box and gray box modeling principles are reviewed in the following section and a few reported soft sensors derived from first principles, have not been presented.

2.1.1 Petroleum Refinery and Petrochemical Industries

Probably the early works on inferential measurement in distillation date back to the early 1990s when Kresta *et al*. (1994) applied the method of partial least squares (PLS) for estimation of heavy key component in the distillate. The latent variables used in the model were formed from various temperature and flow measurements.

Chen and Wang (1998) proposed soft sensor design method using a combination of Bayesian automatic classification system and back propagation neural network. The software sensor was applied for prediction of condensation temperature of light diesel oil

using an input set of 14 process variables comprising of different tray temperatures, column top temperature, reaction temperature, flow rates of feed, recycle, steam and rich absorbent oil.

Park and Han (2000) proposed a multivariate locally weighted regression technique for modeling of non linear processes with correlated process variables. The technique was used to estimate the bottom product composition and temperature of distillate. Toluene composition estimation was done using 16 process variables and estimation of 90% distilled diesel temperature was made using 57 process variables which were measured online.

Back Propagation Neural Network (BPNN) modeling technique was used by Bhartiya and Whiteley (2001) for online monitoring of ASTM 95% end point of kerosene. They used scatter plot, partial correlation coefficient and Mallow's *Cp* criterion to identify 5 potential key input variables from a set of 59 process variables.

Fortuna *et al*. (2003) investigated the performances of BPNN, Radial Basis Function Neural Network (RBFNN), neuro-fuzzy networks and non-linear least squares fitting models as soft sensors for a sulfur recovery unit in a refinery. The BPNN model trained by Levenberg-Marquardt algorithm produced the best performance and was used for real time monitoring of hydrogen sulfide and sulfur dioxide in the tail stream of the sulfur recovery unit using the information of five process input variables.

The techniques of standard support vector machine (SVM) and least square SVM based on statistical learning theory (SLT) were employed by Yan *et al*. (2004) to construct a soft sensing model. The SVM based soft sensor is applied to estimate the freezing point of light diesel oil in distillation column. The optimum regularization and kernel parameters of the SVM were determined using Bayesian evidence framework. The model uses five process variables as inputs which include vapor temperature on the tray, quantity of reflux

of the first intermediate section circulation, extraction temperature and reflux temperature of first intermediate section circulation.

Fortuna *et al*. (2005) proposed soft sensors for estimation of the stabilized gasoline (C5) content in the overheads and the butane (C4) content in the bottom flow of a debutanizer column. BPNN model trained by Levenberg-Marquardt algorithm was used as soft sensor. The soft sensors used column top and bottom temperatures, pressure, side temperature, top flow and reflux ratio as model inputs.

The method of principal component analysis (PCA) was applied by Zamprogna *et al*. (2005) to select the most suitable set of secondary process variables to be used as soft sensor inputs for prediction of the unmeasurable process variable. A regression process model is presented for online estimation of product composition from input temperature data.

Dam and Saraf (2006) suggested neural network architecture using genetic algorithm (GA) for on line property (specific gravity of side draw products, flash point and ASTM temperatures) estimation of crude fractionator products. The genetic algorithm was used to determine the most relevant set of input parameters. The different inputs considered for the ANN model development were: flow rates of feed, reflux, heavy naphtha, kerosene, light gas oil, heavy gas oil and steam; top and bottom temperatures; flash zone pressure and crude true boiling point.

Luo and Shao (2006) suggested use of neuro-fuzzy system based on rough set theory (RST) and genetic algorithm. Rough set was used to obtain the reductive fuzzy rule set and genetic algorithm was used to obtain the optimal discretisation of continuous attributes. The neuro-fuzzy approach based on RST and GA was employed to develop soft sensor for estimating the freezing point of light diesel oil in fluid catalytic cracking (FCC) unit. The soft sensor model used five secondary process variables as model inputs which

include light diesel fuel extracting temperature, gas temperature at 19th column tray, reflux flow rate, reflux extract temperature and the cycle return temperature at the first middle sidetrack in main fractionator.

Statistical approach using non-linear generalized ridge regression method was adopted by Yan (2008) for modeling of complex non linear systems. The technique was used to develop soft sensor for naphtha 95% cut point. He has shown that the performance of the ridge regression method is better than the other methods for modeling of non-linear systems such as line regressions, non-linear ordinary least square regression and non-linear traditional ridge regression. For prediction of the cut point, the soft sensor used the following process information as inputs: temperature and mass flow rate of the inlet feed, temperature and pressure at column top, naphtha mass flow rate, mass flow rate of the side-stripper products, the temperature at the flash zone, the mass flow rate of the steam input to the bottom of the column.

Kaneko *et al*. (2009) proposed soft sensing based on partial least square (PLS) technique for the distillation process for estimating bottom product concentration. The PLS model was updated and the abnormal data were classified using independent component analysis (ICA)-based fault detection. A total of 19 variables (flow rates of feed, reflux, reboiler, top and bottom product, feed, bottom and different tray temperatures and liquid level) were used for developing ICA-based fault detection model and 25 variables (by adding time lagged values of some variables) were used for developing the PLS model.

Ma *et al*. (2009) developed a soft sensing system for the o-xylene purification column. They used the regression method for model development with input variables selected in a stepwise manner based on partial *F*-test criterion. The soft sensor was applied for continuous monitoring of isopropyl benzene impurity of o-xylene distillation column

using seven tray temperatures and flow rates of 1,2,4- Trimethylbenzene and o-xylene as model inputs.

Wang *et al*. (2010) proposed dynamic PLS model for real time monitoring of ASTM 90% distillation temperature of the distillate. In addition to model development, the PLS regression technique was also used for multivariate outlier detection. Using the method of root-mean-square error of cross validation (RMSECV), they chose 14 process parameters as predictor variables from a pool of 35 variables.

Ge and Song (2010) proposed the technique of relevance vector machine (RVM) as an alternative to the largely used standard and least-square support vector machine techniques for non-linear soft sensor design. The soft sensor was applied to a sulfur recovery unit (SRU) for estimation of concentration of H_2S and SO_2 in the tail gas using 5 process variables (air flow, Mono Ethanol Amine flow and gas flow from sour water stripping process) as model inputs.

2.1.2 Polymer Industries

Ohshima and Tanigaki (2000) developed wave-net and extended kalman filter based process models for online monitoring and control of melt index of high density polyethylene. The models use concentrations of hydrogen, ethylene, propylene, butene, co-catalyst and operating temperature as inputs.

Rallo *et al*. (2002) designed soft sensor for prediction of melt index of six different Low density poly ethylene (LDPE) grades. They applied the technique of self organizing map (SOM) to decide on the set of most relevant input variables from a pool of 25 total process variables consisting of pressures, flow rates, temperatures of the cooling/heating streams of the reactor, etc. Subsequently, dynamic clustering technique was used for formation of training data. The training dataset was used for designing fuzzy ARTMAP and RBF neural network models of the LDPE process for melt index prediction.

Kim *et al*. (2005) underlined the shortcomings of mechanistic and empirical modeling technique for modeling of polypropylene process which involves complex catalytic reaction producing more than 100 different grades of products. Their soft sensor is based on a hybrid modeling approach which involves critical to quality (CTQ) based clustering in the first step for different reactors followed by PLS based soft sensing model for each reactor in the subsequent step. The models were developed from 86 operational variables comprising of 32 flowrates, 28 temperatures, 18 pressures and 8 tank levels.

Shi and Liu (2006) proposed a soft sensor based on PCA, radial basis function (RBF) and multi scale analysis (MSA) to predict the melt index of polypropelene using the knowledge of 9 input process variables (pressure, flow rate and temperature, etc.).

Sharmin *et al*. (2006) applied PLS technique to develop a soft sensor which can predict melt flow index using routinely measured process variables obtained from a low density poly-ethylene (LDPE) – ethylene vinyl acetate (EVA) plant autoclave reactors.

Mu *et al*. (2006) proposed recursive PLS based soft sensor for predicting the average crystal size of a purified terephthalic acid (PTA) purification process. The model used fourteen process variables as inputs which include the pressures and the liquid levels of the five crystallizers and four level set-points of the crystallizers.

Roy *et al*. (2006) developed various kinds of ANN models for prediction and optimization of polymer properties. They used the informations of 440 different types of engineering polymers to develop back propagation, recurrent, probabilistic and generalized regression neural network models for prediction of mechanical, thermal, magnetic, optical, electrical, environmental and deteriorative properties from the knowledge of microscopic, mesoscopic, and macroscopic polymer structures.

Liu (2007) applied principal component analysis (PCA) followed by fuzzy *c*-means (FCM) and fuzzy Takagi–Sugeno (FTS) modeling technique to decompose a non-

linear system to several linear sub systems. The technique was used to build a piecewise linear virtual sensor model for inferring melt index of polythene. 14 process variable informations were used by the model to predict the process output i.e. melt index. These input variables include ethylene flow, modifier flow, purge flow rate, temperatures at different locations and pressure maintained in the reactor.

Gonzaga *et al.* (2009) applied BPNN technique for soft sensing the viscosity of poly-ethylene terephthalate (PET). The ANN model uses eight input process variables of primary esterification and secondary esterification reactor temperatures, low polymerizer and high polymerizer temperatures, low polymerizer and high polymerizer pressures and flow rate into polymerizer.

Li and Liu (2011) designed RBF neural network model for melt index prediction in the polypropylene process. The parameters of the RBFNN model were determined by combination of particle swarm and simulated annealing optimization techniques. The process input variables used by the model are the process temperature, the pressure, the level of liquid, percentage of hydrogen in vapor phase, flow rates of propylene and catalyst to the reactor.

Kaneko and Funatsu (2013) proposed genetic algorithm based wavelength selection technique for variable selection followed by application of Support Vector Regression (SVR) modeling technique, for soft sensor development of polymerization process. Using various input process variables such as the temperature in the reactor, the pressure and concentrations of the monomer, co-monomer, and hydrogen, the soft sensor was used to predict melt flow rate.

Soft sensor for online prediction of melt index was also developed by Liu *et al*. (2013) for sequential-reactor-multi-grade polymerization process. A just-in-time sequential non-linear modeling technique was proposed and the technique was combined with the least

square support vector regression technique for design of soft sensor. Similarly, another statistical technique named independent component regression was applied by Ge and Song (2014) for prediction of melt index in polypropylene production process. In their application, it was observed that the technique of independent component regression performed better than the principal component and partial least square regression techniques.

2.1.3 Fermentation and Bioprocess Industries

Linko *et al*. (1997) applied back propagation technique to develop neural network-based soft sensors for online estimation of enzyme activity and biomass concentration in yeast lipase and fungal glucoamylase production. The neural network model used on-line measurable variables such as pH, agitation rate, oxygen uptake rate, carbon dioxide evolution rate, total consumed oxygen, total carbon dioxide produced and ammonia consumption as inputs.

Sotomayor *et al*. (2002) developed discrete extended kalman filter model for online monitoring of microbial oxygen uptake rate and oxygen transfer function in the activated sludge process from measurements of the dissolved oxygen (DO) concentration and the air injection flow rate as inputs.

Bogaerts and Wouwer (2003) have presented a review of software sensor design techniques and their application in bioprocesses. They subsequently (2004) proposed a parameter identification procedure based on a cost function combining a conventional prediction error criterion with a state estimation sensitivity measure. This parameter identification procedure is demonstrated in the context of bioprocess modeling and soft sensor design. The minimization of the proposed cost function enforces higher model sensitivities and better transfer of information from measured to unmeasured variables.

Arauzo *et al*. (2004) proposed a soft sensor and internal model control (IMC) for penicillin fermentation process. They have suggested use of FasArt and FasBack neuro-fuzzy systems. Fast learning and good multi-input multi-output (MIMO) identification encourages the use of FasArt and FasBack for developing adaptive controller and soft sensor. The soft sensor is applied for estimating biomass concentration and viscosity of broth in the penicillin process using the online measurable variables of agitation speed, pressure, temperature, acidity, dissolved oxygen and carbon dioxide production rate.

Desai *et al*. (2006) introduced SVR technique for soft sensing of bioprocess variables. The SVR model parameters were determined by the technique of sequential minimization optimization and the developed soft sensor was applied to fed-batch bio-processes for prediction of invertase and streptokinase concentrations.

Dai *et al*. (2006) proposed an artificial neural network (ANN) soft sensing method based on the assumed inherent sensor and its inversion concepts to estimate hard to measure crucial process variables. The method is based on the assumption of an inherent sensor subsystem in the process whose inputs are the process variables to be estimated and outputs are directly measurable variables. A global invertible condition is proposed which guarantees the existence of an inverse of such inherent sensor. The proposed ANN – inversion soft sensor was applied to real time estimation of the immeasurable variables in erythromycin fermentation process (mycelia concentration, sugar concentration and chemical potency) using various process inputs such as dextrin flow, aqua ammonia flow, propanol flow, water flow, oil flow and pH value.

Ochoa *et al*. (2007) proposed model for simultaneous saccharification-fermentation (SSF) process used in bioethanol industry. The model can be used as a part of the control algorithm, as a simulated plant or as a soft sensor. They developed an unstructured model and a structured model (cybernetic model). The cybernetic model represented the SSF

process more precisely than the unstructured model. However, the unstructured model is simpler, has less model parameters involved and is more suited for model based control or real time optimization applications. They suggest the use of Metropolis Monte Carlo method (stochastic based) over gradient based technique for model's parameter identification.

Liu *et al*. (2010) developed SVM and ANN soft sensor models for online estimation of biomass concentration in erythromycin fermentation process. A combination of GA and simulated annealing optimization techniques were used for optimal SVM parameter estimation. The SVM based soft sensors uses 11 process inputs comprising of dissolved oxygen tension, pH value, temperature, relative pressure, agitator rotate speed, the volume of dextrin, bean oil, propyl alcohol and water, the airflow rate and biomass concentration at previous sampling instant and was shown to perform better than neural network model based soft sensor.

The technique of genetic programming was applied by Sharma and Tambe (2014) for soft sensor development of two bioprocesses. In the process of extracellular production of lipase enzyme, the soft sensor was used to predict the time-dependent lipase activity. In bacterial production poly copolymers, the soft sensor was designed to predict the amount of accumulated polyhydroxyalkanoates.

2.1.4 Metallurgical Industries

Radhakrishnan and Mohamed (2000) developed a BPNN model for the blast furnace process of iron making. The model utilizes 33 process variables as inputs and produces estimated values of six hot metal quality (quantity, temperature, silicon, sulphur, manganese and carbon percentages) and nine slag quality (quantity, TiO_2, SiO_2, Al_2O_3, CaO, MgO, K_2O, FeO and Basicity) parameters. The model inputs included quantity and

compositions of sinter, iron ore, coke, lime stone, manganese ore and temperature, pressure and flow rate of hot air blast.

Tian *et al*. (2006) used genetic algorithm combined with back propagation neural network for the soft sensor model. The genetic algorithm optimizes the weight and bias values of the back propagation network. They used the model to predict molten steel temperature in ladle furnace refining process using the available informations on refining power consumption, the initial temperature, the ladle states, the absorbed and released heat of alloy and slag, the amount of argon blowing, the weight of molten steel and the refining time.

Kano and Nakagawa (2008) presented a brief review of the various methods of data based process monitoring, modeling and control techniques which find application in steel industry. They proposed a data-driven quality improvement (DDQI) method and applied to iron and steel processes. Statistical process model using a combination of PCA and linear discriminant analysis (LDA) was developed to predict surface flaws and internal defects of manufactured steel. The model was developed using 55 input variables which include contents of various additive elements in a steel making process, temperature and residence time in each heating zone in a hot rolling process and temperature at the exit of each stand in a hot rolling process.

Zeng and Gao (2009) investigated the effect of data preprocessing on the model performance. They applied the multivariate outlier detection techniques of Mahalanobis distance (MD), Resampling by half means (RHM) and Smallest half volume (SHV) to the blast furnace dataset and after outlier detection and removal, applied the techniques of subspace system identification and prediction error method for prediction of silicon content in hot metal produced in the blast furnace. The silicon content was predicted using seven process variables consisting of quantity, temperature and pressure of blast, quantity

of coal powder, quantity of coal powder, permeability index, coke ratio and descending speed of materials.

2.1.5 Emission Quality Monitoring

Qin *et al*. (1997) applied techniques of principal component analysis and neural network continuous online monitoring of NO_x content in the flue gas coming out of an industrial boiler. The monitoring is achieved using the information of 7 process variables as inputs which include: flow rates of air, steam, feedwater and fuel; economizer temperature; pressure values at stack and windbox.

Ikonen *et al*. (2000) developed NO_x monitoring soft sensor for a 25 megawatt (MW) semi-circulated industrial fluidized bed combustor by combining neural network and fuzzy logic which is known as distributed logic processor (DLP) model. The model requires the information of primary air flow, fuel flow, bed, free board and throat temperatures and flue-gas oxygen percentage as inputs.

Neural network modeling for soft sensor development has been proposed by Tronci *et al*. (2002) for online prediction of CO, NO_x and O_2 in the flue gas of a 4.8 MW power plant. Steam generation rate, excess and over-fire air percentage and reburning flow rate, temperatures of inlet air and outlet fume and oxygen in combustion chamber are supplied as inputs to the ANN model.

Marengo *et al*., (2006) have developed PLS, PCA and ANN models for prediction of pollutant emission in a cement plant. They have used Kohonen mapping for splitting the data to training and testing sets. The 19 model inputs considered in their work are various physical-chemical properties of the raw material and fuel and the models were developed for prediction of quanitity of NO_x, SO_x and dust released from the clinkerization process. They found that ANN model performed better than PCA and PLS models.

Zheng and Yu (2008) investigated the application of artificial neural network and support vector regression techniques for modeling the relationship between NO_x emission and plant for a 300 MW coal fired boiler. The operating parameters include 19 process variables consisting of primary and secondary air velocities, mill speed, boiler load and coal quality. The SVR model whose hyper-parameters were determined using ant colony optimization technique exhibited better prediction performance than the BPNN model trained with Levenberg–Marquardt algorithm.

Shakil *et al*. (2009) applied PCA for reduction of input data dimension followed by time delay ANN model development for NO_x and O_2 soft sensing in industrial boilers. The hybrid model uses the values of different temperatures in the boiler, air flow, mixed gas flow and air to fuel ratio as model inputs.

Zhou *et al*. (2012) developed SVR, BPNN and Generalized Regression Neural Network (GRNN) models for continuous online monitoring of NO_x emission in coal fired thermal power plant. The SVR model parameters determined using ant colony optimization technique showed superior performances to BPNN model trained by LM and gradient descent with momentum (GDM) algorithm and the GRNN model. The model uses 21 process conditions as inputs which include primary and secondary air velocities, mill speed, total air flow rate, O_2 content in the flue gas and boiler load.

2.1.6 Pulp and Paper Industries

Runkler *et al*. (2003) developed Mamdani and Takagi-Sugeno type fuzzy inference system for modeling of wood chip refiner in fiber board production process. The models were used for prediction of flexural strength and the water uptake of the final fiber board from the knowledge of relevant process input variables (steam pressure, refining speed and refining power). During model development, the available input dataset was used to form

the "*if*" part and the corresponding output dataset was used to form the "*then*" part of the rule base.

Dufour *et al*. (2005) designed BPNN model as soft sensor to predict unmeasured variations in the feedstocks of an industrial pulp digester. This is one of the few researches for industrial soft sensor development where attention was given for proper design of training set for model development

Galicia *et al*. (2011) proposed a reduced order dynamic PLS based soft sensor for real-time monitoring of Kappa number of pulp produced in the digester. The Kappa number is inferred using the process information for 16 secondary variables which include effective alkali, dissolved lignin, total dissolved solids, free liquor temperature of upper recirculation, lower recirculation and extraction flow.

2.1.7 Particle Size in Grinding Processes

One of the earliest works on soft sensor development for a grinding process was reported by Casali *et al*. (1998) which involves designing of an ARMAX model for estimation of percentage of material with +65 mesh-size in the product of the grinding process using a rod mill. The different types of ARMAX models developed take the past output measurements and past and present input measurements for inferring the present output. The various online measurements used by the model were fresh ore feed rate, rod mill water feed rate, rod mill power, sump water addition rate, sump pump speed, solid concentration and mill power.

Another soft sensing approach for ball mills and semi-autogenous grinding (SAG) mills (Herbst and Pate, 1999) makes use of simple differential equations for particle size estimation, followed by correction of this estimation from knowledge of the actual values received from the real sensor using an extended kalman filter. The model uses mass feed rates of coarse and fine, mean residence time, power draw and ore grindability. However,

these models assume the availability of a real sensor and can only be applied in case of momentary unavailability of the real sensor. A crucial problem in the comminution processes is that for many such processes no hardware sensor for online particle size monitoring is available.

Sbarbaro *et al*. (2008) proposed a NARMAX type model structure as soft sensor for a grinding process where the output i.e. particle size was described as a function of the easily measured input variables where the parameters of the model were determined using error projection and recursive least square algorithm. The developed soft sensor was applied for estimating percentage of +65 mesh size particles in the output of ball mill using cyclone feed density, fresh ore feed rate and pressure at the cyclone battery as model inputs.

Use of artificial neural network for soft sensing of particle size of nickel mine ores was reported by Ko and Shang (2011). The neural network model trained by Levenberg - Marquardt algorithm received calculated particle uniformity and initial estimate of particle size based on 2D image analysis as inputs and produced as output the improved estimate of ore particle size. In addition to back propagation neural network technique, methods of recurrent neural network and wavelet network were investigated for soft sensing of particle size in lead-zinc ore beneficiation process (Mitra and Ghivari, 2006).

2.1.8 Batch Processes

Lee and Park (1999) developed back propagation neural network soft sensor model of a sequentially operated batch reactor used in advanced waste water treatment for Biological Oxygen Demand (BOD) and nutrient removal. The neural network model was used for real time estimation of phosphate, nitrate and ammonium concentrations using pH, oxidation-reduction potential and dissolved oxygen as inputs.

Zamprogna *et al*. (2004) applied the multivariate regression technique of PLS to develop an empirical model of a batch distillation column operated at constant reflux ratio, for online estimation of instantaneous mole fraction of the light and intermediate components in the distillate stream and the mole fraction of the heavy component in the reboiler. In this soft sensing approach, the input dataset consisting of initial feed composition, boil-up rate and reflux rate was preprocessed using PCA followed by application of dynamic and multiple PLS regression method for model development. Another soft sensor reported by the same authors (Zamprogna *et al*., 2005) estimates the mole fraction of the light and intermediate components in the distillate stream The soft sensor models are based on PLS regression and ANN techniques and use tray temperature measurements as secondary variables. They proposed PCA based sensitivity analysis method to optimally decide the model inputs from the set of total available inputs.

Marjanovic *et al*. (2006) applied multi-way PLS (MPLS) technique for soft sensing of product quality in a batch reactor used for production of specialty chemicals. Cooling water flow, reactor temperature and pressure and hydrogen flow to the reactor were used as input informations to the model which predicts the concentration of reactant and product in the mixture at any time in the batch.

Gunther *et al*. (2009) proposed an evolving PLS modeling strategy with bias correction for soft sensing of product titer in the fed-batch bio-reactor used for production of recombinant protein.

Facco *et al*. (2009) applied PLS regression technique to design soft sensor for online estimation of resin quality (acidity number and viscosity) produced in an industrial batch polymerization process. Online data for twenty three process variables (temperatures, pressures and valve openings in the reactor) were utilized as secondary variables for development of the soft sensor model.

2.2 Gaps in Existing Research

The literature survey on the different types of soft sensors proposed so far has led to the identification of the following gaps.

2.2.1 Gaps in Data Preprocessing

It has been emphasized in Section 1.3.1 that data preprocessing plays a crucial role in the industrial data-driven soft sensor design. The most important aspect of data preprocessing is the proper detection and removal of outliers. The techniques for outlier removal can either be univariate or multivariate. Most of the industrial processes are multivariate in nature i.e. the output product quality is a function of many input variables. However, in most of the proposed soft sensors the outlier detection has been performed using univariate techniques which may not be appropriate when multiple number of process variables exist and there is existence of strong correlation among the variables.

2.2.2 Gaps in Design of Training Set

In any data based modelling approach, the total dataset must be split into a training set and a validation set. While the model is developed from the training set, the model performance is assessed by simulation of the developed model with the validation set inputs and comparing the model estimated output with the actual output values. Proper selection of the samples for training and validation is a crucial step in data based model development. In most of the reported studies, this subset selection has been performed in a random manner. While random selection is simple to perform, and has been widely followed by researchers for modeling, this method does not guarantee that the training set obtained is a proper representation of the entire dataset and moreover it may give rise to extrapolation problems (i.e. samples at the boundaries may not be included in the training set). This problem can be suitably addressed by following a structured subset selection

technique. Unfortunately, there has been little application of structured sampling methods in industrial processes modeling studies.

2.2.3 Gaps in Modeling Technique

There have been wide application of the different statistical methods ranging from simple multiple linear regression to more sophisticated techniques of principal component, partial least square and support vector regression and their improved versions. Similarly, the technique of black box modeling using artificial neural network (ANN) has also gained good popularity in the soft sensor design community. However there have only been very limited application of fuzzy inference and neuro-fuzzy inference techniques for soft sensor development.

2.2.4 Gaps in Application in Industrial Process

In addition to the process industries mentioned in Section 2.1, there is also scope of soft sensor development and application in other industries. While significant research in soft sensing has been done for some industries such as petroleum refinery and petrochemicals, polymer or bio-processes, very few research works have been reported in cement industries which is the main focus of the present work.

Based on the aforementioned gaps, the objectives are defined and presented in Section 1.4.

2.3 Scope of the Present Work

The present work focuses in the design of soft sensors for two processes involved in cement manufacturing:(1) Cement mill where clinker is ground to fine particles (2) Rotary cement kiln where the raw mixture is converted to clinker.

2.3.1 Soft Sensor for Cement Mill

Clinker grinding in the cement mill for production of cement is one of the most energy demanding processes in cement industry. In cement plants 75% of the total energy consumption is due to grinding of raw materials and clinker (Boulvin *et al* 2003). The

particle size in the cement manufacturing process is expressed as cement fineness or cement Blaine which is surface area per unit mass (m^2/kg). A higher Blaine indicates higher fineness. Lower than prescribed cement fineness indicates poor quality of the finished product resulting in more recycling and thereby increases the grinding cost further. Higher fineness has been reported to increase the compressive strength of cement (Binici *et al*, 2007). However, much higher fineness than required, will result in subsequent loss of cement particles in the bag house separation unit and loss of cement from cement bags during handling and transportation. Moreover, cement fineness has also been shown to indirectly affect the heat of hydration and settling time (Vuk *et al*., 2001).

The aforementioned facts highlight the need for production of cement with proper fineness. Presently, the fineness is measured by offline laboratory sampling at regular intervals. Since no hardware sensor is available for online monitoring of cement Blaine in the grinding process, a soft sensor based on the input parameters of the comminution process will largely help the plant operators to adjust the process inputs so as to produce cement with desired fineness. However, because of process complexity, mathematical modeling of a cement mill is a difficult task. The product particle size in a cement mill is a non linear function of the mill inputs.

A study of the existing literature for particle size soft sensing reveals that hardly any research work has been reported on soft sensing of cement particle size. Neural network based modeling and control strategies for product flow rate and mill load in the ball mill based cement grinding process was reported by Topalov and Kaynak (2004). But there is no effort for prediction of particle size in cement mills. Most of the reported works concern with modeling of ball mill. Traditionally, the closed circuit ball mill equipped with high efficiency separator has been the most commonly used clinker grinding equipment. In cement plants, while the vertical roller mill (VRM) has been in use for raw

mix and coal grinding, its application in clinker grinding is hardly a decade old. As per literature the first VRM used for clinker grinding was in the year 2002 (Simmons *et al*., 2005). Thereafter VRM is gradually becoming the standard for new grinding installations. VRMs offer certain advantages over ball mills as listed below (Simmons *et al* 2005; Sorrentino 2011):

- More energy efficient and consume approximately 40% less power than ball mills.
- More compact layout (drying, grinding and classification are carried out in a single equipment).
- More suitable for handling hot feed and larger size feed than ball mills.

While the older cement plants still continue to use ball mills, due to the aforementioned facts, VRMs are gradually becoming the standard for new grinding installations. However, very few or no research so far has been reported on modeling of grinding processes employing vertical roller mills for clinker grinding.

2.3.2 Soft Sensor for Rotary Cement Kiln

The quality of clinker plays the most important role in determining the quality of cement. Unfortunately there is no hardware sensor available for online sensing of clinker composition coming out of a rotary cement kiln. The clinker quality is determined by measuring its contents of free lime and other important components by offline laboratory analysis. Therefore, any reduction in clinker quality, as determined by offline laboratory analysis hours after production, leads to rejection or recycling of the clinker formed. Any method for online estimation of clinker quality will greatly help in reducing the amount of rejection thereby resulting in lower revenue loss or more profit. While there have been some modeling works reported for rotary cement kiln, very few models focus on soft sensing of clinker quality.

Marengo *et al*. (2006) developed model for prediction of pollutants from the clinkerization process. Stadler *et al*. (2011) developed model to determine the temperature profile of material along the kiln length. In the kiln model reported by Sadeghian and Fatehi (2011), the focus is on detection of various types of process faults associated with kiln operation, such as coating disintegration, ringing, super hot or super chilled. This is achieved by use of model for predicting the kiln back end temperature (entry point of raw material in the kiln). The different inputs considered for the modeling are: material flow, fuel flow, kiln speed, induced draft (ID) fan speed and secondary air pressure. Similarly Sharifi *et al*. (2012) have used the same input variables for predicting current, CO content, pre-heater temperature and back end temperature using wavelet-based fuzzy inference system. Unfortunately, none of the aforementioned models are suitable to be used as soft sensors for inferring clinker quality.

Probably the first ever effort for soft sensing of clinker quality was reported by Lin *et al*. (2007). Principal component regression and partial least square models were derived for prediction of clinker free lime content and NO_x emission from the kiln. For clinker free lime prediction, they used kiln current, kiln feed, fuel flow rates to calciner and kiln and several temperature measurements within the kiln system as the model inputs. However, one possible shortcoming in the reported model is ignoring the input quality. In general, for any process, the output quality is a function of the process operating conditions as well as the type of input. Therefore in addition to the kiln operating conditions, the quality of raw meal also affects the final clinker quality as shown later in this work. This aspect has not been considered in the proposed model.

Qiao *et al*. (2010) used least square support vector regression technique to model rotary kiln for clinker free lime estimation. They used kiln current, coal feed rate, kiln tail pressure, temperatures along kiln length and raw mill quality data as inputs. Similarly in

another soft sensor reported for cement kiln by Lin and Jørgensen (2011) the focus was on monitoring NO_x formation in the kiln with the help of partial least square regression technique using the process values of kiln current, kiln feed , fuel flow to kiln, several temperature measurements along with image features of kiln flame. Subsequently an RBF neural network model had been proposed by Zhong and Du (2012) for soft sensing of clinker free lime using the input variables kiln feed rate, fuel flow rate, kiln RPM, lime saturation factor (LSF), silica modulus (SiM) and alumina modulus (AlM).

While free lime (f-CaO) is the single most important quality parameter for cement clinker, a few other quality parameters also play a crucial role in determining the quality of clinker and cement. Effects of all important clinker quality parameters are briefly mentioned below (Hewlett, 2003):

- Free lime: Calcium Oxide (CaO) present in the cement clinker which has not combined with other oxides during the burning process in the kiln. An increase in clinker free lime means a reduction in total silicates ($C_3S + C_2S$). This ultimately results in less strength of the concrete.
- Alite (C_3S): This reacts quickly with water and is responsible for higher early strength of concrete
- Belite (C_2S): This reacts slowly with water and is responsible for imparting strength at later age
- Aluminate (C_3A): This reacts very fast with water and causes rapid hardening. Low proportions lead to more resistance to sulfates.
- Ferrite (C_4AF): This provides slight effect on strength and contributes to color of cement.
- Lime saturation factor (LSF): This is the ratio of CaO to iron, aluminium and silicon oxides. LSF of clinker is defined as:

$LSF = CaO/(2.8SiO_2 + 1.2Al_2O_3 + 0.65Fe_2O_3)$

High LSF in clinker indicates high proportion of alite to belite. This is usually maintained between 0.92 to 0.98 with values higher than 1 indicate possible presence of free lime in the clinker.

- Silica modulus (SiM): $SiM = SiO_2/(Al_2O_3 + Fe_2O_3)$.

  High SM indicates the presence of more amount of calcium silicates and less amount of aluminate and ferrite in clinker. Typical desired values of SiM are between 2 to 3.

- Alumina modulus (AlM): $AlM=(Al_2O_3/(Fe_2O_3)$. AlM indicates the relative proportions of aluminate and ferrite phase in the clinker. This is usually maintained between 1 and 4.

2.3.3 Outline of the Present Work

The data-driven soft sensors for the clinker grinding process and the rotary cement kiln were developed based on actual industrial data. Data for different process variables relevant to these two processes were collected from a cement plant with a clinker grinding capacity of 235 tons per hour and clinker producing capacity of 10000 tons per day. The collected data were preprocessed for detection and removal of outliers. Three types of univariate outlier detection techniques were applied to the grinding process which involves only a few (three) number of variables. On the contrary, multivariate outlier detection techniques were applied to the rotary kiln data where a number of (nine) process variables are involved.

After data preprocessing, the clean data obtained for the two processes were divided to training set for model development and validation set for model evaluation. Two popular training set design techniques i.e. the Kennard-Stone technique and the DUPLEX algorithm were used. The training sets obtained using the two algorithms were

compared with two types of random data division to explain the improvement in the quality of the training datasets obtained.

The training sets were used to develop the following models for the two processes.

Table 2.1: List of different data-driven models developed for the grinding process and clinkerization process

PROCESS	CLINKER GRINDING IN VERTICAL ROLLER MILL	CLINKER FORMATION IN ROTARY CEMENT KILN
Model Output	**Cement fineness or cement Blaine**	**8 number of clinker qualities**
Soft sensor models developed	Multiple linear regression (MLR) Quadratic response surface Standard SVR Least square SVR BPNN RBFNN GRNN PCA+BPNN Fuzzy inference (Mamdani) Fuzzy inference (Sugeno) Adaptive neuro fuzzy inference system (ANFIS)	Multiple linear regression (MLR) BPNN RBFNN GRNN Fuzzy inference (Mamdani) Fuzzy inference (Sugeno) PCA+BPNN

Performance of the developed models were evaluated based on multiple statistical model performance criteria.

The best models for each of the two processes were tested in the SIMULINK environment for their capability for real time estimation of the respective quality parameters.

Chapter - 3

Methodology

A brief description of the stepwise design procedure for soft sensor is presented in Chapter 1. In this chapter the theoretical details and application procedure of various design techniques are discussed. From the initial phase of data preprocessing to the final step of model development and validation, a number of methods exist in each design step. It is not possible to describe all the techniques in one report. Therefore, the description is limited only to the techniques which are used in the present research work.

3.1 Data Collection and Preprocessing

3.1.1 Variable Selection

Inappropriate selection of soft sensor inputs may reduce the estimation accuracy. Therefore, it is mentioned in Chapter 1 that an inferential sensor works satisfactorily if only the most relevant secondary variables are employed for soft sensor development. In this work, prior process knowledge (gained by rigorous consultation with plant operators and engineers) is used to screen out the irrelevant variables. In this way, three secondary variables were selected as model inputs for soft sensing of cement fineness and nine secondary variables were selected as inputs for soft sensing of clinker quality.

3.1.2 Outlier Identification

Large process industries generally collect and store data on sensitive process parameters which can be utilized for model development. There are sensors which record values for a process variable every few seconds or minutes. So, a dataset for even a period of a few months or some days comprises a vast number of data which often makes the analysis quite complicated. The data extracted from the plant database history cannot often

be used in its raw form because of problems of missing data, presence of outliers and data co-linearity. Data preprocessing and cleaning is done in order to avoid confusion resulting from availability of huge amount of data pertaining to many variables and better process understanding. Preprocessing of the process data extracted from the plant data base can become the key to the success or failure of the final application. The analysis of available data leads to better process understanding, process diagnostics and process improvement.

Outliers are sensor values which deviate from the typical or sometimes also meaningful, ranges of the measured values. In other words outliers are observations that do not follow the statistical distribution of the bulk of the data. In the context of process industry, outliers in dataset may arise due to (Liu *et al.,* 2004; Lin *et al.,* 2007; Fortuna *et al.,* 2007; Hodge 2004): hardware failure, process disturbances or changes in operating conditions, instrument degradation, transmission problems and/or human error.

Outliers can be distinguished into two types: obvious and non obvious. Obvious outliers are those whose values do not satisfy the physical and technological limitations (e.g a negative absolute pressure). Non obvious outliers even though satisfy the technological limitations, have values outside the typical range and hence are not true reflection of the correct variable state. From knowledge of process and hardware limitations obvious outliers can be detected and removed with ease. The real difficulty arises in accurate detection of non obvious outliers where one may face with the problems of masking and/or swamping. Masking refers to the condition when outliers are incorrectly identified as actual process values and swamping is the situation when a normal process value is incorrectly identified as outliers (Lin *et al*., 2007). Presence of outliers in the data used for modeling may lead to model misspecification, biased parameter estimation and incorrect analysis results (Liu *et al*., 2004). Moreover, bad data or outlier characterized by

low signal to noise ratio leads to reduced resolution and may produce false alarms during process monitoring (Sharmin *et al*., 2006).

The different outlier detection mechanisms may be classified as either univariate or multivariate based on whether dataset of only one variable is considered or a set of variables are considered simultaneously.

3.1.2.1 Univariate Outlier Detection

A critical review of commonly used statistical techniques and various nonlinear data cleaning filters used for univariate outlier detection is presented by Pearson (2002). A major drawback associated with data cleaning filters is that they even alter the values which are not outliers. Therefore, in this research, statistical methods for univariate outlier detection were applied which are presented below (Pearson, 2002; Hodge and Austin, 2004).

Three sigma method

Probably the most popular univariate approach is the 3σ outlier detection algorithm which is based on univariate observations of the variable distributions. This algorithm is presented below.

Let value of a particular variable be represented by x_i with i ranging from 1 to N where N is the total number of observations. For each observation x_i, calculate the value $|x_i - \bar{x}|$, where $\bar{x}$ is the mean value. If this value is less than three times the standard deviation (σ) of the entire set, then this is a normal value, else an outlier. Mathematically this is expressed as:

If $|x_i - \bar{x}| \geq 3\sigma$, then the particular sample is an outlier.

However, if multiple outliers occur on the same side of the mean then the value of $\bar{x}$ gets shifted towards the outliers. Moreover, presence of several outliers distant from the

mean results in an inflated estimation of σ. Therefore 3σ, edit rule is likely to detect less outliers as their number increases.

Hampel Identifier

The 3σ edit rule is based on the assumption that the variable under study is normally distributed and hence fails when the normality condition is not satisfied. More robust version of this approach is the Hampel identifier. This method uses more outlier resistant median in place of mean and median absolute deviation from median (MAD) in place of standard deviation to calculate the limits. The MAD scale is defined as

$$MAD = (1.4826)median\{|x_i - x_{0.5}|\} \quad (3.1)$$

The MAD scale represents a normalized version of the median distance of each data point x_i from the reference value $x_{0.5}$. $x_{0.5}$ corresponds to 50% quantile or median. The factor 1.4826 makes the expected value of MAD scale equal to standard deviation σ for Gaussian data sequences. Mathematically, the Hampel's identifier can be written as

If $|x_i - x_{0.5}| \geq 3MAD$, then the particular sample is an outlier.

Box Plot

Another outlier detection method is the box plot which is a graphical tool for determining mild and extreme outliers in a dataset. A box plot is drawn between the upper and lower quartiles with a solid line drawn across the box to locate the median. The different regions in the plot are defined as:

Lower inner fence: $Q_{0.25} - 1.5\times(Q_{0.75} - Q_{0.25})$

Upper inner fence: $Q_{0.75} + 1.5\times(Q_{0.75} - Q_{0.25})$

Lower outer fence: $Q_{0.25} - 3\times(Q_{0.75} - Q_{0.25})$

Upper outer fence: $Q_{0.75} + 3\times(Q_{0.75} - Q_{0.25})$

A mild outlier is a point beyond an inner fence on either side while an extreme outlier is a point beyond an outlier fence.

3.1.2.2 Multivariate Outlier Detection

An industrial dataset contains informations for multiple number of process variables out of which many are likely to be correlated. Univariate outlier detection techniques are largely ineffective while treating a multivariate dataset. Since multivariate outliers may not be extreme values along a particular direction, univariate techniques applied to multivariate dataset may result in masking or swamping.

The common approaches of multivariate outlier detection can be classified as either distance based techniques or projection pursuit methods (Filzmoser *et al*., 2008). Projection pursuit technique involves suitable projection of the raw data in which outliers are likely to appear more prominently. Distance based techniques are based on computation of distance of a particular observation from the centroid of the data cloud. Subsequently, based on certain cut off criteria, a particular observation is either flagged as an outlying observation or retained as a normal observation. There is no universally accepted single technique for a particular problem and both techniques are in use and are areas of open research. In this work, attention is focused on the more commonly used distance based techniques. The two classical multivariate distance estimators are the Euclidean distance (ED) and the Mahalanobis distance (MD).

The MD is better than the ED since this also takes into account the correlation existing among the variables. Therefore, in the early days the classical Mahalanobis distance was used as the multivariate outlier detection technique. The classical MD and leverage value methods work well with a few number of outliers. However, if a significant number of outliers are present on one side of the centre, the mean and covariances are substantially affected. Therefore, use of classical methods to such a dataset may result in masking and swamping effect. This necessitates the need for robust estimation of the mean

and covariance matrix which is the most important challenge associated with the application of distance based methods.

Probably one of the earliest methods proposed for robust estimation of location and dispersion is the ellipsoidal multivariate trimming (MVT) technique (Gnanadesikan and Kettenring, 1972). This involves removing a certain fraction of observations with highest MD values (computed by classical means) followed by computation of the location and scatter (i.e mean and covariance) from the remaining observations. The algorithm stops when convergence is achieved for the values of mean and covariance obtained for two subsequent subsets. However, the MVT has been shown as unreliable due to the use of conventional MD in the initial step (Egan and Morgan, 1998) and also the breakdown point of MVT for *M* variables is only 1/*M* (Daszykowski *et al*, 2007). Breakdown point is defined as the percentage of outliers required in the total dataset to render a particular technique ineffective.

The MVT technique proposed in 1970s was followed by the techniques of minimum volume ellipsoid (MVE) and the minimum covariance determinant (MCD) methods in the 1980s (Rousseeuw and Leroy, 1987). These two techniques have been so far quite popular robust methods for multivariate centre and scatter estimation. The MVE technique involves construction of an ellipsoid of minimum volume by taking half of the observations. The robust estimates of mean vector and the covariance matrix are computed from these *N*/2 observations. The MCD method involves determination of the *N*/2 observations from the total data whose variance-covariance matrix has the minimum determinant value. While MVE and MCD techniques have the advantage of having a high breakdown point (50%), these are computationally complex. For example in the traditional MVE technique, for an N×M dataset, volumes of $\frac{N!}{h!(N-h)!}$ (h is the integer part of

(N+1)/2) number of ellipses are to be computed in order to determine the minimum volume (Hadi, 1992). Various techniques for estimation of the minimum volume ellipsoid have been proposed such as resampling algorithm (Rousseeuw and Van Zomeren, 1990), simulated annealing and genetic algorithm (Woodruff and Rocke, 1993) and forward search algorithm (Atkinson, 1994; Riani *et al*, 2009).

Since MVE and MCD techniques are iterative in nature, successful application of these methods require complex algorithms which may be difficult to program. Moreover, algorithms which are iterative in nature are not suitable for automation and hence cannot be implemented online in the plant for detection of outlying observations. Therefore, some simplified methods for robust estimation of multivariate location and spread have been proposed which include resampling by half means (RHM), smallest half volume (SHV) (Egan and Morgan, 1998) and closest distance to centre (CDC) method (Chiang *et al*, 2003). Reviews on the various outlier detection techniques, issues on robust estimation of multivariate location and dispersion can be found in Hodge & Austin (2004) and Daszykowski *et al*., (2007). In this research work, robust outlier detection has been performed using two types of classical methods and three types of robust techniques. The details of these techniques are given below.

The classical Mahalanobis distance method

In this technique the MD of each observation from the data centroid is calculated and compared with a cut off value. The MD is described by the following equation for a dataset X with N observations and M number of variables (X is an $N \times M$ matrix):

$\overline{X}$ is the 1×M row vector representing the mean values of all the M variables.

$$MD^2 = diag\left(\left(X - \overline{X}\right)C^{-1}\left(\mathrm{X} - \overline{X}\right)^T\right) \tag{3.2}$$

Here C is the covariance matrix which is defined as:

$$C = \frac{\boldsymbol{X^T X}}{N-1} \tag{3.3}$$

$\left(X - \overline{X}\right)$ represents the $N \times M$ mean centered input data matrix obtained by subtracting from any element of X, the mean value of that particular variable.

$diag$ represents the diagonal elements of the resulting $N \times N$ matrix. Therefore MD^2 is the $N \times 1$ column vector representing the squared Mahalanobis distance of all N number of observations.

Observations whose MD are higher than the cut off value are considered as outliers. Usually this cut off value is set at 97.5% quantile of the chi-square distribution. This can mathematically be expressed as follows:

If (multivariate distance)$^2 > \chi^2_{M,0.975}$ then the observation is an outlier.

If (multivariate distance)$^2 \leq \chi^2_{M,0.975}$ then the observation is a normal value.

Hat matrix leverage value method

Similar to conventional MD method, another classical method for multivariate outlier detection is the hat matrix leverage value (Hoaglin and Welsch, 1978). In this method, from the available input dataset, the hat matrix is determined as mentioned below:

$$H = X(X^T X)^{-1} X^T \tag{3.4}$$

Here X is the $N \times M$ input data matrix and H is the hat matrix. The diagonal elements of the hat matrix are called as leverage values and if these values are greater than 2M/N, then the corresponding observations are flagged as outliers.

The Minimum covariance determinant (MCD)

In this method, subsets of data (of size $\frac{N}{2} \times M$) are selected. For each subset the covariance matrix is computed and the determinant value of the covariance matrix (which is a square matrix) is calculated. The subset which results in the minimum value of the

determinant of the covariance matrix, is the one representing most normal $\frac{N}{2}$ samples. The mean vector $\overline{X}$ and the covariance matrix C is determined based on this $N/2$ samples instead of the entire N observations.

The MCD technique is iterative in nature, because to determine the best half representative samples, all possible subsets must be considered. The method of MCD was implemented using the code given by Verboven and Hubert (2005).

The Smallest Half Volume (SHV) method

In this method the Euclidean distance between every pair of samples are determined. In this manner a distance matrix D of dimension $N \times N$ is formed. Any element D_{ij} of this matrix represents the Euclidean distance between i^{th} and j^{th} sample given by

$$D_{ij} = \sqrt{\sum \left(X_i - X_j\right)^2} \tag{3.5}$$

It may be noted that the diagonal elements of the distance matrix which represent distance of an observation from itself are zero.

Subsequently, elements in each column are sorted in increasing order. Then for each column (of length N) the first $N/2$ elements are summed. The column with smallest sum is identified and the $N/2$ elements of that column resulting in this smallest number represents the $N/2$ most similar observations. Thereafter, the mean and covariance matrix required for MD calculation is determined based on this set of $N/2$ observations.

Closest distance to center (CDC) method

This method involves determination of Euclidean distance of each sample X_i ($1 \times M$ row vector) from the centroid $\overline{X}$ ($1 \times M$ row vector). So, for N number of samples an $N \times 1$ column vector representing the distances of samples from the centre is obtained. In this column vector $N/2$ samples with smallest distance to the centroid are identified. These $N/2$ samples which are closest to the center represent the normal observations since outliers are

observations which lie far away from the majority of the data. Subsequently, the mean vector and covariance matrix are determined based on this set of observations.

3.1.3 Missing Value Imputation

Missing data are single sample values or set of sample values, where one or more measurements have a value which does not represent the actual state of the physical measurable quantity. The affected variables usually have values like $\pm\infty$, 0 or any other constant value.

The most common causes are the failure of a hardware sensor, its maintenance or removal, failure of proper transmission of the data between the sensors and the database, errors in the database, problems in accessing the database, etc. Besides these, missing values may also result when the values are entered manually into a log book. Based on the different conditions resulting in missing data, they have been classified as (Lakshminarayan *et al*., 1999, Khatibisepehr and Huang, 2008):

Missing completely at random (MCAR): When the probability of a record containing a missing value does not depend on either the observed data or the missing data.

Missing at random (MAR): When the probability of a record having a missing value could depend on the observed data but not on the value of missing data itself.

Not missing at random (NMAR) or Non ignorable: When the probability of a record having a missing value for an attribute could depend on the value of the attribute.

In the present work, after the outliers are detected and removed from the dataset, it results in missing data at those time instants.

There are different strategies to deal with the problem of missing values in the dataset. One approach to addressing the issue of missing data is case deletion i.e. to skip the data samples consisting of variable or variables with the missing values. However, this may lead to deletion of some of the useful data as well, resulting in loss of useful information.

Case deletion is suitable when the amount of missing data is small and constitutes a negligible fraction of the total data (Qin *et al*., 2007, Lakshminarayan *et al*., 1999).

Data imputation is a statistical procedure for filling in of missing values by practical possible values so as to make the database complete. One can distinguish different approaches for carrying out missing data imputation. These are: (i) single imputation (ii) multiple imputation and (iii) simultaneous imputation.

In single imputation the missing values are replaced by a single value (using e.g. mean/median values). Multiple imputations are iterative techniques where several imputation steps are performed and multiple choices are there (in decreasing order of likelihood) for each missing value. Multiple imputations involve more rigorous computation procedure. Simultaneous imputation is the technique of filling in of missing values for multiple values in the database simultaneously.

The choice of a particular imputation technique to be used depends mainly on the nature and amount of available data, extent of missing data and the application purpose. Mean imputation is a commonly applied strategy in practical scenarios where the missing values are replaced by the mean of all available observations of the affected variable. This method may significantly change the values of variance, correlations and regression coefficients of the imputed variable. Use of mode value instead of mean value is an alternative to mean imputation. When the measurements are expected to be constant over a period of time then the missing data can be replaced by the last measured observation before the missing data. Another approach is to impute the missing value by making a linear interpolation of the values preceding and following it (Kano and Nakagawa 2008; Wang *et al*., 2010).

In this work, linear interpolation between neighboring values or linear extrapolation with the preceding or following values is used for the imputation of missing values as explained below.

If in a dataset, values of x_1, x_2, x_4, x_5 are known at time instants t_1, t_2, t_4, t_5 and the value x_3 obtained at t_3 was identified as an outlier and hence deleted from the dataset, then some meaningful value of x_3 is computed by linear interpolation between x_1, x_2 and x_4, x_5.

Similarly, if the missing value due to deletion of outlier resulted in the beginning of a dataset, imputation is performed by linear extrapolation with the following values. If the missing value occurred in the end of a dataset, imputation is performed by linear extrapolation with the preceding values.

3.1.4 Data Scaling and Normalization

Before modeling, it is necessary that the data of variables be normalized relative to one another. Otherwise, important process variables having small magnitudes will be overshadowed by less important variables having larger magnitudes. Therefore, data scaling or normalization is done so that all variables will have more or less equal weights. Some methods followed for data normalization are mentioned below.

Min – max normalization (Fortuna *et al*., 2007):

$$x_{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}} (x_{normmax} - x_{normmin}) + x_{normmin} \quad (3.6)$$

Z-score normalization or auto scaling (Sharmin *et al*., 2006, Fortuna *et al*., 2007) :

Here, each variable is normalized about its mean. $x_{norm} = \frac{x_i - \overline{x}}{\sigma}$ (3.7)

Zero – mean normalization method (Lee *et al*, 2007):

$$x_{norm} = \frac{x_i - \overline{x}}{R_{max}} \text{ where } R_{max} = max[x_{max} - \overline{x},\ \overline{x} - x_{min}] \quad (3.8)$$

In this work, a commonly used normalization (Radhakrishnan 2000, Yan *et al*., 2004) in the range of 0 and 1 is adopted:

$$x_{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{3.9}$$

3.2 Data Co-linearity and Dimensionality Reduction

If the two or more input variables are related to each other, then these are known as correlated variables and the data of these variables are said to be co-linear. A challenging issue for soft sensor design, is co-linearity existing in the available data of the process variables. Very often, the variables measured in the process industry are strongly co-linear and at times redundant. Data pertaining to variables other than the ones of importance are unnecessary and increase the model complexity, which has often negative effect on the model design and subsequently model performance. Moreover, highly correlated variables can give numerical problems for multiple regression based models (Lin *et al*., 2007).

One way to address the co-linearity problem is to select a subset of the input variables which is less co-linear. This is done by analysis of scatter plots between pairs of variables. A scatter plot is a statistical tool for exploring variable correlation. In this method each candidate variable is plotted against the system output to search for any structure. If the variables move together on the plot then they are correlated. A straight line indicates a linear input – output correlation while a curve indicates a non-linear input – output correlation. The relationship is characterized either by using Pearson correlation coefficient or coefficient of variation

$$\text{The Pearson correlation coefficient} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \tag{3.10}$$

$$\text{Coefficient of variation} = {Standard\ Deviation}/{Mean}$$

Another method is to perform cluster analysis. Cluster analysis is an exploratory data analysis methodology in which a set of observations is partitioned into a certain number of unknown groups or clusters in such a way that all observations within a group are similar, while observations in different groups are different (Chen and Wang 1998). The rows of the data matrix which represent samples, are partitioned into distinct groups (clusters). Cluster analysis is performed by constructing a proximity matrix (proximity measures the nearness of two objects).

The method of correlation analysis is performed by computing the estimated normalized correlation function between each possible independent variable and the system output. The magnitude of any peak in the cross correlation function gives information about the relevance of the input variable while its position gives information about the correct regressor to be considered in the model.

Principal Component Analysis (PCA) and Partial Least Square (PLS) regression are among the most popular methods to deal with data co-linearity in the process industry. PCA and PLS are useful multivariate statistical tools widely applied in soft sensor design. They are applied in a number of steps ranging from outlier detection (data processing) and data rectification to dimensionality reduction, model identification and process monitoring (Wang *et al*., 2002). While PCA is applied only on the input dataset, PLS is a regression technique applied to both the input and output datasets. In this work, PCA was applied to both the grinding process and rotary kiln process data to investigate the correlation existing among the variables.

3.2.1 Principal Component Analysis (PCA)

The method of principal component analysis aims at reducing the dimensionality of a multidimensional dataset consisting of a large number of interrelated variables. PCA is used to convert a large number of interrelated variables to less number of uncorrelated

variables. This reduction is achieved by transforming to a new set of variables, the principal components, which are uncorrelated. Each variable is transformed so that it has a mean zero and a standard deviation 1 (and so the variance is also 1). So the total variance in the dataset, which is the sum of variances of the observed variables is always equal to the number of observed variables analyzed. The transformed variables are linear combinations of original ones and the last variables can be ignored with minimum loss of information (Joliffe, 2005; Fortuna *et al*., 2007).

In PCA, the input data matrix X is represented as:

$$X = TP^T + E \tag{3.11}$$

Where P is the principal component loading matrix and T is the score matrix. The loading vectors are the eigen vectors of the covariance (or correlation) matrix X^TX. The number of principal components can be determined by considering the eigen values associated with the loading vectors.

The calculation steps of PCA are presented in the form of a block diagram in Figure 3.1 (Wold *et al*., 1987; Jolliffe, 2005).

In step 2, standardization corresponds to calculation of $\frac{X-\overline{X}}{\sigma}$.

$\frac{X-\overline{X}}{\sigma}$ has also dimension of $N \times M$

Here all values are mean centered and have unit variance.

Scaling of the data by dividing the mean centered values with standard deviation is essential because PCA is a least square method and in the absence of scaling, variables with large variances will have large loadings (Wold *et al*, 1987).

The M columns of C' represent the M eigen vectors.

Eigen vectors are called loadings and identify the directions where the majority of the data variability occurs while the corresponding eigen values give the amount of variability

associated with each direction. This gives the components in order of significance. Let C' be the matrix containing the eigen vectors sorted in decreasing order of eigen values.

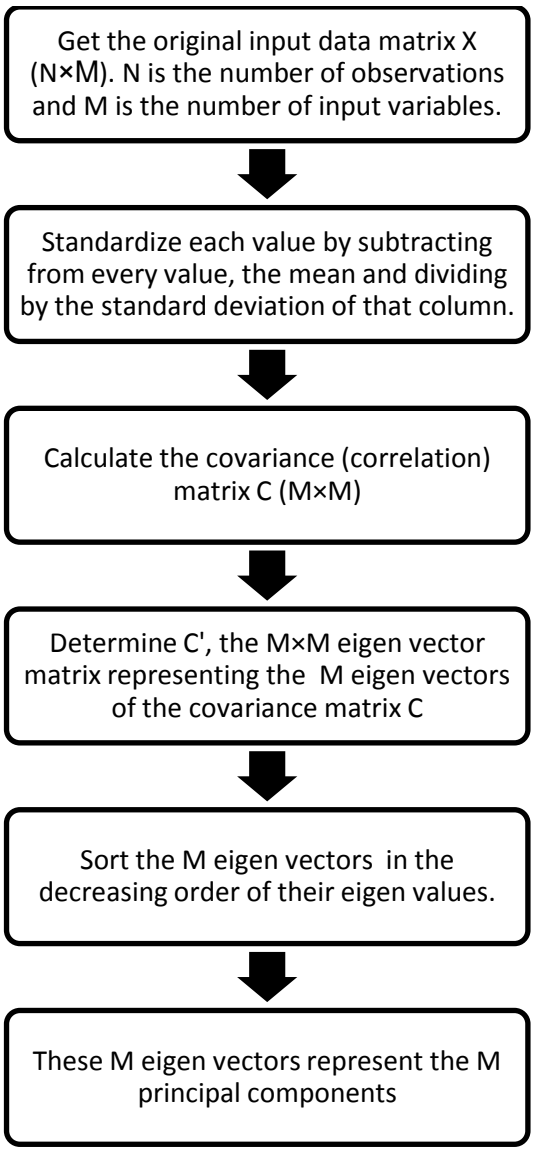


Figure 3.1: Procedure for conducting Principal Component Analysis (PCA)

$$C' = \begin{bmatrix} c_{11} & \cdots & c_{1M} \\ \vdots & \ddots & \vdots \\ c_{M1} & \cdots & c_{MM} \end{bmatrix} \tag{3.12}$$

$$\text{The input data } X = \begin{bmatrix} x_{11} & \cdots & x_{1M} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{NM} \end{bmatrix} \tag{3.13}$$

For M number of input variables an equal number of principal components are obtained which are linear combinations of the original variables.

The transformed data matrix $PC = \begin{bmatrix} PC_{11} & \cdots & PC_{1M} \\ \vdots & \ddots & \vdots \\ PC_{N1} & \cdots & PC_{NM} \end{bmatrix}$ (3.14)

The derived latent vectors or principal components are determined by multiplication of the scaled input data with the eigen vectors as explained below in Equation 3.15:

$$PC_{11} = c_{11} \times x_{11} + c_{21} \times x_{12} + \cdots + c_{M1} \times x_{1M}$$

$$\vdots$$

$$PC_{1M} = c_{1M} \times x_{11} + c_{2M} \times x_{12} + \cdots + c_{MM} \times x_{1M} \quad (3.15)$$

Similarly the second row of transformed data are determined by carrying out the above computation with the second row of input data i.e $x_{21}, x_{22}, \cdots x_{2M}$

The first principal component extracted in a PCA accounts for a maximum amount of total variance in the observed variables. The second component accounts for a maximum amount of variance in the data not accounted for by the first component. In this manner each component accounts for a maximum amount of variance in the observed variable that was not accounted for by the preceding component and therefore is uncorrelated with all the preceding components. PCA proceeds with each new component accounting for progressively smaller amount of variance.

If there is significant correlation existing among some of the input variables, the last few PCs will account for very negligible amount of total data variance and hence can be ignored without much loss of information. Therefore, finally *M* number of input variables can be reduced to P number of principal components where *P*<*M*. The number of principal components *P* is determined by constructing a table of principal components versus cumulative variance or a scree plot (principal components vs variance) and P is decided to be the value upto which the total accounted variance is around 85% to 90 % or more. The initial *N*×*M* original data values are finally reduced to a modified input dataset of dimension *N*×*P*.

3.3 Design of Training Set

Proper selection of the samples for training and validation is a crucial step in data-based model development (Rajer-Kanduč *et al*., 2003) and effective modeling requires the training set to be a proper representative of the entire dataset. One approach of dividing the dataset into training and validation subsets is random selection. Random selection is simple to perform and for small datasets it may be possible to preserve the characteristics of the total dataset in the training subset. However, for large datasets, method of random selection does not guarantee that the training set obtained is an appropriate reflexion of the entire dataset and moreover it may give rise to extrapolation problems (i.e. samples at the boundaries may not be included in the training set). This problem can be addressed by following a structured subset selection technique. While structured sampling methods have been quite popular for modeling in the field of chemometrics, most of the industrial processes modeling studies have been based on random selection so far and the method of data splitting has not received the due attention. There have been only limited applications (Dufour *et al*., 2005) for soft sensor development of industrial processes.

In the literature, the method proposed by Kennard and Stone (1969) has been reported to perform better than other subset selection techniques of Kohenen self organizing maps and *D* optimal design (De Groot *et al*., 1999, Rajer-Kanduč *et al*., 2003). Therefore, in this work, the performance of Kennard-Stone algorithm was compared with that of random selection and another popular statistical method known as the DUPLEX algorithm. The code for implementation of the two subset selection techniques were adopted from freely available TOMCAT toolbox (Daszykowski *et al*., 2007).

3.3.1 Random Selection

For random selection 79 random values (corresponding to sample numbers) were chosen in the range of 1 to 158 using 'Multiplicative lagged Fibonacci generator' algorithm available in MATLAB.

3.3.2 Kennard-Stone Algorithm

The steps involved in Kennard-Stone algorithm are outlined below (Kennard and Stone, 1969):

- Select the two most distant objects by measuring the Euclidean distance between every two samples.
- For the remaining samples, choose the ones that have the shortest Euclidean distances from the two chosen samples and select the one with the maximum distance. Include this sample in the training set.
- Repeat the procedure till the desired number of training samples are selected.

This algorithm results in sample selection from all parts of the data space for non-homogeneous distributions (Feudale *et al*., 2002).

3.3.3 DUPLEX Algorithm

The steps involved in DUPLEX algorithm are outlined below (Snee, 1977).

- Select the two most distant objects by measuring the Euclidean distance between every two samples and include in the training set.
- In the remaining data, include the pair of objects with maximum distance in the validation set.
- Identify the object with maximum distance from the two points of training set, include in the training set.
- Identify the object with maximum distance from the two points of validation set, include in the validation set.

- Proceed till all the objects of the original set are assigned to either the training or the validation set.

3.4 Statistical Modeling

After the initial activities of data cleaning followed by separation of the total dataset to training and validation datasets, the next step for soft sensor designing is development of data-based process models. The data-based process models may either be statistical or be based on the methods of soft computing. The following statistical models were developed in this research work.

3.4.1 Linear Regression Model

In linear regression models, the output is modeled as a linear combination of (not necessarily linear) functions of the inputs. The generalized form of such a model takes the form:

$$\hat{y} = \beta_0 + \beta_1 f_1(x_1) + \beta_2 f_2(x_2) + \cdots + \beta_M f_M(x_M) \tag{3.16}$$

Depending on the nature of the function f the linear regression models can be simple linear additive models or more complex polynomial models. In this work, linear additive, principal component regression and quadratic response surface models were developed for the two processes.

The structure of linear additive model for a single output and three input system is expressed as:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \tag{3.17}$$

The structure of a quadratic response surface model for the same process is given as:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 {x_1}^2 + \beta_5 {x_2}^2 + \beta_6 {x_3}^2 + \beta_7 x_1 x_2 + \beta_8 x_2 x_3 + \beta_9 x_3 x_1 + \beta_{10} x_1 x_2 x_3 \tag{3.18}$$

The structure of the principal component regression model is exactly the same as linear additive model except the fact that the input variables are the chosen principal components instead of the actual process variables.

The parameters of the regression models i.e. the regression coefficients were determined using the method of least sum of squared error criterion. As per this criterion the coefficients are evaluated as:

$$\beta = (X^T X)^{-1} X^T Y \tag{3.19}$$

Where β is the row vector consisting of the regression coefficients β.

It may be noted that if a constant or bias term is added in the equations mentioned above then a column of 1s must be added in the beginning of the input data matrix. That means if a dataset has values for M number of variables and N number of observations, then the dimension of the input data matrix X is $N\times(M+1)$ instead of $N\times M$.

3.4.2 Support Vector Regression Model

The support vector machine (SVM) technique, which was initially proposed in the mid 1990s for carrying out classification tasks (Cortes and Vapnik, 1995) was soon extended to solve regression or function approximation problems (Vapnik *et al*., 1997). The SVM technique used for non linear regression is known as support vector regression (SVR).

The SVR is based on structural risk minimization unlike empirical risk minimization (involves minimization of certain model performance index such as mean of absolute error or root mean squared error) commonly used for other statistical or ANN based modeling. The structural risk minimization principle involves determination of a function $f(x)$ that has at most ε deviations from target values for all sets of training data. The loss function parameter ε represents the radius of a tube around the regression function. This tube is known as error insensitive zone.

The non-linear mapping is done by non-linear transformation of the inputs to a high dimensional feature space (similar to input to hidden layer projection in feed-forward neural networks) followed by linear correlation of feature space terms with the output. A complete presentation of SVR theory and associated derivations can be found in books written by Vapnik (1998) and Cristianini & Shawe-Taylor (2000). Only the important modeling equations will be outlined here.

SVR is based on estimation of the function $f(x) = w\varphi(x) + b$ (3.20)

Here, $\varphi(x)$ represents the non linear transformation of the input vector to high dimensional feature space satisfying the Mercer's condition (Mercer, 1909), $\varphi(x_i)^T\varphi(x_j) = K(x_i^T x_j)$ where K is the kernel function, w is the weight vector and b is bias value.

The various types of valid kernel functions (satisfying Mercer's condition) are linear, polynomial, Gaussian radial basis function (RBF), exponential RBF, sigmoid and splines (Gunn, 1998). The commonly used Gaussian RBF kernel has the form:

$$K(x_i, x_j) = exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (3.21)$$

$x_i \; and \; x_j$ are ith and jth input training value and σ is the width of the RBF kernel. Based on the kind of optimization function used, the two types of SVR methods in use are known as the standard SVR and the least-square SVR (LS-SVR). In this work, both standard and least square SVR models were developed for the clinker grinding process. The theoretical details of the two methods are presented below:

3.4.2.1 The Standard SVR

The standard SVR technique uses error insensitive loss function and inequality constraints. The primal optimization function in standard SVR method is given as:

Minimize $J(w, \xi) = \frac{1}{2}\|w\|^2 + C\sum_i(\xi_i + \xi_i^*)$ (3.22)

Subject to:

$$w\varphi(x_i) + b - y_i \leq \varepsilon + \xi_i^*$$

$$y_i - (w\varphi(x_i) + b) \leq \varepsilon + \xi_i^*$$

$$\xi_i^*, \xi \geq 0 \quad (3.23)$$

Here, *C* is known as the penalizing factor or regularization parameter that controls the tradeoff between model complexity and training errors i.e. it determines the extent to which errors beyond ±ε are tolerated.

$\xi_i \; and \; \xi_i^*$ are positive slack variables defined by

$$|\xi| = 0 \; if \; |\xi| \leq \varepsilon$$

$$|\xi| = |\xi| - \varepsilon, \text{Otherwise} \quad (3.24)$$

The above primal optimization problem is more readily solved in its dual formulation by the introduction of Lagrange multipliers $\alpha_i \; and \; \alpha_i^*$. The values of the Lagrange multipliers are determined by solving the dual optimization problem using Quadratic Programming (QP). Once $\alpha_i \; and \; \alpha_i^*$ are determined, the optimal weights of the original function f(x) are computed as follows:

$$w = \sum(\alpha_i - \alpha_i^*)\varphi(x_i) \quad (3.25)$$

And the optimal bias is determined as $b = \frac{1}{l}\sum_{i=1}^{l}(y_i - \varphi(x_i)^T w)$ (3.26)

The solution of the dual problem is given by: $f(x) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)K(x_i, x)$ (3.27)

3.4.2.2 The Least Square SVR (LS-SVR)

The least squares SVR is a modification of the standard SVR that uses a quadratic (least square) loss function and equality constraints. The advantages of LS-SVR over standard SVR is, reduced complexity and computational requirement because instead of solving a quadratic optimization problem, the technique involves solution of a set of linear equations.

The optimization problem of LS-SVR is given below (Suykens and Vandewalle, 1999):

$$\text{Minimize } J(w,\xi) = \frac{1}{2}w^T w + C\frac{1}{2}\sum_{i=1}^{N}\xi_i^2 \tag{3.28}$$

$$\text{Subject to: } y_i = w^T\varphi(x_i) + b + \xi_i \tag{3.29}$$

The dual optimization problem for LS-SVR is given below:

$$\text{Minimize } L(w,b,\xi,\alpha) = \frac{1}{2}w^T w + C\frac{1}{2}\sum_{i=1}^{N}\xi_i^2 - \sum_{i=1}^{N}\alpha_i(w^T\varphi(x_i) + b + \xi_i - y_i) \tag{3.30}$$

The optimal solution to the above problem is obtained as below:

$$\left.\begin{aligned}
&\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n}\alpha_i\varphi(x_i)\\
&\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n}\alpha_i = 0\\
&\frac{\partial L}{\partial \xi} = 0 \Rightarrow \alpha_i = C\xi_i\\
&\frac{\partial L}{\partial \alpha} = 0 \Rightarrow w^T\varphi(x_i) + b + \xi_i - y_i = 0
\end{aligned}\right\} \tag{3.31}$$

The final LS-SVR model for function approximation is obtained as:

$$y = \sum_{i=1}^{N}\xi_i K(x,x_i) + b \tag{3.32}$$

3.4.2.3 Tuning of SVR Hyper-Parameters

A successful SVR model largely depends on proper selection of regularization parameter (C), kernel parameter (σ) and error bound ε for standard SVR and regularization parameter (C), kernel parameter for LS-SVR. A very high value of C results in only empirical risk minimization whereas very small C value will under fit the training set. As ε decreases, the number of support vectors increase which subsequently increases the model complexity. Similarly, very low value of σ results in better fitting of the training data and poor generalization whereas high values lead to poor approximation (Wang *et al*., 2003).

Proper selection of SVR hyper-parameters is an optimization problem for which no single method has been universally accepted. The various methods proposed in the literature include gradient descent algorithm for minimization of generalization error

(Chapelle *et al*., 2002), grid search combined with cross validation method (Hsu *et al*., 2003), analytical method (Cherkassky and Ma, 2004), genetic algorithm (Huang and Wang, 2006), particle swarm optimization (Yuan and Chu, 2007, Lin *et al*., 2008), simulated annealing (Lin *et al*., 2008) and ant colony optimization (Zhou *et al*., 2012).

Gradient descent method suffers from the drawback of getting trapped in local minimum (Li and Tan, 2010). A major drawback associated with the evolutionary computation approaches is the need of prior knowledge of some algorithm parameters in order to perform the optimization. Values of these parameters significantly affect the final result of the optimization algorithm. While one can always ensure optimum values in grid search method by increasing the search domain, significant computation is required with increase in search domain and the number of SVR hyper parameters. Grid search is difficult to perform, once the number of parameters exceeds two (Chapelle *et al*., 2002). This is a major advantage of LS-SVR over standard SVR enabling the designer to conduct grid search based optimization for LS-SVR hyper-parameters.

The proposed approaches for SVR hyper-parameter tuning are based either on cross validation technique (n-fold or leave one out) (high computational effort) or use of a separate validation set. The n-fold cross validation involves dividing the data to *n* number of subsets, perform training on *n-1* subsets and test with the left out subset and determine the error, repeat these steps with each of the n subsets and finally combine all the errors generated with the test subsets. The optimal set of hyper-parameters is the one that produces least of these errors. The method of leave one out (LOO) cross validation method is even more rigorous where *n* is equal to the number of training observations so that for each set of hyper-parameters, model development is done the number of times equal to the number of observations.

In this work, for design of standard SVR model, grid search with cross validation method was adopted. However, to reduce the computational burden, the number of parameters for grid search were limited to two (ε and σ) and the parameter C was decided in priori using the analytical expression proposed by Cherkassky and Ma (2004) which is proposed as mentioned below:

$$C = max\left(|\overline{y} + 3\sigma_y|, |\overline{y} - 3\sigma_y|\right) \tag{3.33}$$

The standard SVR model was developed using one of the widely used MATLAB SVM Tool box (Gunn, 1998).

For design of LS-SVR, grid search operation with criterion of error minimization on the validation set was adopted for the LS-SVR hyper parameter selection. The LS-SVR model was developed using the freely available MATLAB toolbox LS-SVMlab (Pelckmans *et al*., 2002).

3.5 Artificial Neural Network Modeling

A neural network model requires a complete input – output dataset. The theoretial details of artificial neural network can be found in standard text books such as Samarasinghe (2007).

3.5.1 Back Propagation Neural Network (BPNN)

The back propagation neural network (also known as multilayer perceptron) is a kind of feed-forward neural network consisting of an input layer, an output layer and one or more hidden layer neurons as shown in the Figure 3.2.

The back propagation algorithm has been the most popular neural network design algorithm till date. Neural network models using back propagation algorithm has been developed for estimation of: PET viscosity in polymerization process (Gonzaga *et al*., 2009), pollutant emission from cement kiln (Marengo *et al*., 2006), process output variables of water desalination plant (Al-Shayji, 2002), water content of natural gases

(Mohammadi and Richon, 2007), crude oil viscosity (Elsharkwy and Gharbi, 2000), river flow (Kisi and Cigizoglu, 2007), various polymer properties (Roy *et al*., 2006), enzyme activity and biomass concentration (Linko *et al*., 1997).

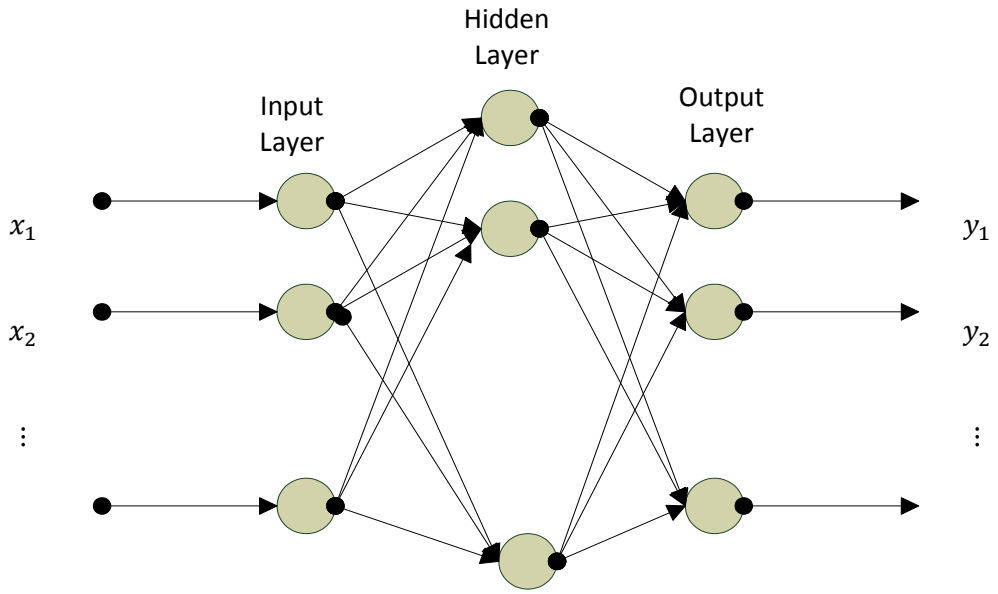


Figure 3.2: Feed forward neural network structure

In a BPNN model, the number of neurons in the input layer and output layer are same as number of input variables and output variables of the process respectively. Choosing the number of hidden layers and number of neurons in each hidden layer are the most critical decisions to successful design of BPNN. Unfortunately, there is no universal method to determine the optimum network topology and these are mostly decided based on a trial and error procedure so as to produce the least error (Mohammadi and Richon, 2007; Gonzaga *et al*., 2009). Usually a single hidden layer is used to solve functional approximation problems and if the performance goal is not attained in a single hidden layer gradually the number of hidden layers can be increased. The more the number of hidden layers the more is the complexity associated and large training time required. Few number of neurons in the hidden layer leads to poor model accuracy whereas many number of neurons result in model over fitting and poor generalization.

3.5.2 Radial Basis Function Neural network (RBFNN)

Multilayer feed forward networks with sigmoidal activation functions have been proven to be universal approximators which are mostly trained by back propagation method using gradient descent algorithm. However, the disadvantages of back propagation neural networks are: excessive computational or training time due to use of non-linear optimization techniques and possibility of getting trapped in local minima resulting in sub-optimal solution (Chen *et al*., 1991; Gurumoorthy and Kosanovich, 1998; Samanta, 2010).

Radial basis function networks are a class of feed-forward supervised networks. It is a three layer network consisting of an input layer, a hidden layer and an output layer with linear parameters. Non-linear basis functions are used at the hidden layer neurons. A centre is associated with every hidden layer node. Hidden layer nodes calculate the Euclidean distance between the centre and the input vector which is sent as input to the basis function. The different types of basis functions used are (Chen *et al*., 1991):

Thin plate spline function: $\varphi(x) = x^2 \log(x)$ (3.34)

Gaussian function: $\varphi(x) = \exp\left(-\frac{x^2}{\sigma^2}\right)$ (3.35)

Normalised Gaussian function $\varphi_i(x) = \frac{exp\left(\frac{-\|x-c_i\|^2}{\sigma_i^2}\right)}{\sum_{j=1}^{N} exp\left(\frac{-\|x-c_j\|^2}{\sigma_j^2}\right)}$ (3.36)

Multiquadratic function: $\varphi(x) = (x^2 + \sigma^2)^{1/2}$ (3.37)

Inverse multiquadratic function: $\varphi(x) = (x^2 + \sigma^2)^{-1/2}$ (3.38)

σ is the scaling parameter or width which controls the spread of the function around the centre.

Out of the above functions, the Gaussian type is mostly used as activation function for hidden layer nodes.

For an input vector x, the network output is given as:

$$\hat{y} = f(x) = \sum_{i=1}^{N} w_i \, \varphi(\|x - c_i\|) \tag{3.39}$$

w_i is the weight associated with i[th] RBF centre and $\|x - c_i\|$ is the Euclidean distance between centre c_i and the input vector x.

The linear parameters used in RBF networks result in faster training and less convergence problems in comparison to BP neural networks. Also RBF networks have better approximation ability with simpler network architecture as compared to Multi layer perceptrons (Sarimveis *et al*., 2002). Selection of an appropriate radial basis network requires careful selection of basis function and their associated parameters (centres and widths). The performance of an RBF network largely depends on the centres chosen. As a strict interpolator the network must have as many RBF centres as the training data. However, this results in a large structure when the data are plenty. Moreover, the large model structure results in over-fitting of the training data and poor generalisation capability of the network. On the other hand use of very less number of centres results in under fitting of the data (Ghodsi, 2003).

The centres and widths are obtained using *k*-means clustering algorithm or density estimation methods. This involves classifying the input data into *k* number of clusters. The cluster centres are determined by minimising the total squared error incurred in representing the dataset by *k* cluster centres. However, the drawbacks of this standard algorithm are that for determining the hidden nodes many passes of all training data are required resulting in large computational time for large dataset (Sarimveis *et al*., 2002). Moreover, though this method has faster training but results in local optimum yielding suboptimal models (Marinaro and Scarpetta, 2000; Li *et al*., 2004; Billings and Zheng, 1995). The second category makes use of algorithms to determine the network structure as well as the parameters. Some of the proposed algorithms are: orthogonal least squares

algorithm (Chen *et al*., 1991; Samanta, 2010), genetic algorithm (Billings and Zheng, 1995), individual training of each hidden unit based on functional analysis, fuzzy partition of input space followed by linear regression (Sarimveis *et al*., 2002).

In the present work, a two-layered feed-forward neural network was constructed. The first layer has radial basis neurons with Gaussian activation function as given in equation to perform the non linear transformation of the input signal. The second layer has linear neurons which produce linear outputs. Orthogonal least squares algorithm was used for designing of the RBF neural network model. The following iteration is performed until the network's mean squared error falls below goal or the maximum number of neurons are reached:

1) The network is simulated with no neurons in the first layer.

2) The input vector with the greatest error is determined.

3) A radial basis neuron is added with weights equal to that vector.

4) The output layer weights are redesigned to minimize error.

3.5.3 Generalized Regression Neural Network

Generalized regression neural network (GRNN) which was first proposed by Specht (1991) is a powerful tool for non linear function approximation. In the general regression algorithm, the form of input – output dependence is expressed as a probability density function determined from the observed data. The algorithm has the form (Specht, 1991):

$$\hat{y} = \frac{\sum_{i=1}^{N} y \times exp\left(\frac{-D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{I} exp\left(\frac{-D_i^2}{2\sigma^2}\right)} \tag{3.40}$$

D_i^2, the Euclidean distance between two input vectors is given as:

$$D_{ij}^2 = \left(x_i - x_j\right)^T \left(x_i - x_j\right) \tag{3.41}$$

The regression equations can be implemented in a neural network like structure which is then known as GRNN.

A typical GRNN has four layers: an input layer, a pattern layer, a summation layer and the output layer as shown in the figure below.

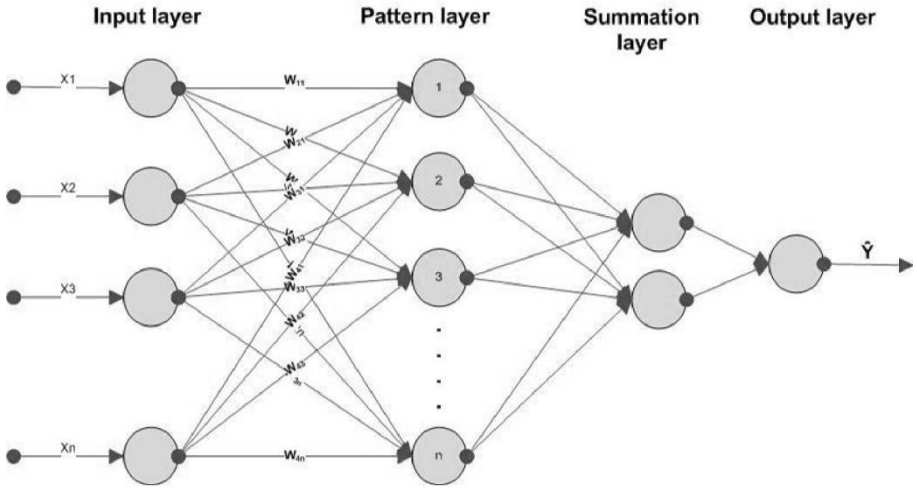


Figure 3.3: Generalized regression neural network structure

Input layer has the same number of neurons as the number of input variables and pattern layer has the same number of neurons as the number of training cases. Pattern neurons compute a distance which is the square of differences across all weights as described in Equation 3.41.

The activation function associated with the pattern neuron is exponential and can be written as $exp\left(\frac{-I_j^2}{2\sigma^2}\right)$.

Here $I_j = \sum_{i=1}^{N}\left(w_{ij} - x_j\right)^2$ (3.42)

The choice of smoothing function or spread parameter σ is critical to the successful design of a GRNN. A large value of σ results in more generalization and smoother fitting whereas a low value results in more accurate fitting and poor generalization. The method

suggested for optimum selection of σ is the hold out or leave one out method (Specht, 1991; Goh, 1999).

The summation neurons calculate the sum of weighted inputs from pattern layer. There have been some applications of GRNN modeling for estimation of crude oil viscosity (Elsharkwy and Gharbi, 2001), river flow (Kisi and Cigizoglu, 2007), polymer property (Roy *et al*., 2006), soil quality (Goh, 1999), river sediments (Cigizoglu and Alp, 2006), coal grindability (Peisheng, 2005), plasma process parameters (Kim *et al*., 2009), water quality (Palani *et al*., 2008), compressive strength and elasticity modulus (Dehghan *et al*., 2010), NOx emission (Zheng *et al*., 2008).

3.6 Fuzzy Inference System (FIS)

The basic structure of a general fuzzy inference model is shown in Figure below:

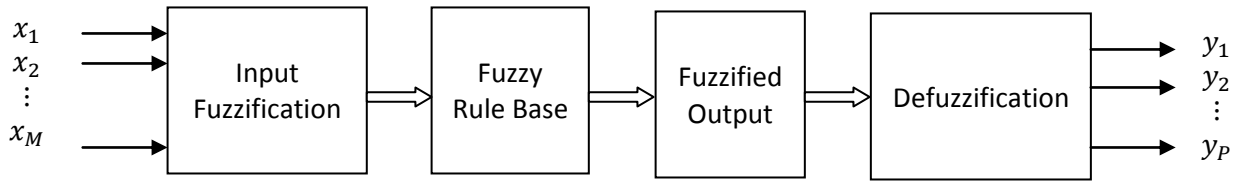


Figure 3.4: Fuzzy inference system

The important steps in any fuzzy system design are: fuzzification of the actual process variable data, construction of the rule base and finally defuzzification of the fuzzified output to obtain the actual output value. Fuzzification involves transformation of the real process values to a particular fuzzy linguistic variable with appropriate membership value in the range 0 to 1. The most common membership functions in use are triangular, trapezoidal or gaussian. The fuzzified output is determined by composition of fuzzified inputs and the appropriate rules from the rule base. The commonly used composition methods are min-max or max product composition. Finally the fuzzy output value produced for a particular observation has to be defuzzified in order to get the actual process output value. Commonly used defuzzification methods are (Jang *et al*., 1997):

centroid of area, bisector of area, mean of maximum, weighted average, smallest of maximum and largest of maximum.

In the present work, both Mamdani and Sugeno type fuzzy inference models were developed for the two industrial processes. Five linguistic variables were used for each process variable (input or output for Mamdani model and inputs for Sugeno model). The linguistic variables are listed below along with their ranges:

VS: very small [0 - 0.2]

S: small [0.1 - 0.45]

M: medium [0.3 - 0.65]

L: large [0.5 - 0.85]

VL: very large [0.7 - 1]

A combination of triangular and trapezoidal membership functions (MF) are used. Triangular function is applied to the variables of 'very small' and 'very large' category and trapezoidal function for the other types of linguistic variables as described above. The mathematical expressions for triangular and trapezoidal membership function are presented below in Equations 3.43 and 3.44 respectively (Jang and Sun, 1997).

$$\mu(x,a,b,c) = max\left(min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right) \quad \text{(for triangular MF)} \tag{3.43}$$

$$\mu(x,a,b,c,d) = max\left(min\left(\frac{x-a}{b-a},1,\frac{d-x}{d-c}\right),0\right) \quad \text{(for trapezoidal MF)} \tag{3.44}$$

μ is the fuzzy membership value for the inputs. The parameters a, b and c (a<b<c) represent the three corners of the triangular membership function and a,b,c,d (a<b ≤c≤d) represent the four corners of the trapezoidal membership function.

For each data sample, the antecedent of the corresponding fuzzy rule has inputs equal to the number of process inputs variables. Logical *AND* operator is used to obtain the result of the rule as mentioned below:

$$\mu(x) = \min\{\mu(x_1), \mu(x_2),..., \mu(x_M)\} \text{ (for Mamdani FIS)} \quad (3.45)$$

$$\mu(x) = product\{\mu(x_1), \mu(x_2),..., \mu(x_M)\} \text{ (for Sugeno FIS)} \quad (3.46)$$

The fuzzified output is obtained by the min-max composition as shown below:

$$y = x \text{ o } R \quad (3.47)$$

here, y represents the fuzzified value of the output, x is the vector of fuzzified values of the inputs, R is the set of rules. The symbol 'o' represents composition.

For a particular set of inputs there may be a possibility of application of multiple rules producing multiple number of fuzzy outputs. Aggregation refers to combination of different fuzzy output membership values to produce a single fuzzy output. This is a maximum method which involves fuzzy union of all possible fuzzy outputs. Centroid and weighted average method mentioned below were used as the method of defuzzification for Mamdani and Sugeno type FIS respectively.

Centroid method: $$y = \frac{\int y\mu_{out}(y)dy}{\int \mu_{out}(y)dy} \quad (3.48)$$

Weighted average method: $$y = \frac{\sum_{i=1}^{N} \mu_{out}(y)^i w^i}{\sum_{i=1}^{N} \mu_{out}(y)^i} \quad (3.49)$$

μ_{out} is the fuzzy membership value for the output

Unlike Mamdani method, the output in a Sugeno type model is either linear function of the inputs or constant following which the model is known as either a first order Sugeno fuzzy model or zero order Sugeno fuzzy model.

3.7 Hybrid Modeling

3.7.1 PCA-BPNN

In a pure BPNN modeling the inputs to the neural network modeling are the actual process variables. In the combined approach of PCA-BPNN modeling, the inputs to the neural network model are the principal components or latent variables which are the linear combinations of actual variables obtained by conducting PCA on the input data matrix. The number of modified inputs (principal components) are decided using the method described in Section 3.2.

3.7.2 Adaptive Neuro-Fuzzy Inference System (ANFIS)

In the basic fuzzy inference system, the membership functions are chosen arbitrarily. However, the shape of the membership functions depends on parameters associated with it, and changing these parameters change the shape of the membership function.

Neuro fuzzy inference systems are basically kinds of adaptive network models which are functionally equivalent to fuzzy inference models. The network model consists of 5 layers. Based on the input value appropriate membership value is generated from the first layer. The output from a node in layer 2 is the product of all membership values reaching that node. In other terms the output from a particular node in second layer is the firing strength from that node. A particular node in the 3^{rd} layer calculates the ratio of the individual firing strength to the sum of all rules firing strength. Outputs of this layer are called normalized firing strengths. Every node in the 4^{th} layer is an adaptive node. The outputs produced from the nodes of 4^{th} layer are equal to the product of normalized firing strength received and the function relating the output and the inputs. The parameters associated with the functional relationships are called as consequent parameters. Nodes in the final layer calculate the overall outputs as the summation of all input signals.

Number of nodes in the first layer will be equal to the number of fuzzy linguistic variables created for all input variables. Similarly number of nodes in layers 2, 3 and 4 are equal to the number of fuzzy *if-then* rules. The membership function parameters are optimized using either a back propagation algorithm alone or a hybrid algorithm (a combination of least-squares and back propagation gradient descent method). The algorithm details can be found in Jang (1993).

In process industries, ANFIS modeling approach has been investigated for estimation of wastewater effluent quality (Perendeci *et al*., 2008; Pai *et al*., 2009). In particulate studies ANFIS has been applied for prediction of weight of the residual granule (WRG) values in drying process (Azadeh et. Al., 2012). This WRG is an indirect indication of the particle size. However the potential of ANFIS technique for direct estimation of particle size has rarely been explored.

Chapter - 4

Design of Soft Sensors for Quality Monitoring in Cement Manufacturing Processes

The various techniques of soft sensor development described in Chapter 3 were applied for product quality monitoring in cement plant. Two processes were considered in the cement plant, a chemical process (pyroprocess) where the raw materials are converted to clinker and a physical process (cement grinding) where the clinker is converted to finished product i.e. cement. Description of these two processes along with the soft sensor development procedure is described in the following sections.

4.1 Soft Sensor Development for Clinker Grinding Process

4.1.1 Process Description

In the clinker grinding process, clinker coming from rotary cement kiln is ground in the cement mill along with a small amount of gypsum and/or fly ash. A schematic diagram of the vertical roller mill used for clinker grinding is shown in Figure 4.1.

Vertical roller mill (VRM) is the most important constituent of the grinding process which consists of a grinding table with grinding rollers fitted on the table periphery. A minimum clearance is maintained between the table top surface and the grinding roller surfaces. The table rotates with certain fixed rpm about the axis passing through the centre and perpendicular to it. The material (clinker, gypsum and/or fly ash) is discharged from the top directly onto the centre of the grinding table. Due to centrifugal action, the material is spread outward and gets crushed by coming in between the table top surface and roller surface. The grinding mechanism in the VRM is shown in Figure 4.2.

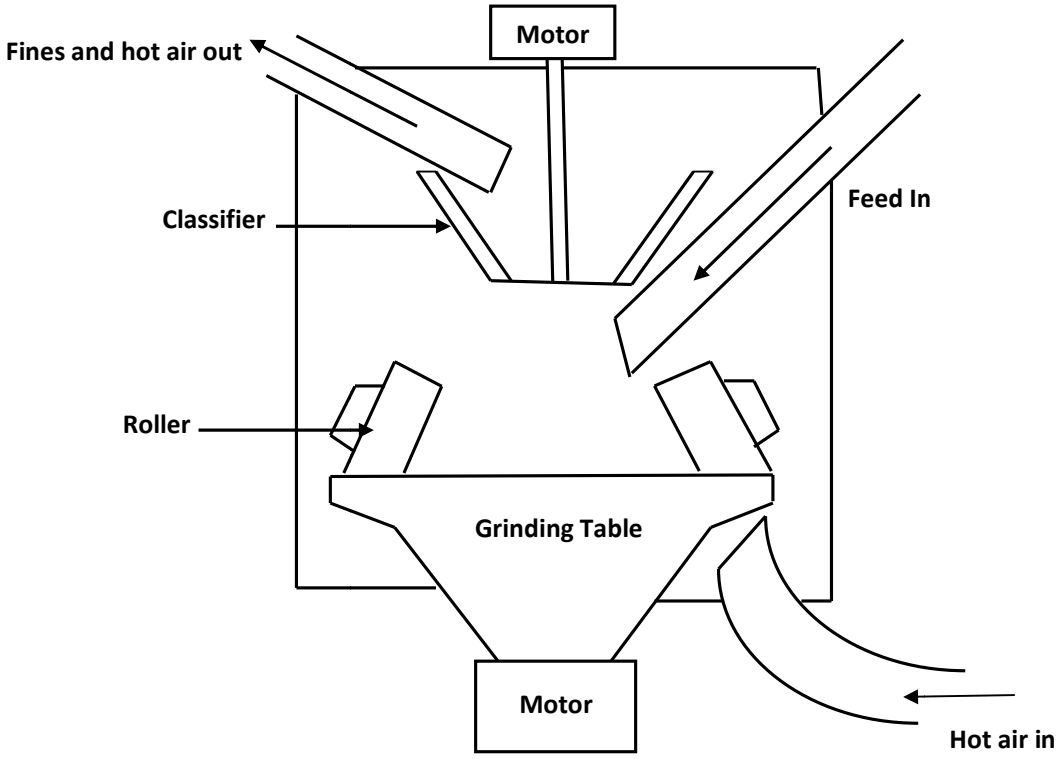


Figure 4.1: Schematic diagram of vertical roller mill

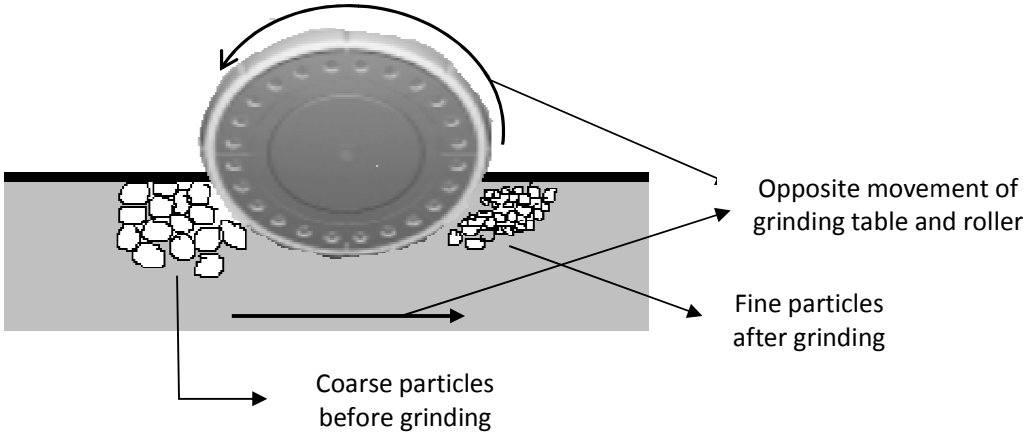


Figure 4.2 Grinding process in the mill

The fine size particles remain on the top surface and some materials also escape out of the table without grinding, because of the existing gap between the rollers. An induced draft fan operates at the process outlet to carry the ground material by air to the subsequent sections. A classifier is arranged at the top of the mill to separate the fines from the coarse material. The rejected material is collected at the bottom and circulated back to the mill by

the use of bucket elevators. The cement is separated from the carrier gas in the bag house and sent to the cement silo for subsequent packaging and distribution. The entire grinding process flow diagram is presented in Figure 4.3.

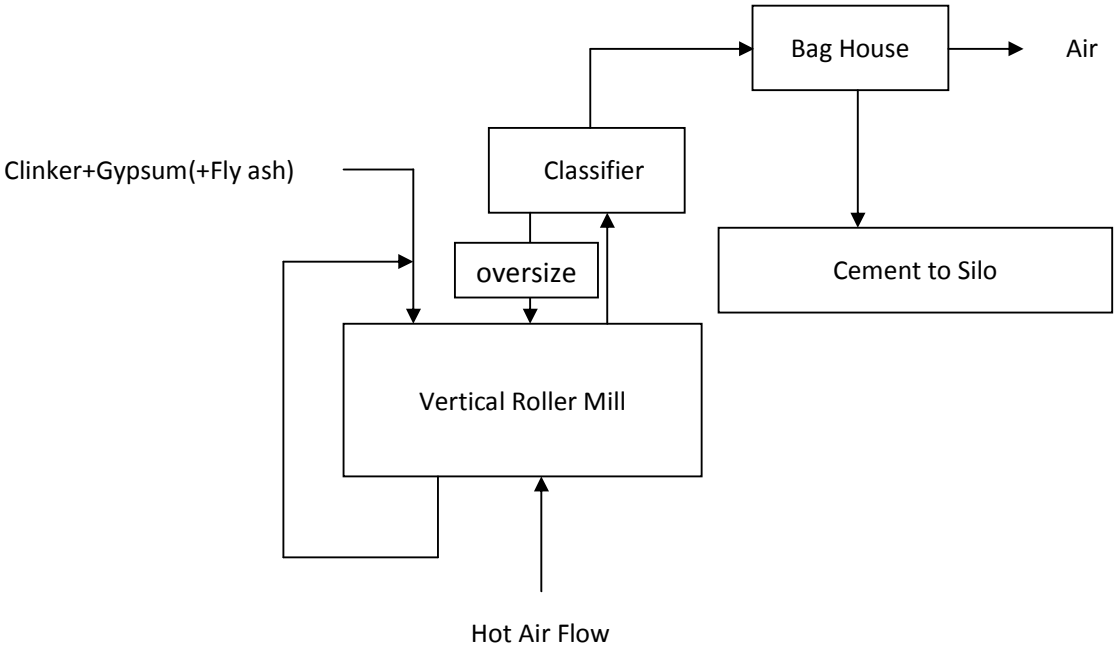


Figure 4.3: Clinker grinding process in cement plant

4.1.2 Data Collection and Treatment

The quality parameter i.e. cement fineness is determined by offline laboratory analysis in terms of Blaine number which is defined as the surface area per unit mass (expressed in the units of m^2/kg). High Blaine number indicates more fineness of the cement. The key factors affecting the output quality from the mill are: the incoming flow rate of the material i.e. clinker, gypsum and/or fly ash, air flow rate through the mill and the rpm maintained in the classifier. Higher inflow rate of clinker results in less efficient grinding and therefore less cement fineness. Similarly higher air flow rate results in more amount of ground material forced through the classifier thereby decreasing the cement fineness. A higher classifier rpm does not allow coarse particles to pass through it and results in increase of cement fineness.

It can be noted that the rotational speed of the grinding table and the hardness of the inlet clinker are also potential input variables. However, the process study and the consultation with plant operators revealed that these two parameters are maintained constant during the grinding operation. Some may argue that if the model is built assuming constant properties (hardness) of the inlet material, the model performance will be unreliable when there is a change in this property. While this may be true, in case of cement grinding process, the inlet material cement clinker is received from the kiln after proper quality testing. Therefore, any deviation of clinker from the prescribed quality at the kiln output will lead to rejection of the clinker instead of being sent to the grinding mill. Therefore, based on these reasoning, online data pertaining to the aforementioned three input process variables were collected from plant database history which significantly affect the mill output i.e. cement fineness. Data for cement fineness were collected from the plant laboratory where the fineness is measured every one hour by taking ground cement sample from the mill. Figure 4.4 shows the raw input data as collected from the plant database and Table 4.1 shows the characteristics of the raw data as received from the plant.

Outliers are more likely to occur in the dataset in which values are produced by installed online hardware sensors than in the laboratory data because of the possible scenarios of sensor malfunctioning or plant shut down. The abnormally high values and standard deviation of the inputs variables (Figure 4.4 and Table 4.1) indicate the presence of outliers in the raw data received from the plant. The three popular univariate outlier detection techniques described in Section 3.1.2.1 were used for data preprocessing.

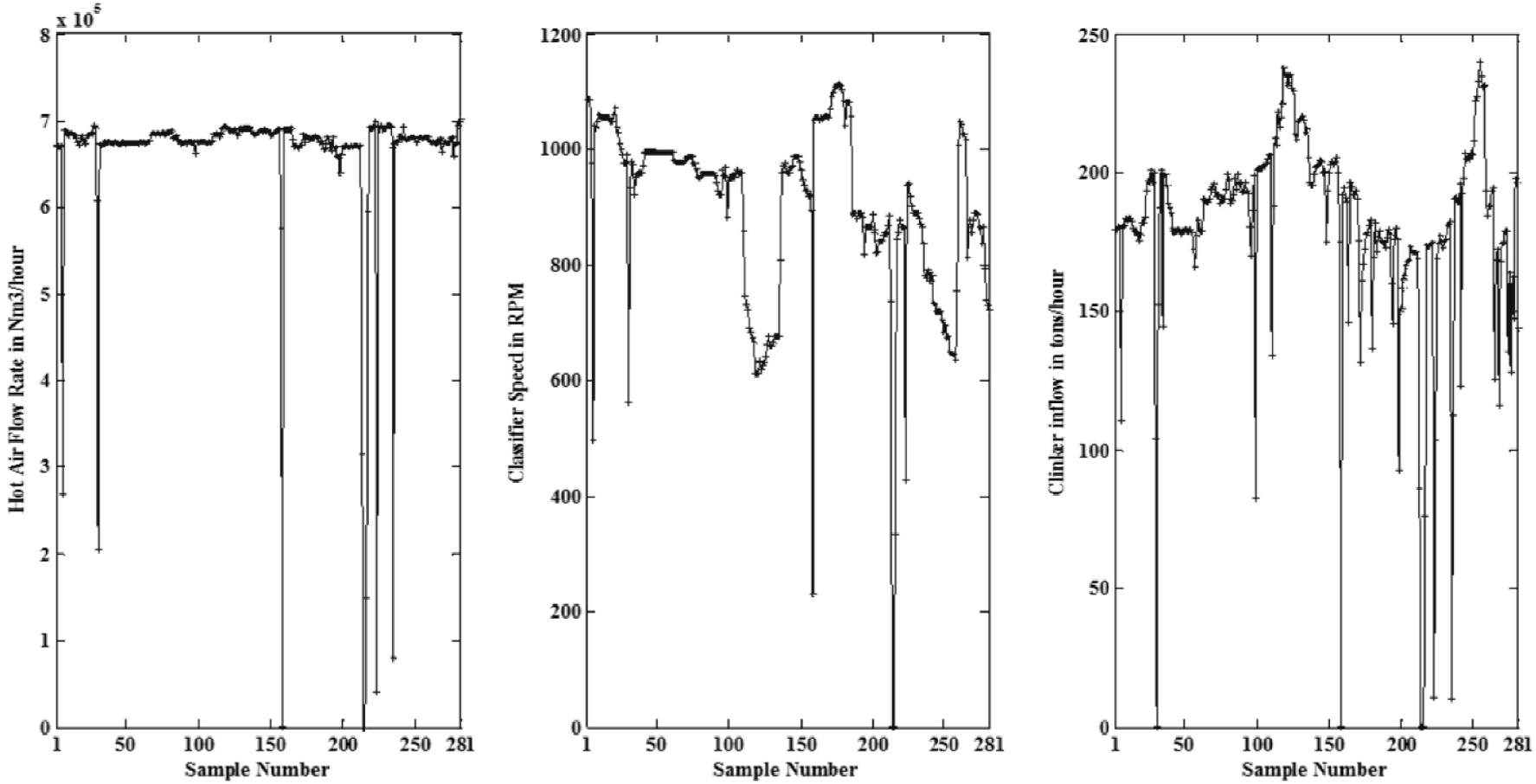


Figure 4.4: Raw input data for cement mill

Table 4.1: Characteristics of raw data as received from the cement plant

VARIABLES		MINIMUM	MAXIMUM	AVERAGE	STANDARD DEVIATION	SKEWNESS	KURTOSIS
Inputs	Hot air flow	0	703517	660915.99	101843.158	-5.516	32.78
	Classifier RPM	1.375	1114.133	896.86	162.834	-1.929	9.405
	Clinker inflow	0	239.828	180.723	38.344	-2.693	12.794
Output	Cement Blaine	266	385	338.79	34.268	-0.624	1.722

Under each technique, the detected outliers were removed and the resulting missing values were imputed by using linear interpolation or extrapolation with the neighborhood values. The performance of the three techniques were assessed by analyzing the descriptive statistical results of the resulting datasets. The clean data resulted from application of Hampel's method were subsequently used for modeling. The reason for choosing the Hampel's method are presented in Chapter 5.

For the purpose of data-driven soft sensor design, the input and output datasets should be of the same length. While a total of 281 values were obtained for the input variables, 158 values for cement fineness were obtained from the laboratory for the same period. Therefore, to maintain the dimensional uniformity between the input and output, 158 sets of input variables were retained or calculated by linear interpolation corresponding to the time instants when the laboratory information was available. The resulting sets of modified data are shown in Figure 4.5. After outlier detection, removal and missing value imputation, the whole dataset was normalized in the range 0 to 1 using Equation 3.9.

Figure 4.5 represents the total number of available datasets for the three input and one output variables. This total dataset was split equally into a training subset and a validation subset each having 79 number of observations using three different techniques of random selection, Kennard-Stone (K-S method) maximal intra distance criterion and the duplex method. Out of the three different pairs of subsets, the pair obtained using the K-S method

was subsequently used for the development of different empirical and hybrid models. The reasons for using K-S method of selection over the other two methods are provided in Section 5.2.

For developing principal component regression and PCA-BPNN models, the three input variables were transformed to three principal components (latent variables) by conducting PCA on the input data matrix (The procedure is described in Section 3.2).

4.1.3 Soft Sensor Model Development

Simple methods of linear regression, quadratic regression and more recently proposed technique of support vector regression techniques were applied for statistical model development. While in linear model the output is expressed as a linear combination of the inputs, in quadratic model the output is expressed as a combination of the inputs, squared inputs and the interaction terms. The parameters of the regression models were determined using least square error optimization method (Equation 3.19). The two statistical models obtained are produced below:

Linear regression model: $y = -0.13563 - 0.10143x_1 + 1.054x_2 + 0.3195x_3$ (4.1)

Response surface model:

$$y = -0.3933 + 1.1507x_1 + 1.4708x_2 - 0.215x_3 - 0.75615{x_1}^2 - 0.144{x_2}^2 + 0.256{x_3}^2 - 0.92911x_1x_2 + 0.08x_1x_3 + 0.3748x_2x_3 \quad (4.2)$$

In the above equations, x_1, x_2 and x_3 represent hot air flow rate through the mill (Nm^3/h), RPM maintained in the classifier and the clinker inflow rate (tons/h) to the mill, respectively. Here, y represents the grinding process output i.e cement fineness (m^2/kg).

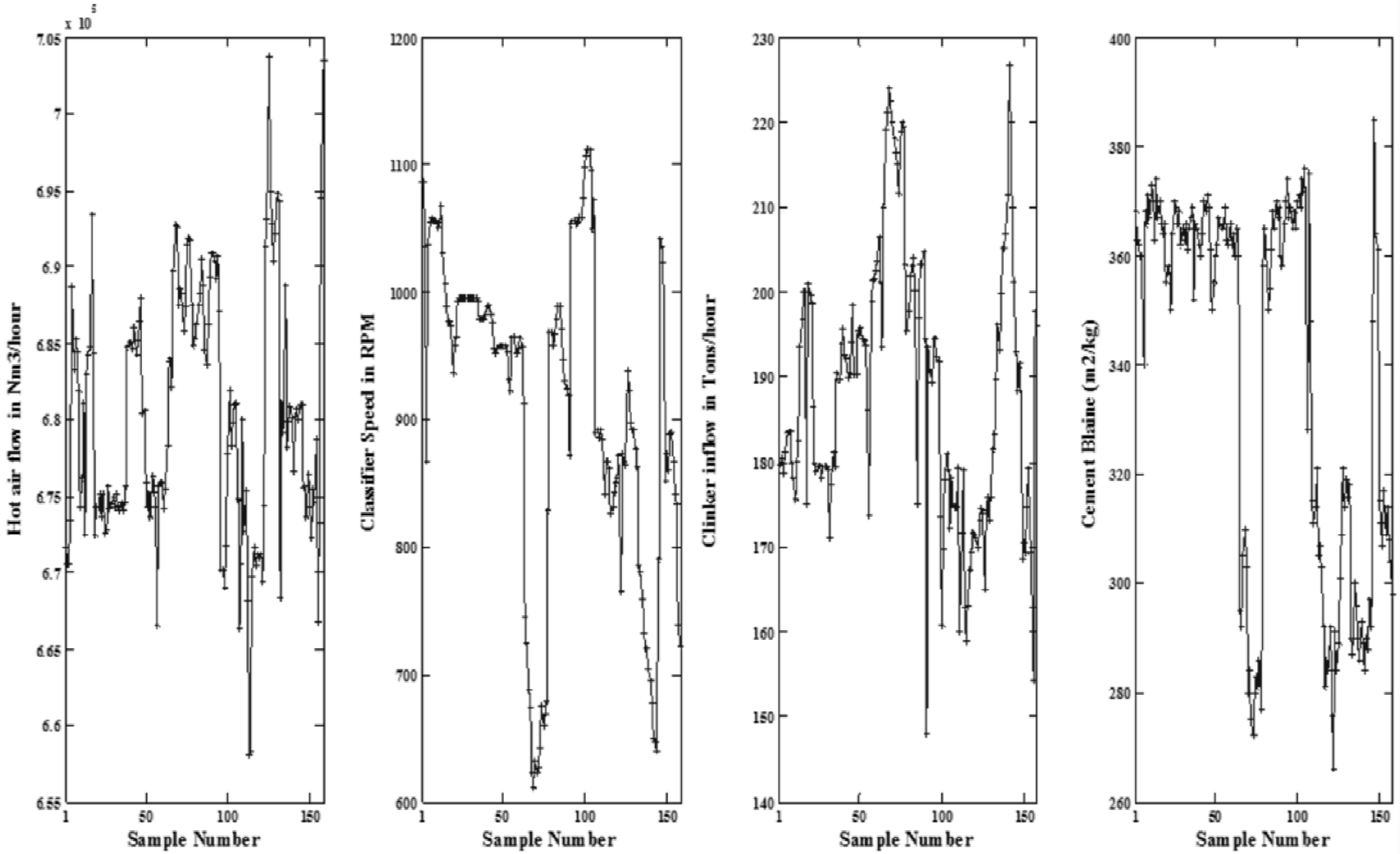


Figure 4.5: Final input-output data for modeling of cement mill

For development of principal component regression (PCR) model, three principal components were obtained for the three actual process input variable out of which two principal components account for more than 80% of the total variance of the input dataset (Details are presented in Table 5.6, Chapter 5). Therefore, the input dimension was reduced from three actual input variables ($x_1,\ x_2, x_3$) to two principal components or latent variables (p_1 and p_2). The least square regression model equation with the two latent variables is given below:

Principal component regression model: $y = -0.0966 + 0.758p_1 - 0.629p_2$ (4.3)

The key issue in support vector modeling is the optimal selection of the parameters *C*, ε and σ for standard SVR and optimal selection of *C* and σ for LS-SVR. In this work, to ensure best values of the parameters, grid search with cross validation method was adopted. However, to reduce the computational burden, the number of parameters for grid search were limited to two (ε and σ) and the parameter *C* was decided a priori using the analytical expression proposed in the literature (Equation 3.33).

The error bound ε and the RBF kernel width were determined by performing grid search followed by cross validation on the validation set so as to produce the least error on both the training and validation set. The search was done first by performing a coarse grid search followed by a fine grid search as suggested in the literature. The ranges of values used for the search operation are mentioned below:

Coarse grid search $\sigma = [2^{-4}, 2^{-2}, 2^{0}, 2^{2}]$

$\varepsilon = [2^{-8}, 2^{-6}, 2^{-4}, 2^{-2}]$ (4.4)

Fine grid search $\sigma = [2^{-2}, 2^{-1.875}, 2^{-1.75}, 2^{-1.5}, 2^{-1}]$

$\varepsilon = [2^{-4}, 2^{-3.875}, 2^{-3.75}, 2^{-3.5}, 2^{-3}]$ (4.5)

From grid search, two sets of optimum SVR parameters were obtained with different number of support vectors as given below:

SVR_1: $[C, \varepsilon, \sigma] = [1.4608, 0.068157\ (2^{-3.875}), 0.2726\ (2^{-1.875})]$

Number of support vectors (NSV) = 45 (56.9%)

SVR_2: $[C, \varepsilon, \sigma] = [1.4608, 0.003906\ (2^{-8}), 0.2726\ (2^{-1.875})]$

Number of support vectors (NSV) = 77 (97.4%).

The purpose of choosing two SVR models are explained in Section 5.3.1 (Chapter 5).

The parameters to be chosen for LS-SVR model are, the regularization parameter (C) and the kernel parameter(σ). Same grid search was performed for optimal LS-SVR hyper parameter selection. The domain of grid search is mentioned below:

Coarse grid search: $C = [2^{-5},2^{-3},2^{-1},2^{1},2^{3},2^{5},2^{7}]$

$$\sigma = [2^{-5},2^{-3},2^{-1},2^{1},2^{3},2^{5},2^{7}] \quad (4.6)$$

Fine grid search: $\sigma = [2^{-3} - 2^{-1}]$ and $C = [2^{1} - 2^{3}]$ (4.7)

The optimal values obtained are $C = 2^{2.75} = 6.72$ and $\sigma = 0.2332$.

In a feed forward back propagation neural network model, the number of nodes in the input and output layer is decided by process conditions. In this case there are three input variables of gas flow rate through the mill, solid flow rate to the mill and the classifier rpm. The output variable is the cement quality variable i.e. fineness values obtained from the laboratory. So, the important design decisions were to determine the number of hidden layers in the network, number of neurons to be present in each hidden layer and the activation function of the hidden layer neuron. Usually one or two hidden layers are used since increase in number of hidden layers results in increase in training time of the network. Therefore, one hidden layer was used. Since the entire dataset was normalized in the range 0 to 1, logarithmic sigmoidal activation function was used for the hidden layer neurons. This activation function has the form:

$$Output = \frac{1}{(1+e^{-net\ input})} \quad (4.8)$$

Linear activation function was used for output layer neurons. To decide the number of neurons to be used in the hidden layer, networks with different number of neurons were created and these were trained with different back propagation algorithms. The algorithms that were used are: gradient descent method with momentum, resilient back propagation algorithm and Levenberg-Marquardt algorithm.

For designing RBFNN, any of the basis functions given in Equations 3.34-3.38 can be used for the RBF neurons. The choice of the function does not play a crucial role in the performance of the model (Chen *et al*., 1991). Therefore, the simple and commonly used basis function, the Gaussian activation function presented in Equation 3.35 was used for the RBF neurons to perform the non linear transformation of the input signal. The second layer has linear neurons which produce linear outputs. Orthogonal least squares algorithm (Chen *et al*., 1991) which is available in MATLAB neural network toolbox was followed to determine the network parameters.

The optimum scaling parameter for regression neural network was determined by constructing networks with different spread values and assessing their performance with the validation dataset (Details are presented in Figure 5.8, Chapter 5).

The details of the three optimum neural network structures are presented in Table 4.2

Table 4.2: Neural network model details for the clinker grinding process

TYPE OF FEED FORWARD NEURAL NETWORK MODEL	NUMBER OF NEURONS	ACTIVATION FUNCTIONS USED
BPNN (Trained by resilient back propagation algorithm)	3 input layer neurons, 20 hidden layer neurons, 1 output layer neuron	Sigmoidal in hidden layer and linear in output layer
RBFNN (Orthogonal least square algorithm)	3 input layer neurons, 25 RBF layer neurons, 1 output layer neuron	Gaussian basis function in RBF layer and linear in output layer
GRNN	3 input layer neurons, 79 pattern layer neurons, 1 output layer	Gaussian function in pattern layer, linear in output layer

The three neural network models developed are produced in Figure 4.6

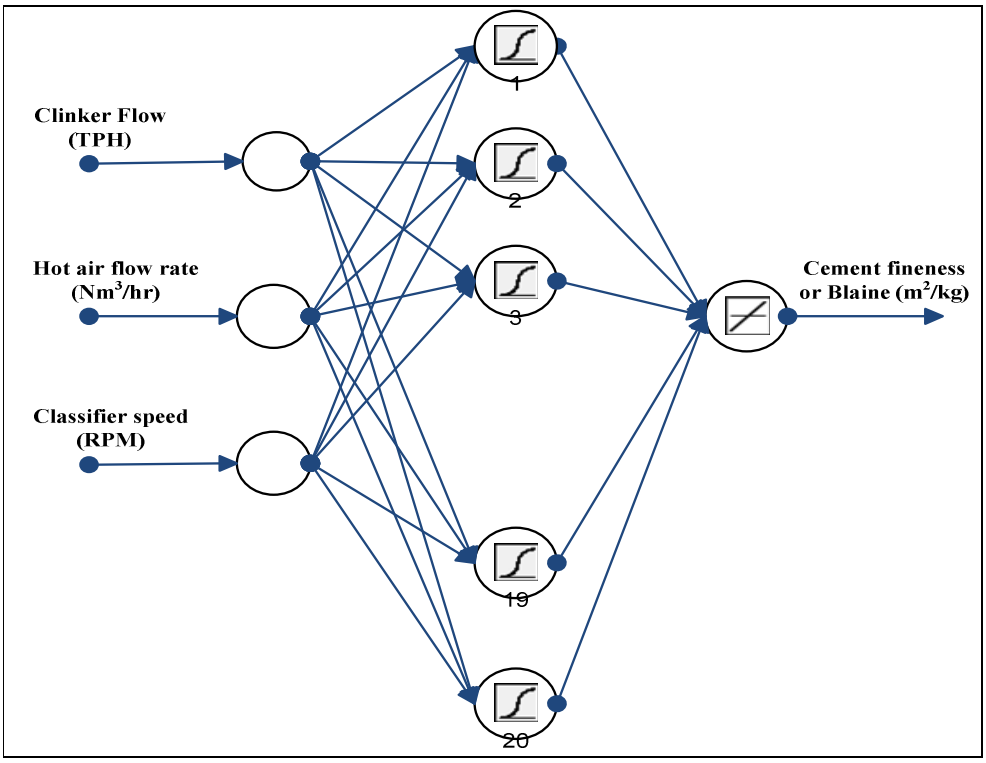


Figure 4.6(a): BPNN Model for the grinding process

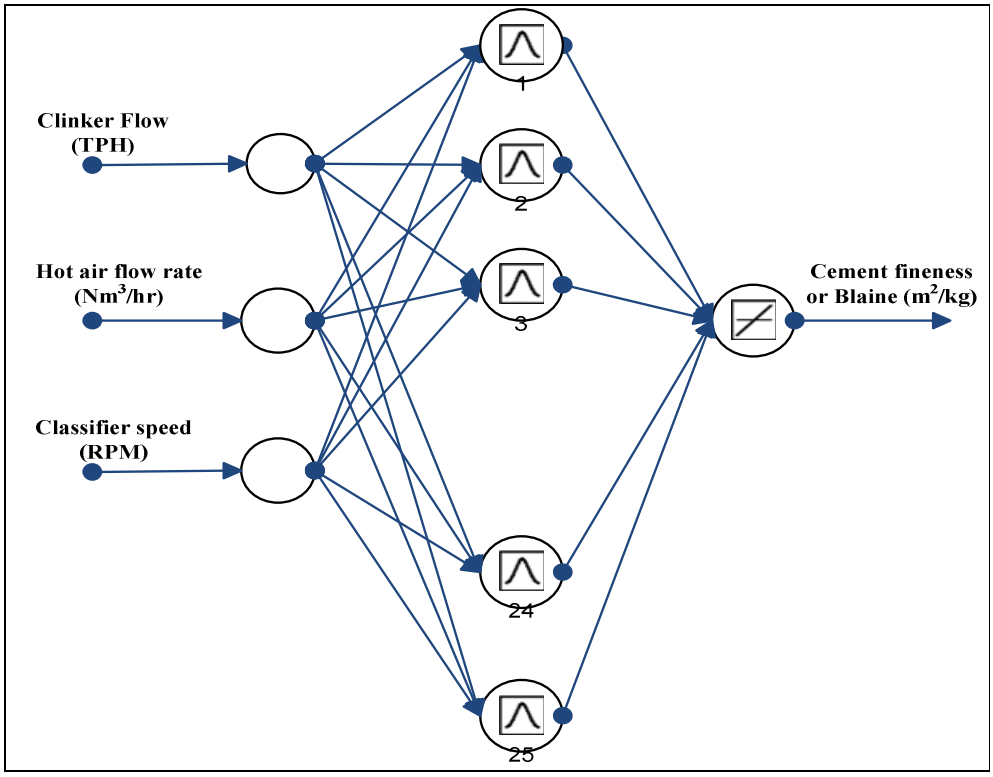


Figure 4.6(b): RBFNN Model for the grinding process

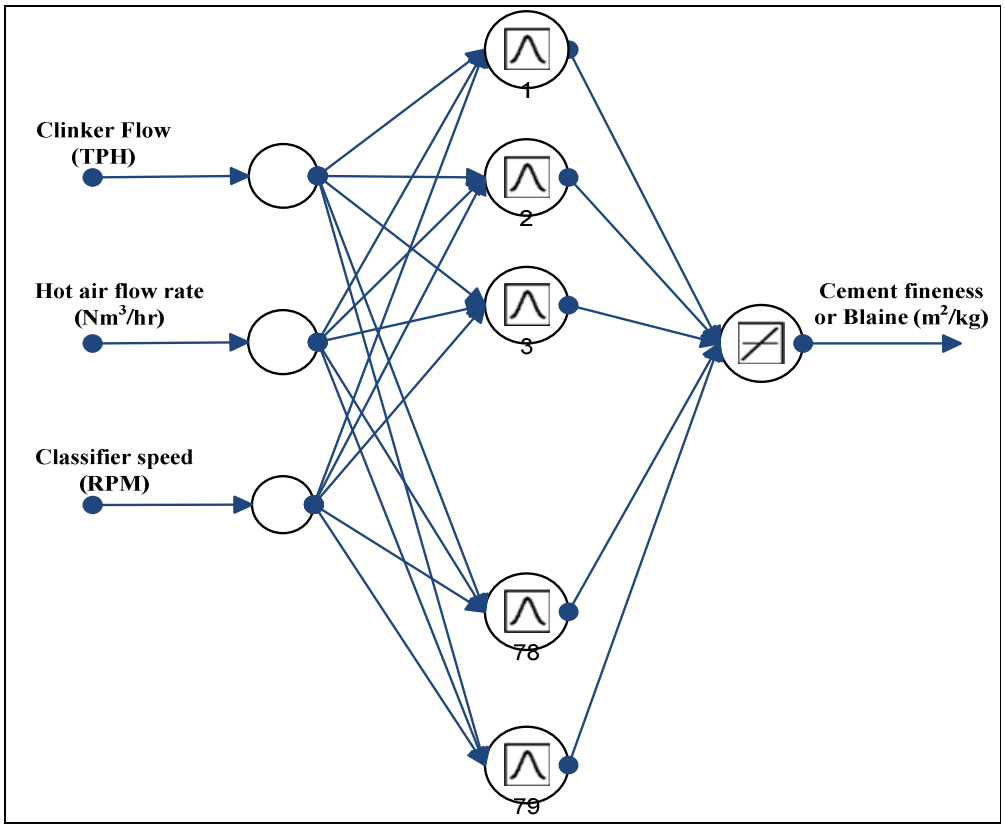


Figure 4.6(c): GRNN Model for the grinding process

The general structure of a fuzzy inference system (FIS) has been shown in Section 3.6. The specific model structure for the clinker grinding process is shown below in Figure 4.7.

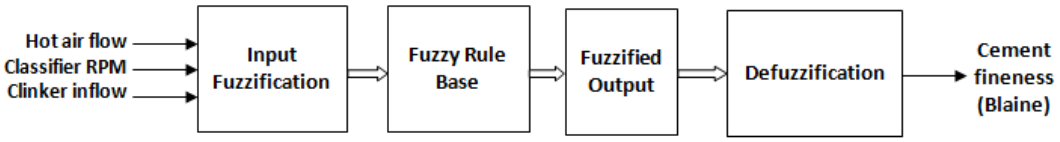


Figure 4.7: Fuzzy inference system for the clinker grinding process

It has been mentioned in Section 3.6 that designing of a fuzzy inference model requires fuzzification of the actual process variable data, construction of the rule base and finally defuzzification of the fuzzified output to obtain the actual output value. The normalized values of the three inputs and the output variable were used for fuzzification. Both Mamdani and Sugeno type fuzzy inference models were developed for the cement

grinding process. Unlike Mamdani method, the output in a Sugeno type model is either linear function of the inputs or constant following which the model is known as either a first order Sugeno fuzzy model or zero order Sugeno fuzzy model.

Five linguistic variables were used for each process variable (input or output for Mamdani model and inputs for Sugeno model). The linguistic variables are listed below along with their ranges:

VS (very small) :[0 - 0.2]

S (small) :[0.1 - 0.45]

M (medium) :[0.3 - 0.65]

L (large) :[0.5 - 0.85]

VL (very large) :[0.7 - 1]

For Sugeno type model, 8 constant output values in the range 0 to 1 were taken as mentioned below:

ES (extremely small) : 0.05

VS (very small) : 0.15

S (small) : 0.25

LM (low medium) : 0.35

M (medium) : 0.45

L (large) : 0.65

VL (very large) : 0.75

EL (extremely large) : 0.90

Figure 4.8 explains the fuzzification process for one input and the output (for Mamdani type).

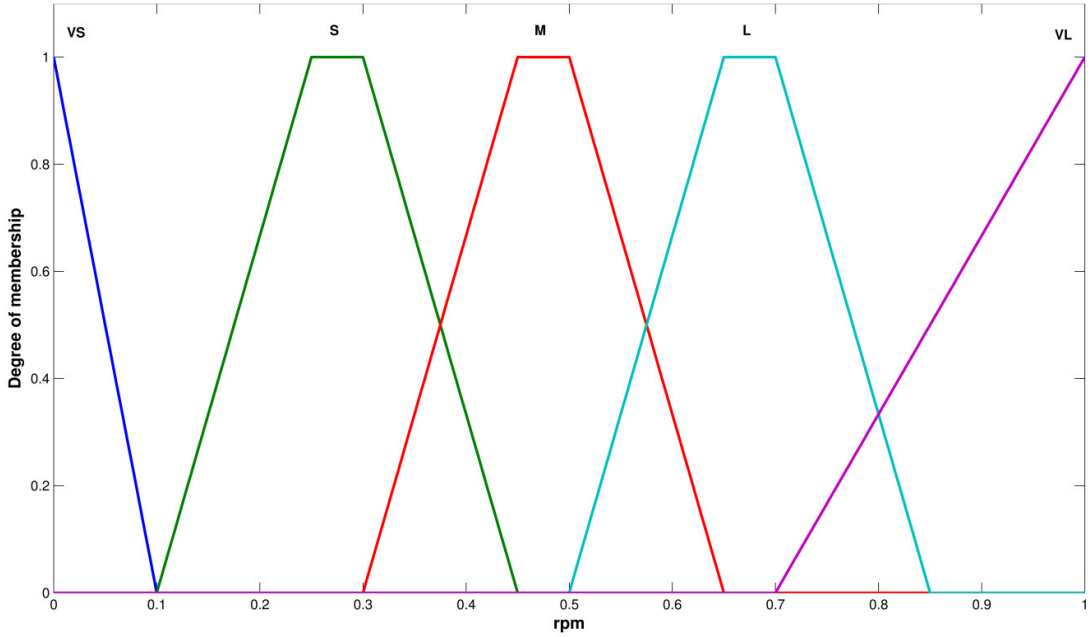


Figure 4.8 (a): Fuzzification of the input variable: classifier RPM

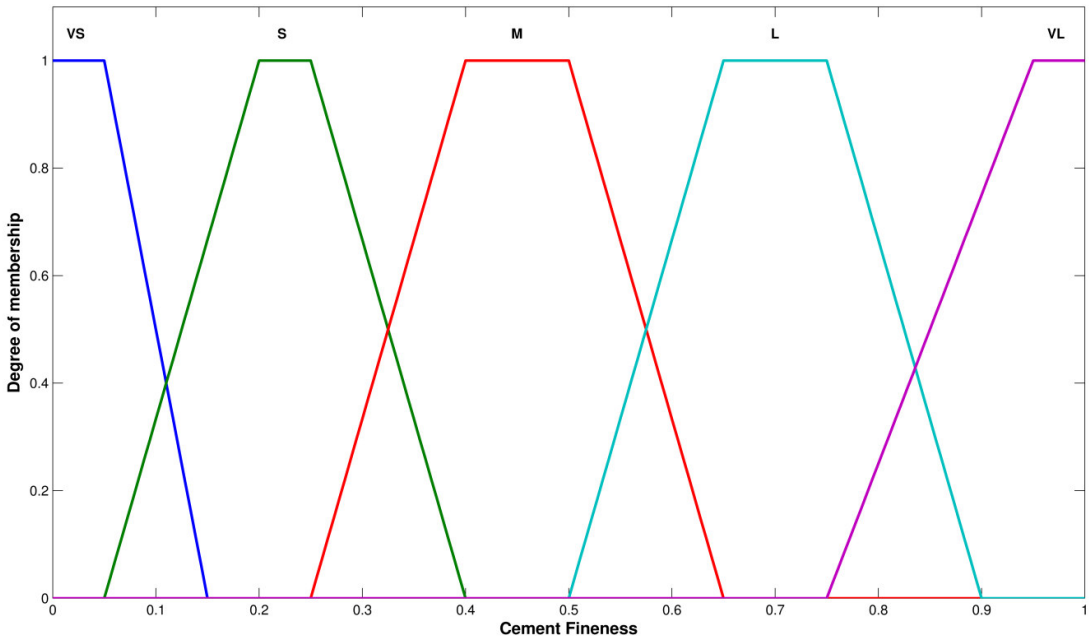


Figure 4.8 (b): Fuzzification of the output variable: cement fineness

Fuzzification of the other two inputs is same as the one shown above in Figure 4.8(a). The rule base was framed based on the physical understanding of the process and the available input - output dataset for model development. A total of 45 rules were framed for the Sugeno model and 63 fuzzy *if-then* rules were framed for the Mamdani type model.

For each data sample the antecedent of the corresponding fuzzy rule has inputs equal to the number of process inputs variables. AND operator is used to obtain the result of the rule as mentioned below:

$\mu(x) = \min\{\mu(x_1), \mu(x_2), \mu(x_3)\}$ (for Mamdani FIS)

$\mu(x) = product\{\mu(x_1), \mu(x_2), \mu(x_3)\}$ (for Sugeno FIS) (4.9)

The structure and working of the ANFIS model has been explained in Section 3.7.2. The developed ANFIS for the clinker grinding process is shown in Figure 4.9.

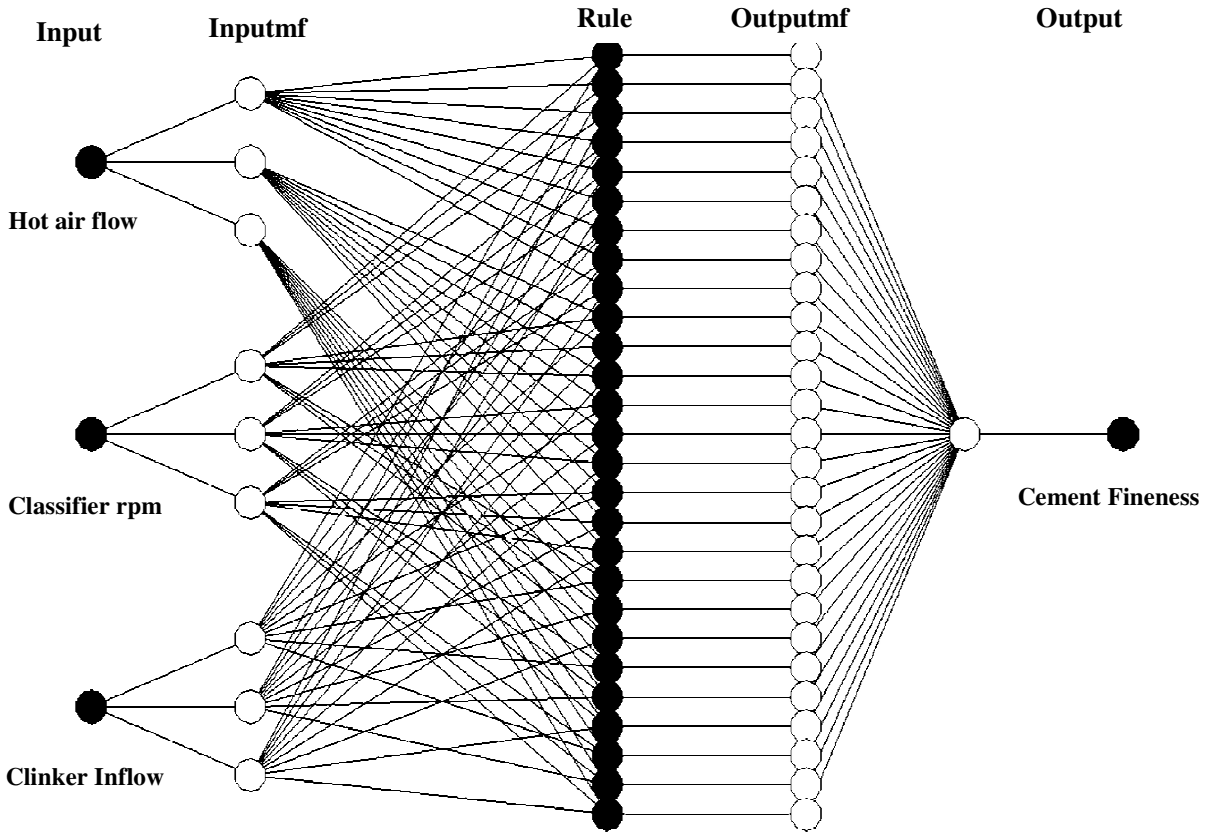


Figure 4.9: ANFIS model of the grinding process

The model structure consists of 5 layers named - *input*, *inputmf*, *rule*, *outputmf* and finally *output*.

Input layer has the same number of nodes as the number of process inputs which receive the three input values. In the subsequent *inputmf* layer a particular input value is fuzzified by being converted to a particular membership value. Every node in the *rule*

layer receives exactly 3 membership values from the three different process inputs. The AND operator produces a single membership value from the 3 membership values. The *outputmf* layer calculates the output membership value by multiplying the input with the linear relationship between the inputs and the output. If multiple rules exist for a single set of inputs, then multiple number of output membership values are produced. The following layer consists of a single node (single process output) which performs the aggregation operation (summing of the membership values) and the defuzzification operation.

The details of the three fuzzy inference models are produced in Table 4.3.

Table 4.3: Fuzzy inference model details for the clinker grinding process

MODEL DETAILS	MODEL TYPE		
	FIS (MAMDANI)	FIS (SUGENO)	ANFIS
Inputs	3	3	3
Outputs	1	1	1
No of input MFs	[5 5 5]	[5 5 5]	[3 3 3]
No of output MFs	5	9	27
No of rules	63	45	27
AND method	Minimum	Product	Product
Aggregation method	Maximum	Sum	Sum
Defuzzification method	Centroid	Weighted average	Weighted average
Input MF type	Triangular and trapezoidal	Triangular and trapezoidal	Triangular
Output MF type	Trapezoidal	Constant	Linear

The PCA-BPNN model was constructed using the values of these two latent variables and the corresponding actual output values. The details of the optimum PCA-BPNN model is presented in Table 4.4.

Table 4.4: PCA-BPNN model details for the clinker grinding process

Number of nodes in input layer	2
Number of hidden layers	1
Number of nodes in hidden layer	35
Number of nodes in output layer	1
Activation function for hidden layer neurons	Logarithmic sigmoidal
Activation function for output layer neuron	Linear
Training algorithm	Gradient descent with momentum

4.2 Soft Sensor Development for Rotary Cement Kiln

4.2.1 Process Description

The main raw material for cement production is limestone ($CaCO_3$). The limestone is mixed in a proper ratio with ores of iron oxide, aluminum oxide and silica (clay). The mixture known as raw mix or raw meal is subsequently ground to fines in a raw mix grinder (usually a vertical roller mill used for this purpose). The ground raw meal then enters the clinker production process. The conversion of raw mix to clinker is known as pyroprocessing. The schematic diagram of the pyro-process is presented in Figure 4.10.

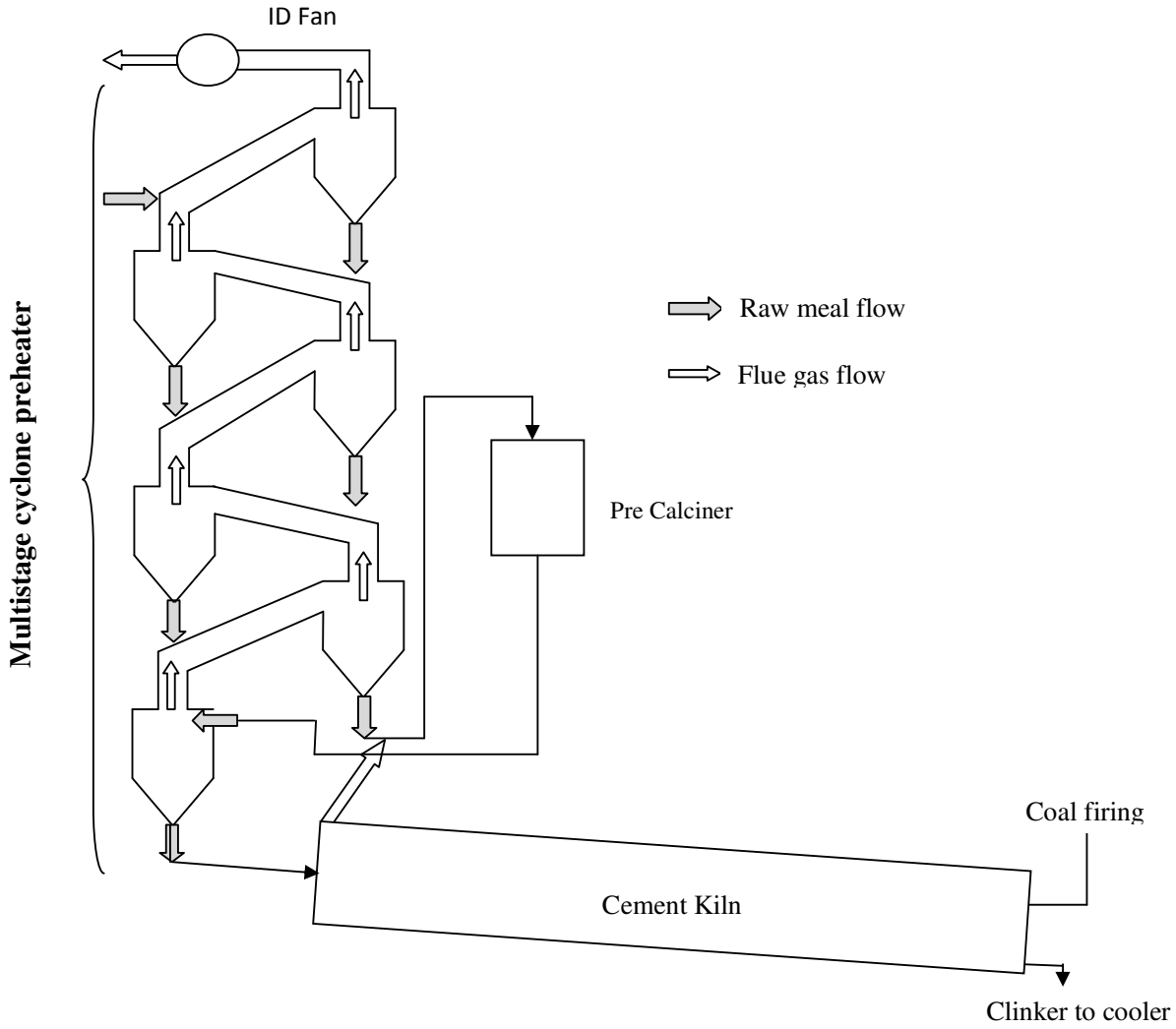


Figure 4.10: Process flow diagram for clinker production

The conversion of raw mix to clinker takes place through four stages: preheating, calcining, sintering and cooling. The preheating of the raw mix is accomplished in a multistage cyclone preheater. These cyclones are arranged in a tower above the feed end of the kiln. The raw mix enters into the inlet of the uppermost cyclone where the gas temperature is of the order of 500°C and during its movement sequentially and stagewise downwards, it gets heated by coming in contact with the hot flue gas coming from the kiln. Finally, the bottom cyclone discharges into the kiln itself. During this preheating operation the temperature of the raw mix powder rises to about 850°C. The calcination process is already initiated by the time the raw mix enters the kiln. Calcination process is described by the following reaction:

$CaCO_3 \rightarrow CaO+CO_2$

In modern cement plants, before entry to the kiln, a precalciner (additional combustion process) is used to augment the calcination process to decrease the processing load in the kiln. Bulk of the calcination reaction as described above is achieved during the vertical downward movement of the raw meal in the multistage cyclone preheater and in the precalciner.

In the kiln, the temperature of the feed further increases to around 1500^0C. At this high temperature, the CaO formed due to calcination, reacts with other components of raw mix to form complexes as mentioned below (Hewlett, 2003):

$2CaO+SiO_2 \rightarrow Ca_2SiO_4$ (Dicalcium silicate, CCN: C_2S or belite)

$C_2S+CaO \rightarrow Ca_3SiO_5$ (Tricalcium silicate, CCN: C_3S or alite)

$3CaO+Al_2O_3 \rightarrow Ca_3Al_2O_6$ (Tricalcium aluminate, CCN: C_3A or aluminate)

$4CaO+Al_2O_3+Fe_2O_3 \rightarrow Ca_4Al_2Fe_2O_{10}$ (tetraracalcium aluminoferrite, CCN: C_4AF or ferrite)

Here, CCN stands for 'Cement Chemistry Notation'. According to this notation the various oxides are denoted as below:

$CaO \rightarrow C$; $SiO_2 \rightarrow S$, $Al_2O_3 \rightarrow A$ and $Fe_2O_3 \rightarrow F$

Finally the hot clinker coming out from the kiln is cooled by contacting with air in the cooler. The product i.e. clinker quality is determined by offline laboratory analysis usually with a sampling period of 1 h. Subsequently based on the clinker free lime (f-CaO) content information received from the quality control laboratory, the kiln operator adjusts the various kiln operating parameters to maintain the free lime content within the prescribed limit. While free lime (f-CaO) is the single most important quality parameter for cement clinker, other quality parameters such as alite, belite, aluminate, ferrite, LSF, AlM and SiM also determine the quality of cement as mentioned in Section 2.3.2.

4.2.2 Data Collection and Treatment

Table 4.5 presents different input and output variables considered for modeling.

Table 4.5: Input and output variables for rotary cement kiln model

INPUTS		OUTPUTS
Raw meal quality	Kiln operating variables	Clinker quality
SiO_2 Al_2O_3 Fe_2O_3 CaO	Kiln feed rate Kiln RPM Raw meal inlet temperature Coal feed rate Kiln current	Free lime (f-CaO) Lime saturation factor (LSF) Silica modulus (SiM) Alumina modulus (AlM) C_3S (Alite) C_2S (Belite) C_3A (Aluminate) C_4AF (Ferrite)

The quality variables of raw meal quality and clinker quality are measured by offline laboratory analysis whereas the kiln operating parameters are measured online. Figure 4.11 shows the values of the kiln parameters as received from the plant over a period of one month and Table 4.6 presents the statistical characteristics of the raw data.

Table 4.6: Statistical characteristics of raw kiln data

KILN OPERATING VARIABLES	RANGE	STANDARD DEVIATION (σ)	SKEWNESS (γ)	KURTOSIS (κ)
Coal feed rate (tons/h)	16778	495	33.8	1142
Kiln current (Ampere)	14465	413	32.9	1105
Feed inlet temperature (^{0}C)	99135	6268	11.26	145
Kiln feed rate (tons/h)	556.24	80.5	-3.8	17.6
Kiln RPM	4.5	0.8	-3.1	13

It can be observed that the occasional peaks appearing in the plots are the outlying observations arising due to sensor malfunctions. Presence of outliers in the raw data is further confirmed in Table 4.6 which shows abnormally high values of range, standard deviation, skewness and kurtosis for the 5 kiln operating variables. These values must be properly detected and removed before using the data for model development. Both univariate and multivariate approaches for outlier detection were carried out.

The three univariate outlier detection techniques of 3 sigma method, box plot method and Hampel's method (described in Section 3.1.2.1) were applied to each variable dataset. Subsequently, the detected outliers are removed and the resulting missing values were imputed by linear interpolation. The statistical characteristics of the resulting datasets were studied to determine the best performing technique. The dataset resulting from application of Hampel's identifier was subsequently used to develop various neural network models of the kiln for estimation of clinker quality. All input-output data were normalized in the range 0 to 1 using Equation 3.9 prior to model development. The final input dataset (raw meal quality and kiln operating variables) containing values of all 9 input variables in the normalized form is shown in Figure 4.12.

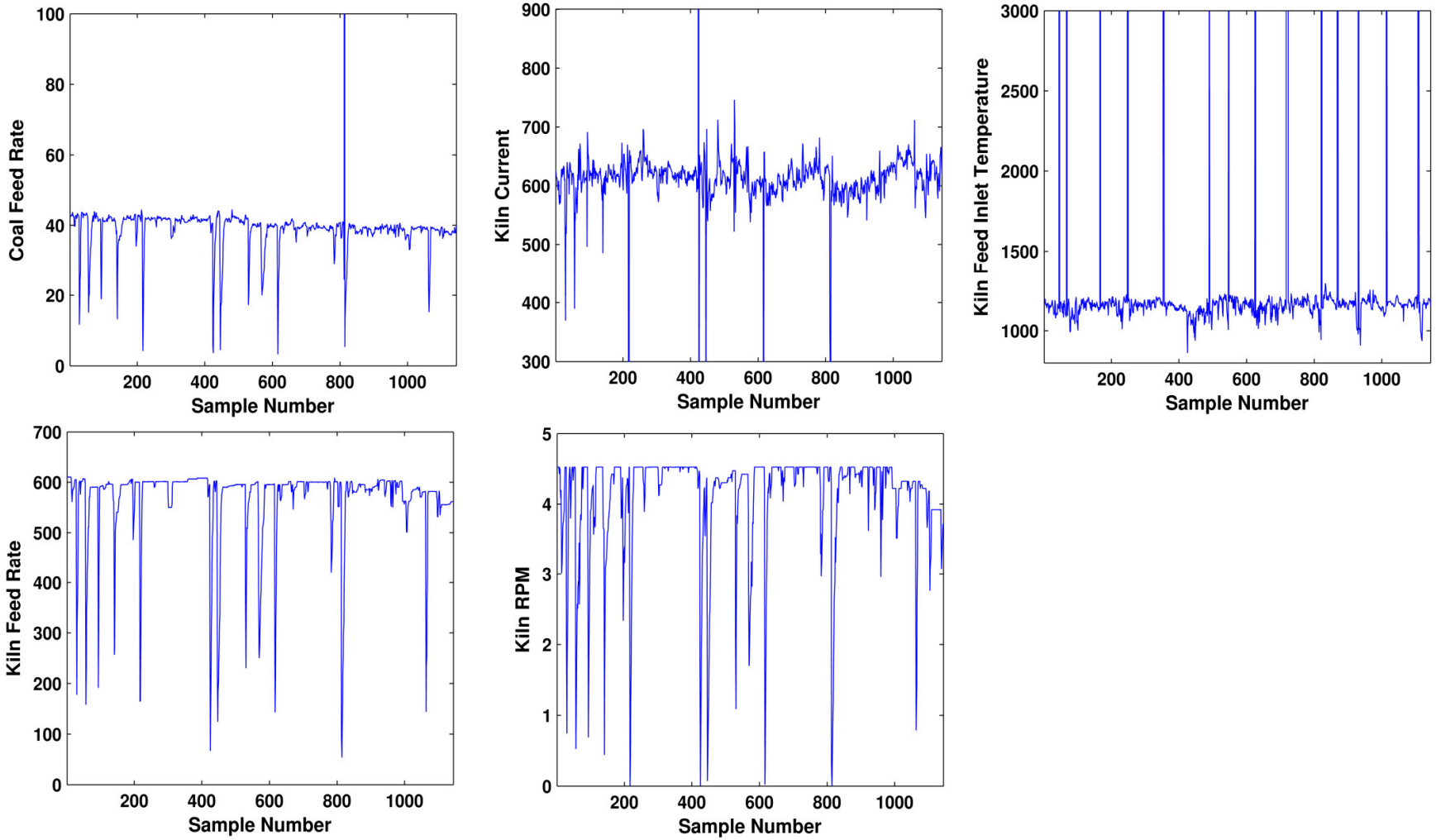


Figure 4.11: Actual raw data of kiln operating variables as received from the plant

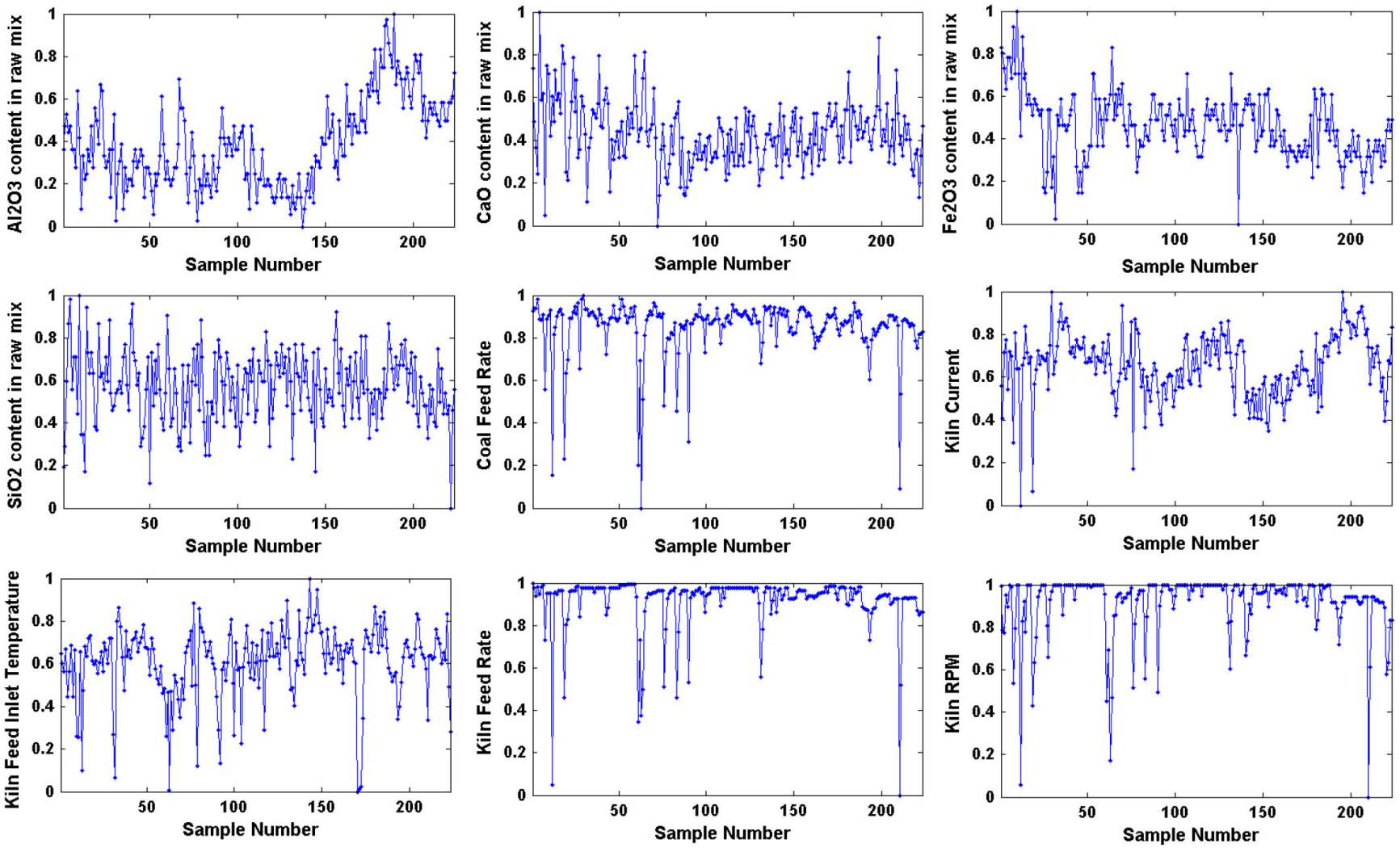


Figure 4.12: Final input data for neural network modeling (After applying univariate outlier detection method)

A total of 223 input – output datasets were prepared out of which 156 were used for training the neural network model and 67 for model validation. This data division was performed using random selection by allocating the first 156 data for training and the rest 67 samples for validation.

However, when a number of input variables are present and which are likely to be correlated, univariate outlier detection technique may not be effective. This is because, multivariate outliers may not be extreme values along a particular direction. This phenomenon is explained in Figure 4.13 by plotting values of two kiln operating variables i.e. coal feed rate and the raw mix input to the kiln. The two points highlighted are not extreme values either in x or y direction. So univariate outlier detection methods will not detect these points as outliers. However, the outlyingness of these two samples is quite obvious in the bivariate plot shown. Similarly many values in the initial part (x from 30 to 35) will be incorrectly shown as outliers in a univariate analysis because most of the values are concentrated in the region x from 35 to 40. However, a simultaneous decrease of both values indicates a practical phenomenon of slowdown of the operation. Therefore, univariate techniques applied to multivariate dataset may result in masking or swamping effects i.e. some multivariate outliers may not be detected or some normal values may be detected as outliers.

Five types of multivariate outlier detection techniques were applied on the multivariate dataset. These techniques include two classical methods (classical Mahalanobis distance and hat matrix leverage value method) and three robust methods (MCD, SHV and CDC). These techniques have already been described in Section 3.1.2.2. The method of MCD was implemented using the code available in Verboven and Hubert (2005) and MATLAB codes for other techniques were written which are provided in the Appendix. For comparison, the outliers detected in each technique were removed and the statistical characteristics of the resulting datasets were analyzed. Based on the analysis the

CDC method was found to be best performing (Detailed performance analysis of the techniques are presented in Table 5.3, Chapter 5).

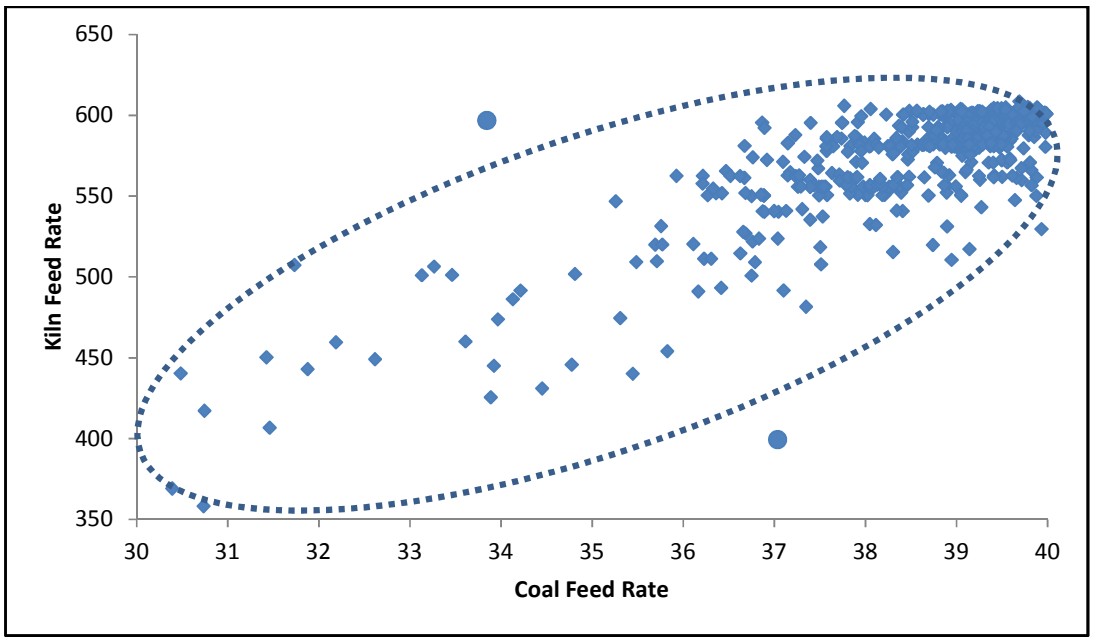


Figure 4.13: Bivariate scatter plot of two kiln operating variables explaining the multivariate nature of outliers

The number of data values obtained for the kiln operating variables were in excess of one thousand whereas the number of laboratory values obtained for raw meal and clinker quality were 223. Therefore, a total of 223 number of operating variable data values were selected based on the time instant at which the clinker quality data were available. For any time instant while the instantaneous values of clinker quality and kiln operating variables were selected, the previous values of raw meal quality data were selected. This is due to the fact that laboratory values are available only with a time lag of minimum 1 hour. Therefore, at any instant to predict the instantaneous clinker quality parameters, while the instantaneous values of kiln operating variables are available to the model, the same for raw meal quality values are not available and only the last reported values from the laboratory are available. Therefore any observation set for model development comprises of:

$$[y_1, y_2, \dots, y_8] = f\{x_1(t), \dots, x_5(t), x_6(t-1), \dots, x_9(t-1)\} \quad (4.10)$$

Here x_1, x_2, x_3, x_4 and x_5 are model inputs corresponding to kiln operating variables (second column of Table 4.5) and x_6, x_7, x_8 and x_9 are model inputs corresponding to input raw meal quality (first column of Table 4.5). y_1, y_2,.., y_8 are model outputs corresponding to clinker quality (third column of Table 4.5). The total data were normalized in the range of 0 to 1 for each process variable using the minimum and maximum values of that variable. The total processed dataset of 223 number of samples after normalization, were divided into a training set of 112 observations and a validation set of 111 observations. The data divsion was performed using Kennard-Stone (K-S) maximal intra distance criterion described in Section 3.3.2 and the code for implementation was adopted from the TOMCAT toolbox (Daszykowski *et al*., 2007).

4.2.3 Soft Sensor Model Development

The model development was done using two different sets of data. One set comprises of data available after univariate outlier detection followed by random division of the data into 156 training samples and 67 validation samples (Dataset 1). In this dataset, the raw meal quality data are also the instantaneous values. The second set of data was formed by using CDC method of robust multivariate outlier detection followed by application of Kennard-Stone method for division of the data to 112 training samples and 111 validation samples (Dataset 2). In this dataset the raw meal quality data are the time lagged values (sine at any instant, the laboratory values for the last drawn sample is available).

4.2.3.1 Modeling Using Dataset 1

For designing BPNN model, neural network models with one and two hidden layers were tested. The number of neurons was determined by conducting model training for different number of neurons ranging from 3 to 20 and choosing the one producing the least error.

The RBFNN model was developed using Gaussian function (Equation 3.35) in the RBF layer followed by linear activation function in the output layer. The desired error goal

was set to zero and maximum number of neurons to 70. After adequate number of simulations with different spread values, an optimum spread value of 0.4 was used for the RBFNN.

In the similar manner the optimum spread value for the GRNN model was determined to be 4.9. The number of hidden layers in a GRNN model is same as the number of input patterns i.e. 156 in this case.

The details of the three optimum neural network structures are presented in Table 4.6 and the model figures are presented in Figures 4.14-4.16.

Table 4.7: Neural network model details for the clinkerization process using univariate outlier detection and random data division

TYPE OF FEED FORWARD NEURAL NETWORK MODEL	NUMBER OF NEURONS	ACTIVATION FUNCTIONS USED
BPNN (Trained by conjugate gradient algorithm)	Input layer: 9 Hidden layer 1: 9 Hidden layer 2: 12 Output layer: 8	Hidden layer 1 and 2: Sigmoidal Output layer: Linear
RBFNN (Orthogonal least square algorithm)	Input layer: 9 RBF layer: 70 Output layer: 8	RBF layer: Gaussian Output layer: Linear
GRNN	Input layer: 9 Pattern layer: 156 Output layer: 8	Pattern layer: Gaussian Output layer: Linear

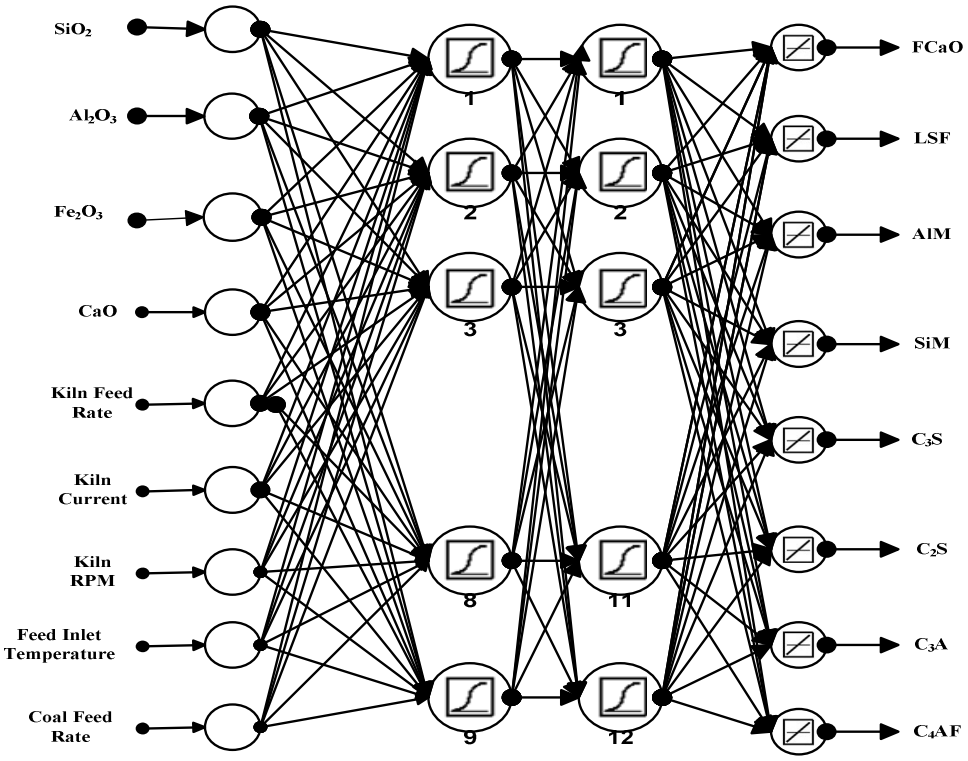


Figure 4.14: BPNN model structure for clinkerization process developed using dataset 1

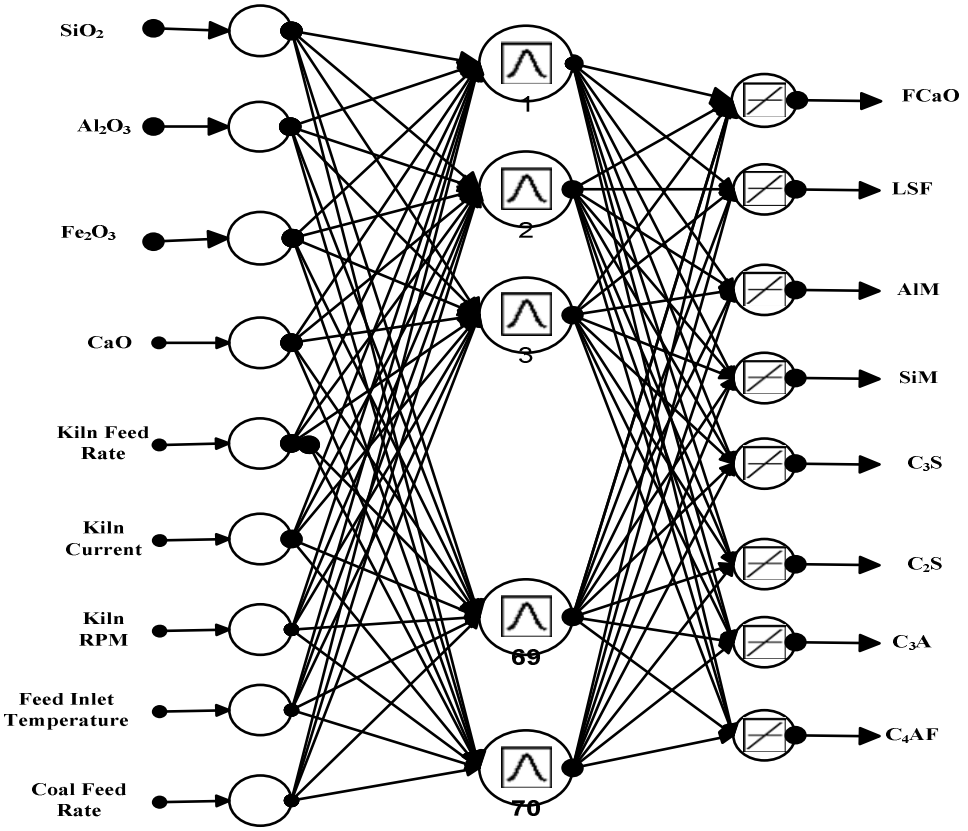


Figure 4.15: RBFNN model structure for clinkerization process developed using dataset 1

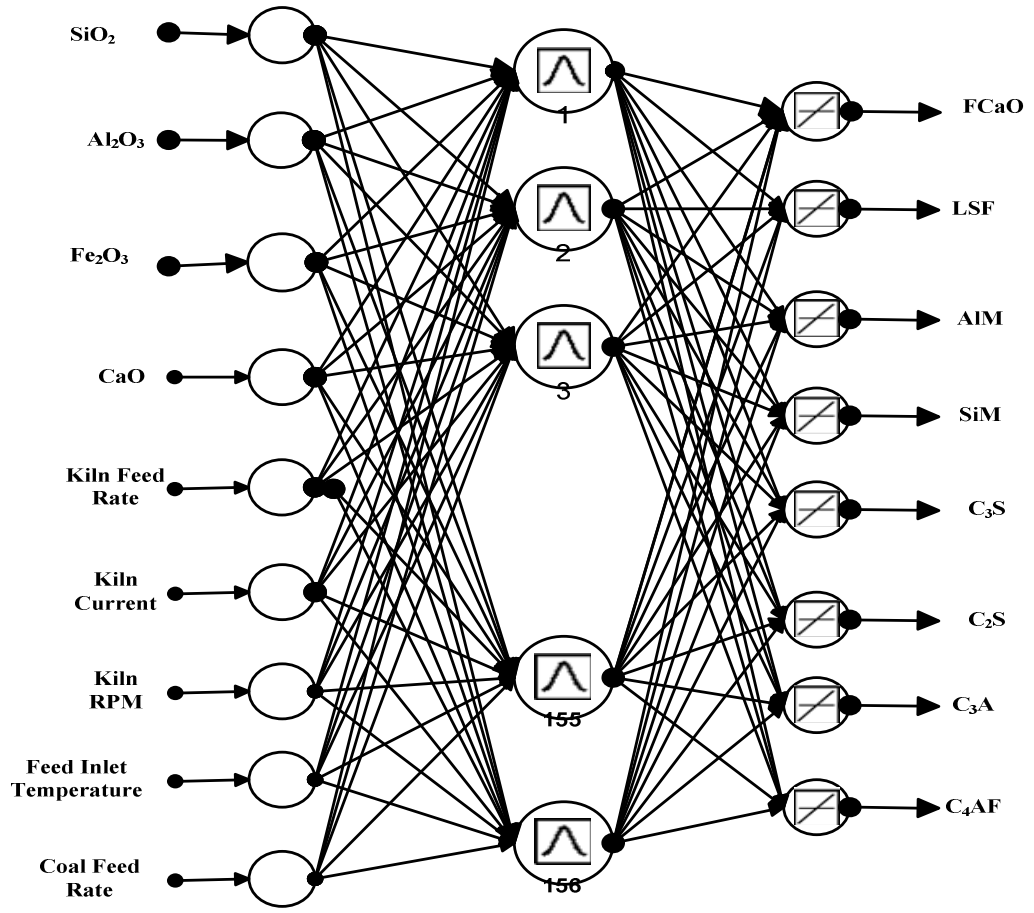


Figure 4.16: GRNN model structure for clinkerization process developed using dataset 1

4.2.3.2 Modeling Using Dataset 2

The training set prepared by using K-S method as described earlier was used for data-driven model development of rotary cement kiln. It has been mentioned in Section 2.3.2 that the model reported by Lin *et al.* (2007) for soft sensing of free lime has used different kiln operating variables as inputs and the input raw meal quality has not been considered in modeling. Therefore, before proceeding further with model development it was investigated whether the inclusion of raw meal quality as inputs improves the model performance or not. To address this issue, initially two types of multiple linear regression models for the kiln were developed, one only with kiln operating variables as inputs (Total 5 inputs) and the other using kiln operating variables as well as raw meal quality as inputs (Total 9 inputs). The linear regression models using the two possible set of inputs are presented in Table 4.7.

Table 4.8: Multiple linear regression model of the cement kiln with and without raw meal quality

NUMBER OF INPUTS	REGRESSION EQUATIONS
5 (Only kiln operating variables)	$y_1 = 0.131 - 0.05x_1 - 0.02x_2 + 0.28x_3 - 0.043x_4 + 0.027x_5$ $y_2 = 0.301 - 0.05x_1 - 0.02x_2 + 0.148x_3 - 0.123x_4 + 0.022x_5$ $y_3 = 0.139 + 0.015x_1 + 0.047x_2 + 0.009x_3 - 0.046x_4 + 0.037x_5$ $y_4 = 0.56 + 0.044x_1 + 0.061x_2 - 0.111x_3 - 0.351x_4 + 0.057x_5$ $y_5 = 0.384 - 0.037x_1 + 0.001x_2 + 0.063x_3 - 0.07x_4 + 0.026x_5$ $y_6 = 0.415 + 0.0004x_1 + 0.0356x_2 - 0.09x_3 + 0.182x_4 - 0.079x_5$ $y_7 = 0.33 + 0.067x_1 + 0.05x_2 - 0.022x_3 - 0.223x_4 + 0.003x_5$ $y_8 = 0.55 - 0.03x_1 - 0.12x_2 - 0.0022x_3 + 0.33x_4 - 0.14x_5$
9 (Kiln operating variables + Raw meal quality)	$y_1 = 0.297 + 0.0318x_1 - 0.0132x_2 - 0.178x_3 - 0.0308x_4 - 0.124x_5 - 0.062x_6 + 0.2053x_7 + 0.018x_8 + 0.0405x_9$ $y_2 = 0.2834 + 0.0128x_1 + 0.1627x_2 - 0.0048x_3 - 0.0734x_4 - 0.1428x_5 - 0.0384x_6 + 0.1x_7 - 0.0953x_8 + 0.0755x_9$ $y_3 = 0.2708 - 0.024x_1 - 0.1236x_2 - 0.1024x_3 - 0.0253x_4 - 0.0135x_5 - 0.002x_6 + 0.011x_7 - 0.0296x_8 + 0.0284x_9$ $y_4 = 0.5844 - 0.0537x_1 + 0.2351x_2 - 0.2794x_3 + 0.0363x_4 - 0.059x_5 - 0.0504x_6 + 0.097x_7 - 0.1936x_8 + 0.0603x_9$ $y_5 = 0.2982 + 0.0208x_1 + 0.0786x_2 + 0.069x_3 - 0.051x_4 - 0.0536x_5 - 0.0634x_6 + 0.077x_7 - 0.0815x_8 + 0.0474x_9$ $y_6 = 0.573 - 0.0276x_1 - 0.1748x_2 - 0.1607x_3 + 0.1056x_4 + 0.0284x_5 - 0.069x_6 - 0.0942x_7 + 0.2131x_8 - 0.1165x_9$ $y_7 = 0.2586 - 0.0025x_1 + 0.2548x_2 - 0.1434x_3 + 0.0402x_4 - 0.0136x_5 + 0.051x_6 + 0.0042x_7 - 0.1213x_8 + 0.0007x_9$ $y_8 = 0.338 + 0.0641x_1 + 0.0032x_2 + 0.2961x_3 - 0.0177x_4 + 0.047x_5 - 0.0544x_6 + 0.0211x_7 + 0.199x_8 - 0.104x_9$

Here, y_1 to y_8 are the clinker quality parameters in the order in which they appear in column 3 of Table 4.5. In the first set of model equations, x_1 to x_5 are kiln operating parameters in the order in which they appear in second column of Table 4.5. In the second set of model equations, x_1 to x_5 are same as mentioned before and x_6 to x_9 are time lagged values of raw meal quality in the order in which they appear in the first column of Table 4.5. It is be shown subsequently in Section 5.3.2.1 that, inclusion of raw meal quality parameters as model inputs leads to significant improvement in model

performance. Therefore, subsequently the nine input variables (instantaneous values of kiln operating parameters and time lagged values of raw meal quality) are used for data-driven model development.

With the available training set, BPNN, RBFNN and GRNN models of the rotary cement kiln were developed in the same manner as described in Section 4.1.3. The details of the three optimum models developed are presented in Table 4.8.

Table 4.9: Neural network model details of the clinkerization process using multivariate outlier detection and K-S method of training data selection

TYPE OF FEED FORWARD NEURAL NETWORK MODEL	NUMBER OF NEURONS	ACTIVATION FUNCTIONS USED
BPNN (Trained by gradient descent with momentum algorithm)	Input layer: 9 Hidden layer: 15 Output layer: 8	Hidden layer 1 and 2: Sigmoidal Output layer: Linear
RBFNN (Orthogonal least square algorithm)	Input layer: 9 RBF layer: 14 Output layer: 8	RBF layer: Gaussian Output layer: Linear
GRNN	Input layer: 9 Pattern layer: 112 Output layer: 8	Pattern layer: Gaussian Output layer: Linear

The general structure of the fuzzy inference model has been presented in Figure 3.6 and the model structure for the grinding process has been presented in Figure 4.7. The fuzzy model structure for the rotary cement kiln is shown in Figure 4.17. The important steps involved in model development have already been explained in Section 3.6. Two types of fuzzy inference models were designed for the kiln system: Mamdani type FIS and Sugeno type FIS.

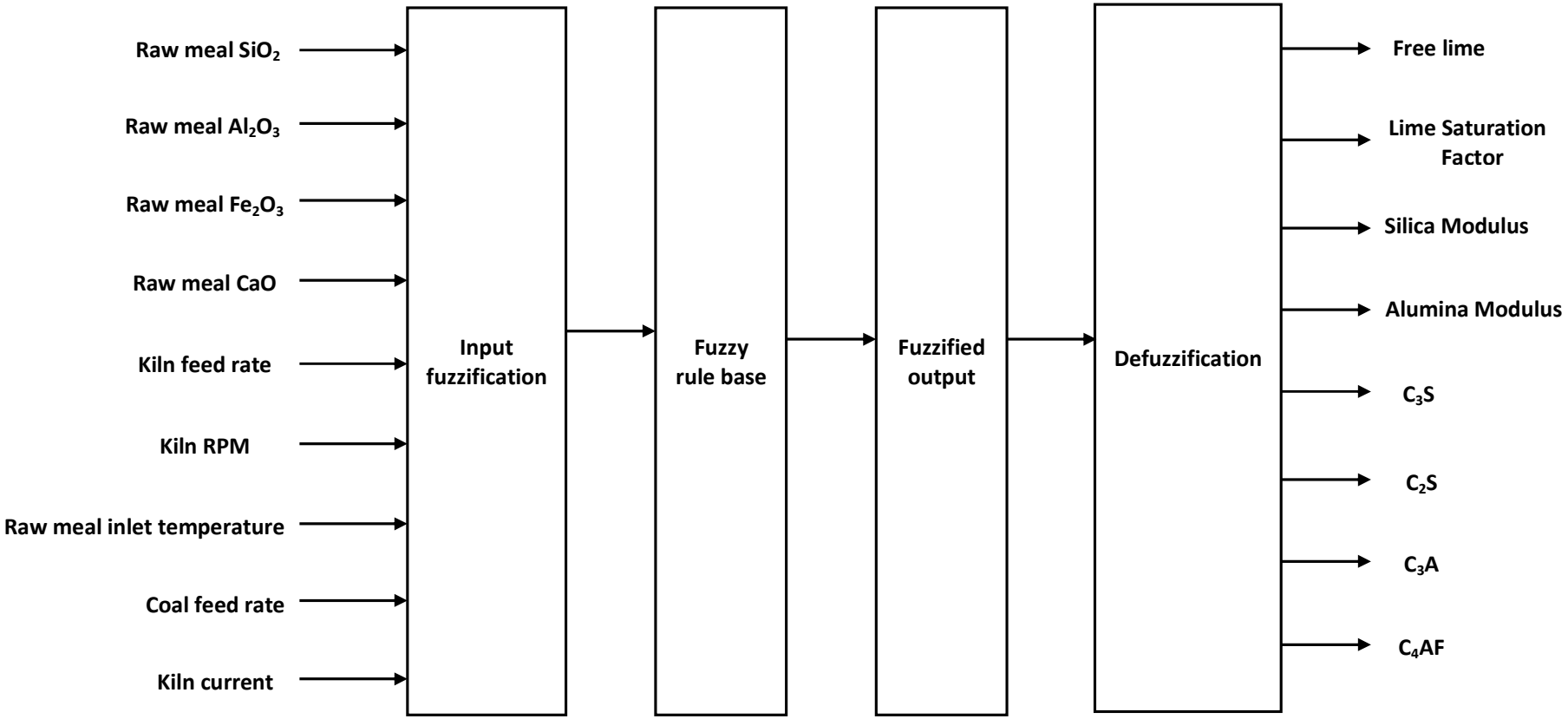


Figure 4.17: Fuzzy inference system (FIS) for the rotary cement kiln

The normalized values of the nine inputs and eight output variables were transformed to fuzzy linguistic variables by applying a membership function. For Mamdani fuzzy model, the 8 output variables i.e. cement clinker quality parameters were fuzzified with trapezoidal membership function. For Sugeno type model, 8 constant output values in the range 0 to 1 were chosen.

Five linguistic variables were used for each process variable (input or output for Mamdani model and inputs for Sugeno model). The linguistic variables are listed below along with their ranges:

VS (very small): [0 - 0.2]

S (small): [0.1 - 0.45]

M (medium): [0.3 - 0.65]

L (large): [0.5 - 0.85]

VL (very large): [0.7 - 1]

Triangular function is applied to the variables of 'very small' and 'very large' category and trapezoidal function for the other types of linguistic variables as described above.

The eight constant output values for Sugeno fuzzy modeling are mentioned below:

ES (extremely small): 0.05

VS (very small): 0.15

S (small): 0.25

LM (low medium) 0.35

M (medium): 0.45

L (large): 0.65

VL (very large): 0.75

EL (extremely large): 0.9

Figure 4.18 explains the fuzzification process for one input and one output (for Mamdani type).

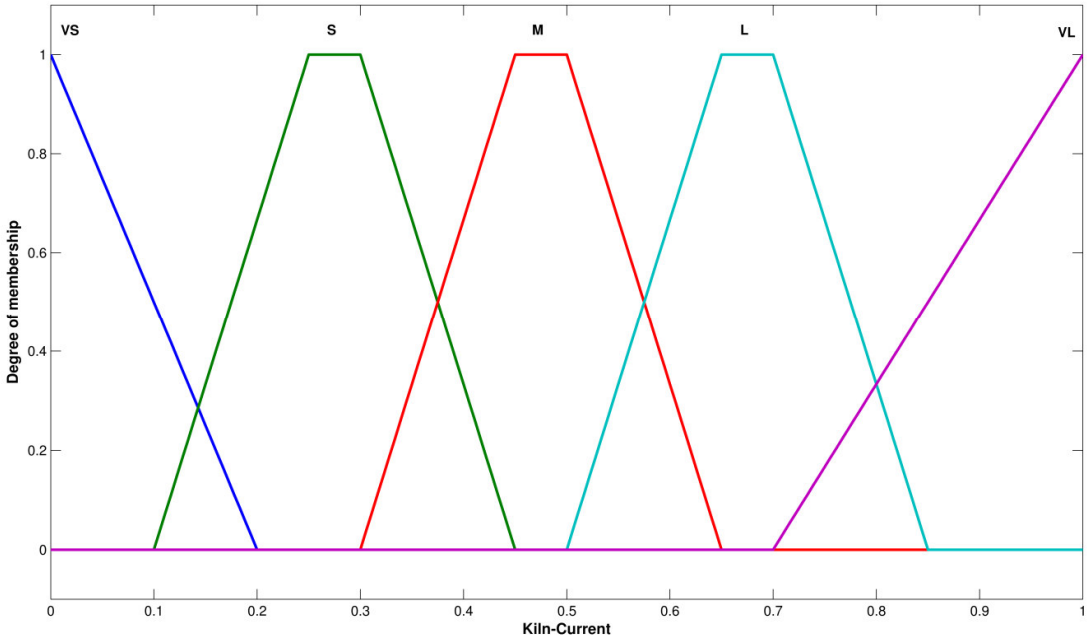


Figure 4.18 (a) Input fuzzification for the variable 'Kiln Current'

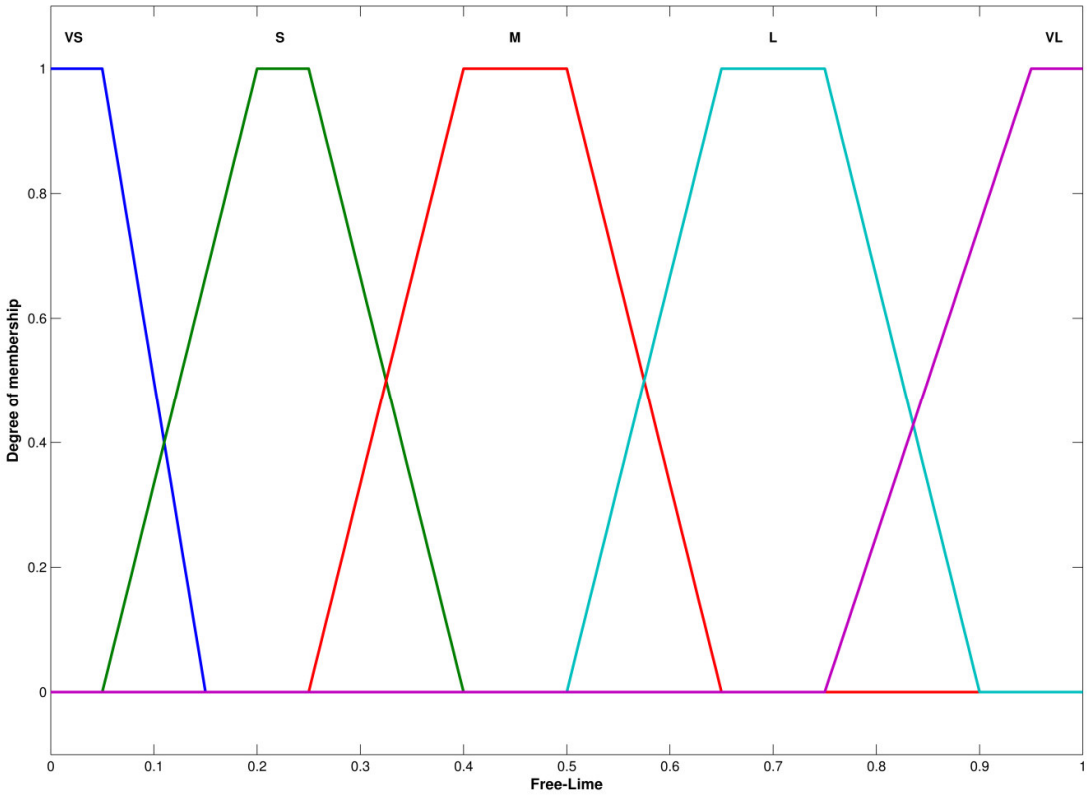


Figure 4.18 (b) Output fuzzification for the variable 'Free lime'

The rule base was framed based on the available input - output dataset for model development. A total of 147 rules were framed for the Mamdani model and 129 fuzzy *if-then* rules were framed for the Sugeno type model.

Since nine process inputs are involved, for each data sample the antecedent of the corresponding fuzzy rule has nine inputs. AND operator is used to obtain the result of the rule as mentioned below:

$$\mu(x) = min\{\mu(x_1), \mu(x_2), \mu(x_3), \mu(x_4), \mu(x_5), \mu(x_6), \mu(x_7), \mu(x_8), \mu(x_9)\} \text{(Mamdani)} \quad (4.11)$$

$$\mu(x) = product\{\mu(x_1), \mu(x_2), \mu(x_3), \mu(x_4), \mu(x_5), \mu(x_6), \mu(x_7), \mu(x_8), \mu(x_9)\} \text{(Sugeno)} \quad (4.12)$$

The fuzzified output is determined by composition of fuzzified inputs and the appropriate rules from the rule base.

One input value had more than one fuzzy membership values under different liguistic variables. Therefore, for one set of input data, multiple rules are applied producing multiple number of fuzzy outputs. Aggregation was performed by combining different fuzzy output membership values to produce a single fuzzy output. This is a maximum method which involves fuzzy union of all possible fuzzy outputs. Finally, the fuzzy output value produced for a particular observation was defuzzified in order to get the actual process output value. Centroid and weighted average method mentioned in Equations 3.48 and 3.49 were used as the method of defuzzification for Mamdani and Sugeno type FIS respectively.

For development of hybrid model combining principal component analysis and back propagation neural network, initially PCA was performed on the 223×9 input data matrix. Subsequently using cumulative percentage variance criterion 5 principal components were selected which are the linear combinations of all nine input variables. BPNN was developed using this reduced input dataset consisting of five latent variables instead of the nine actual input variables.

Chapter - 5

Results and Discussion

The theory and procedure of soft sensor development for the clinker grinding process and pyro process are described in the preceding Chapters 3 and 4. The results obtained for the two processes in each step of the soft sensor development starting from data analysis and pretreatment to model development and analysis are presented in this chapter.

5.1 Performance Analysis of Outlier Detection Techniques

The performances of different outlier detection techniques (univariate and multivariate) were analyzed by evaluating various statistical parameters such as standard deviation, skewness, kurtosis, range etc. The statistical parameters were determined initially for the raw dataset as for the processed datasets. The processed datasets were obtained after the application of a particular outlier detection technique, removal of the outliers detected by a technique followed by imputation of missing values by linear interpolation with the neighboring values.

5.1.1 Performance of Univariate Outlier Detection Methods

The skewness value of a dataset is defined as:

$$\gamma = \frac{\sum(x_i - \overline{x})^3}{N\sigma^3} \tag{5.1}$$

The kurtosis is defined as:

$$\kappa = \frac{\sum(x_i - \overline{x})^4}{N\sigma^4} \tag{5.2}$$

Table 5.1 presents the statistical characteristics of the resulting datasets for the clinker grinding process.

Table 5.1: Descriptive statistics of the input variables after using different outlier detection methods for the grinding process

VARIABLES	OUTLIER DETECTION METHOD	MINIMUM	MAXIMUM	AVERAGE	STANDARD DEVIATION (σ)	SKEWNESS (γ)	KURTOSIS (κ)
Hot Air Flow	3 sigma rule	538273	706023.8	677660.15	16107	-4.32	31.282
	Hampel's rule	657680.31	706023.8	680091.88	8285.72	0.152	2.919
	Box plot	608619.25	706023.8	679609.4	9805.87	-1.59	13.125
Classifier RPM	3 sigma rule	427.786	1114.133	906.354	134.605	-0.84	3.084
	Hampel's rule	563.165	1114.133	909.214	129.27	-0.72	2.658
	Box plot	427.786	1114.133	906.354	134.605	-0.84	3.084
Clinker inflow	3 sigma rule	76	239.828	183.64	28.2	-1.31	6.2
	Hampel's rule	147.43	227.427	188.33	16.05	0.21	2.728
	Box plot	103.6	239.82	185.92	23.27	-0.59	4.67

Skewness is a measure of symmetry of the dataset and for a normally distributed data, the skewness value is zero. Similarly, kurtosis represents the extent of peakedness or flatness of the dataset and for a normally distributed data has a value of 3. Presence of outliers in a dataset results in higher standard deviation, large deviation of skewness and kurtosis values from 0 and 3 respectively.

A closer comparison on the nature of raw data presented in Table 4.1 with that of cleaned data in Table 5.1 shows the significant improvement over raw data when outlier removal techniques are applied. The methods of 3 sigma rule and box plot are influenced by the presence of outliers in the data. On the other hand, the Hampel's method that uses outlier resistant median and MAD values is less affected by the outliers present in the original dataset. It can be noticed in Table 5.1 that Hampel's method has resulted in the least standard deviation and skewness and kurtosis values closest to 0 and 3 respectively. Table 5.1 establishes the superiority of Hampel method over the other two methods. Table 5.2 presents the statistical characteristics of the resulting datasets for the rotary cement kiln.

Table 5.2: Descriptive statistics of the input variables after using different outlier detection methods for the clinkerization process

KILN OPERATING PARAMETERS	VALUES FOR TREATED DATA					
	3 SIGMA METHOD		HAMPEL'S METHOD		BOX PLOT RULE	
	γ	κ	γ	κ	γ	κ
Kiln feed rate	-3.79	19.119	-1.181	4.12	-1.669	5.658
Kiln current	-7.318	75.024	-0.056	2.676	0.068	4.716
Kiln RPM	-2.5	9.497	-1.317	3.923	-1.892	6.274
Kiln feed inlet temperature	14.34	227.488	-0.624	3.51	-1.038	4.7
Coal feed rate	-3.862	21.554	-0.671	4.321	-1.429	6.47

The raw data characteristics for the kiln processes have been presented in Table 4.6. Comparison of the values for the raw data in Table 4.6 with the values for the processed data in Table 5.2 indicates that all three methods have resulted in improvement of statistical characteristics of the input data. However, it can be noticed that Hampel's method again produces the best result in comparison to the other two methods. The better performance of the Hampel's method is due to the fact that this method is outlier resistant due to the use of median and MAD values (Equation 3.1) which are outlier insensitive. On the other hand, performance of the 3 sigma method is least satisfactory because this method uses the values mean and standard deviation which are heavily affected by the presence of outliers.

The modified datasets for the kiln process obtained using the three univariate outlier detection techniques are plotted in Figure 5.1 where it can be verified that the clean data obtained by application of 3 sigma method still contain a number of outliers.

5.1.2 Performance of Multivariate Outlier Detection Methods

The multivariate nature of the input data for the clinkerization process has been explained in Section 4.2.2. For detection and removal of multivariate outliers present in the raw dataset, two classical methods (classical Mahalanobis distance and hat matrix leverage value method) and three robust methods minimum covariance determinant (MCD), smallest half volume (SHV) and closest distance to center (CDC) were separately applied on the original dataset. The performances of the different methods were analyzed

by evaluating the statistical characteristics of the dataset obtained after application of each technique. The results are presented in Table 5.3.

The traditional methods of Mahalanobis distance and Hat matrix leverage values perform poorly as compared to the other three robust methods. It is difficult to conclude the superiority of one robust method over others in general. In this work, the best values obtained under a particular statistical parameter is highlighted.

Table 5.3 indicates superior performance of SHV technique over other methods. This is quite understandable if we look at the total number of outliers detected. SHV method detects more than 40% of the data as outliers! It is very hard to believe that all the sensors were malfunctioning for almost 15 days during that 1 month period. SHV seems to be too conservative and slightly deviated values are detected as outliers. It is quite possible that following SHV method if we remove half of the values as outliers, we may be losing important process information and practical applicability of the model will be very limited to the small range of process values (because data-driven models do not extrapolate well).

On the other hand, CDC method detects around 17% values as outliers and also produces statistical performance almost at par with SHV method. Removing 17% data values is acceptable and we will be able to have an increased domain of applicability of the developed soft sensor. Therefore based on this reasoning the processed dataset obtained after application of CDC method was used for subsequent modeling.

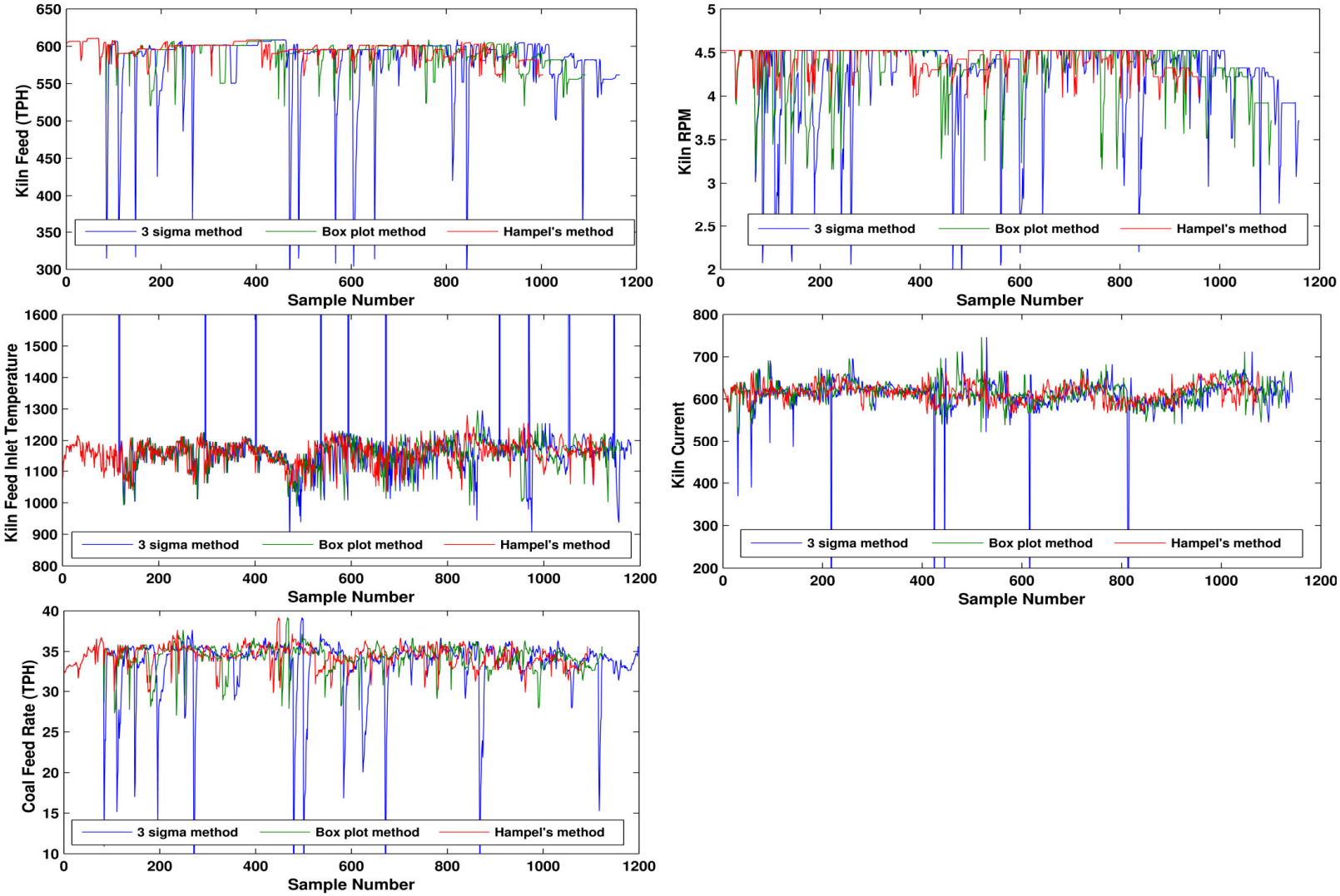


Figure 5.1: Comparison of the three outlier detection techniques for the rotary cement kiln data

Table 5.3: Performance of multivariate outlier detection techniques

OUTLIER DETECTION METHOD	NO OF OUTLIERS DETECTED	DESCRIPTIVE STATISTICS FOR VARIABLES AFTER DELETION OF OUTLIERS USING DIFFERENT METHODS																			
		COAL FEED				CURRENT				INLET TEMPERATURE				KILN FEED				RPM			
		RANGE	**σ**	**γ**	**κ**	**RANGE**	**σ**	**γ**	**κ**	**RANGE**	**σ**	**γ**	**κ**	**RANGE**	**σ**	**γ**	**κ**	**RANGE**	**σ**	**γ**	**κ**
Raw Data		16778	495	33.8	1142	14465	413	32.9	1105	99135	6268	11.26	145	556.24	80.5	-3.8	17.6	4.5	0.8	-3.1	13
Mahalanobis Distance	66	32.7	3.13	-3.2	19	193	23.7	-0.1	1.25	15103	747	14.4	232	315	43.5	-3.8	19.9	2.86	0.47	-2.8	11.8
Hat Matrix Leverage Value	85	27.1	2.65	-2.8	17.8	193	22.9	-0.1	4	15103	753.5	14.24	228	252.4	33.75	-3.6	18.8	2.47	0.39	-2.5	10.2
Closest Distance to Centre (CDC)	195	9.11	1.66	-0.125	2.28	151	21	**-0.016**	3.38	585.2	47	-0.97	8.88	73.2	14.22	-1.46	4.66	0.9	0.2	**-1.56**	**4.81**
Smallest Half Volume (SHV)	486	**7**	**1.42**	**-0.06**	2	89.9	**16.47**	-0.09	**2.85**	118.2	**24**	**-0.12**	**2.5**	40	**7.96**	**-0.74**	**3**	0.67	**0.11**	-1.88	7
Minimum Covariance of Determinant (MCD)	287	9.11	1.6	-0.15	**2.3**	130	19.8	-0.06	3.2	267.9	36.5	-0.83	4.24	69.9	12.7	-1.47	4.97	0.8	0.16	-1.63	5.25

5.2 Performance Analysis of Training Set Design Techniques

The available dataset was split into two parts. The training sets were used for development of data based model and the performances of the developed models were evaluated by analyzing their results with the validation sets. The effect of training data selection on the model performance has been mentioned in Section 3.3. Both random as well as algorithm based subset selection was carried out.

For the clinker grinding process, two types of random selection were performed. In random selection 1, out of 158 total available observations, the first 108 values were allocated for model development and the rest 50 were used for model performance assessment. In random selection 2, the total data were split equally to training and validation subsets each having 79 number of samples. 79 random values (corresponding to sample numbers) were chosen in the range of 1 to 158 using 'Multiplicative lagged Fibonacci generator' algorithm available in MATLAB for formation of training set. After formation of training set, the rest 79 samples were used as validation set.

Two popular statistical techniques were used for algorithm based subset selection. The Kennard-Stone maximal intra distance criterion and the DUPLEX algorithm. The procedural details of these two algorithms have already been presented in Sections 3.3.2 and 3.3.3, respectively. The statistical characteristics of the total dataset as well as the training sets formed using the aforementioned techniques are presented in Table 5.4.

It has been emphasized earlier that a major problem associated with data based models is the problem of extrapolation. The model must not be used with any input values which falls outside the range of inputs used for model development. In other words, the maximum possible range of data should be included in the training dataset. It can be observed in Table 5.4 that while both K-S and DUPLEX algorithms perform better than the random selection in retaining the total data characteristics in the training set, K-S

algorithm ensures maximum coverage of the entire dataset for all three concerned variables. However, both algorithms perform almost at par as far as the location and dispersion of the datasets are concerned.

Table 5.4: Performance of the three subset selection methods

DATA TYPE	INPUT VARIABLE	MINIMUM	MAXIMUM	RANGE	MEAN	STANDARD DEVIATION
Total Input Data (158 samples)	Air Flow	0	1	1	0.4833	0.1787
	Classifier RPM	0	1	1	0.6022	0.2544
	Clinker Inflow	0	1	1	0.5154	0.2
Training Data by Random Selection (79 samples)	Air Flow	0.182	1	0.818	0.539	0.1862
	Classifier RPM	0	0.892	0.892	0.496	0.273
	Clinker Inflow	0	0.964	0.964	0.572	0.224
Training Data by duplex algorithm (79 samples)	Air Flow	0.006	1	0.994	0.4696	0.176
	Classifier RPM	0	1	1	0.6162	0.246
	Clinker Inflow	0.1395	0.9448	0.805	0.5048	0.1926
Training data by Kennard-Stone algorithm (79 samples)	Air Flow	0	1	1	0.4966	0.2
	Classifier RPM	0	0.991	0.991	0.5489	0.2676
	Clinker Inflow	0	1	1	0.506	0.2253

The performance of the three data splitting methods are further analyzed by developing multiple linear regression models based on the three different training sets resulting from the three data splitting methods. The results are produced in Table 5.5.

Table 5.5: Effect of different data splitting methods on multiple linear regression model performance

DATA SPLITTING METHOD	MEAN ABSOLUTE ERROR (MAE)	
	TRAINING SET	VALIDATION SET
Random	0.1072	0.1019
Duplex algorithm	0.093	0.112
Kennard-Stone algorithm	0.1174	0.0971

The results of Table 5.5 indicate that, while the K-S method produces the maximum error for the training set, it produces the minimum error for the validation set. This is due to the difference in nature of functioning of the two algorithms. While K-S algorithm

results in formation of only the training dataset, the DUPLEX algorithm simultaneously forms the training as well as the validation datasets from the total data. Therefore, the range of the dataset is higher in case of the K-S algorithm. As data range increases, the fitness becomes poorer.

However, a model developed from a larger range of data is likely to generalize better than a model developed from a smaller range which will frequently have extrapolation problems. The better generalization behaviour of the K-S based model in comparison to other types of models is quite evident when one looks at the validation error produced by the different models. Moreover, a good model is considered as one which produces good result for the unknown data. Therefore, it can be concluded from Table 5.4 and Table 5.5 that the Kennard-Stone method of data splitting performs better than that of random selection method and DUPLEX method and hence the training dataset obtained by using K-S method was used subsequently for development of various types of data-driven models.

5.3 Performance Analysis of Developed Soft Sensors

While the parameters of linear and principal regression models were determined based on least of sum of squared error criterion, other models were developed so as to minimize the mean absolute error (MAE) between the actual and model predicted values. The MAE is defined as

$$MAE = \frac{\sum_{i=1}^{N} \left| y_i - \hat{y}_i \right|}{N} \tag{5.3}$$

5.3.1 Results for Clinker Grinding Process

A closer look at the regression coefficients of the linear equation (Equation 4.1) indicates that the values are quite in agreement with the physical understanding of the

process. The sign of a particular coefficient indicates the nature of relation between the dependent and the independent variables (i.e. direct or inverse) and the magnitude indicates the extent of dependence. The equation reveals that the classifier RPM affects the fineness the most. Positive coefficient indicates a direct relationship. This can be physically interpreted quite well. At higher RPM, most of the coarse particles will not be able to pass through the classifier resulting in an increased fineness of the product. Similarly, when hot air flow rate through the mill is increased more particles will be forced through the classifier resulting in a drop in cement fineness.

For design of standard SVR model, the error bound ε and the RBF kernel width σ were determined by performing grid search followed by cross validation on the validation set so as to produce the least error on both the training set and the validation set. The search was done first by performing a coarse grid search followed by a fine grid search as suggested in the literature (Gunn, 1998). The ranges of values used for the search operation are mentioned below:

Coarse grid search $\sigma = [2^{-4}, 2^{-2}, 2^{0}, 2^{2}]$

$\varepsilon = [2^{-8}, 2^{-6}, 2^{-4}, 2^{-2}]$

Fine grid search $\sigma = [2^{-2}, 2^{-1.875}, 2^{-1.75}, 2^{-1.5}, 2^{-1}]$

$\varepsilon = [2^{-4}, 2^{-3.875}, 2^{-3.75}, 2^{-3.5}, 2^{-3}]$

The results of the grid search operation for selected parameters are produced in Figure 5.2.

A closer look at Figure 5.2 verifies the SVR theory on the effect of one SVR hyper-parameter on another. It can be seen that for both training and validation data, with increase in the value of error bound ε, the model performance error also increases with a decrease in the number of support vectors. From grid search, two sets of optimum SVR parameters were obtained with different number of support vectors as given below:

SVR_1: $[C, \varepsilon, \sigma] = [1.4608, 0.068157\ (2^{-3.875}), 0.2726\ (2^{-1.875})]$ (5.4 a)

Number of support vectors (NSV) = 45 (56.9%)

SVR_2: $[C, \varepsilon, \sigma] = [1.4608, 0.003906\ (2^{-8}), 0.2726\ (2^{-1.875})]$

Number of support vectors (NSV) = 77 (97.4%).

For development of LS-SVR model of the grinding process from the training data, rigorous grid search was performed for optimum SVR hyper-parameter selection. The domain of grid search is mentioned below:

Coarse grid search: $C = [2^{-5},2^{-3},2^{-1},2^{1},2^{3},2^{5},2^{7}]$

$\sigma = [2^{-5},2^{-3},2^{-1},2^{1},2^{3},2^{5},2^{7}]$

The results of coarse grid search are displayed in Figure 5.3(a) and 5.3(b) for the training and validation data respectively.

It can be noticed that for the data used for modeling (training data), there is almost a monotonic increase in the model error values with increase in kernel width (σ) and decrease in error value with increase in value of regularization parameter (C). Moreover, the effect of C is more prominent at lower values of kernel width and error variation with C continuously reduces as the value of σ increases resulting almost in convergence at high kernel width. Figure 5.3 (a) indicates that the best values will be the highest possible values for C and lowest possible values for σ. However, such a selection though results in very accurate fitting, produces poor generalization capability of the model as is evident from Figure 5.3(b). It can be observed in Figure 5.3(b) that the set of C and σ values producing very low error values for the training data (Figure 5.3(a)), produce unacceptably high error values for the validation data. The optimum range corresponds to the middle portion of Figure 5.3(b) producing a dip in error value.

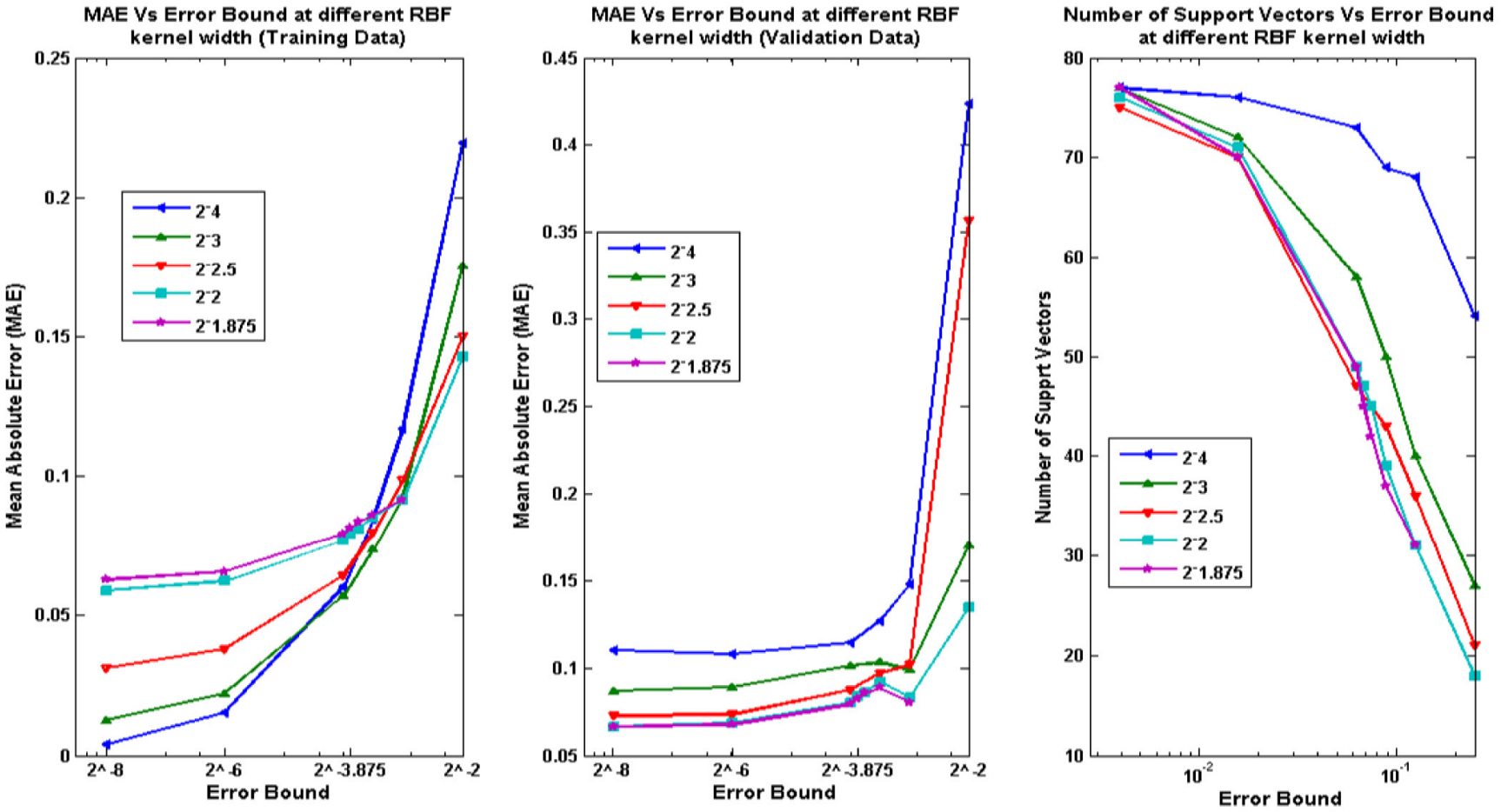


Figure 5.2: Results of the grid search and cross validation for optimal selection SVR hyper-parameters

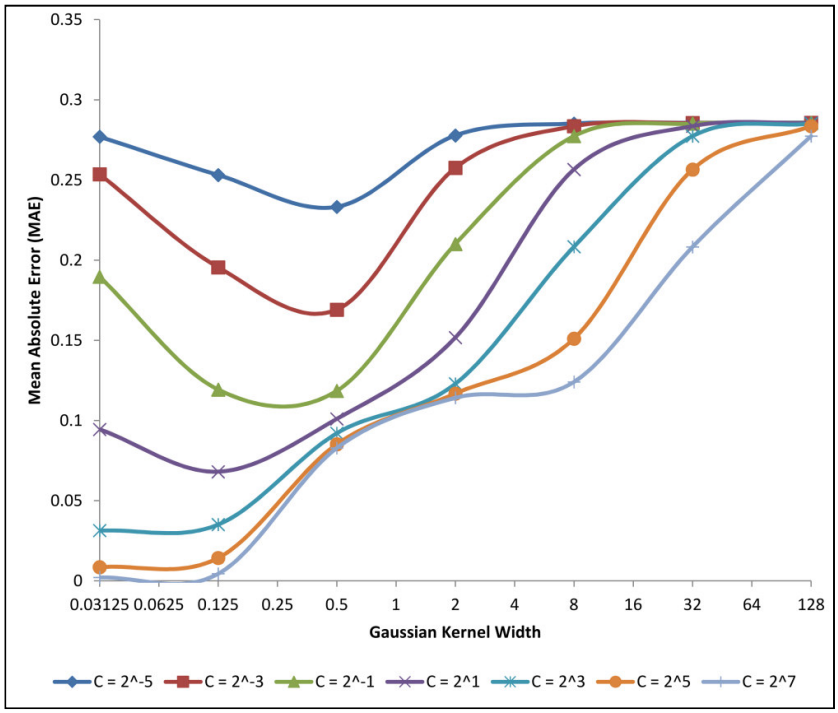


Figure 5.3(a): Coarse grid search result for the training data

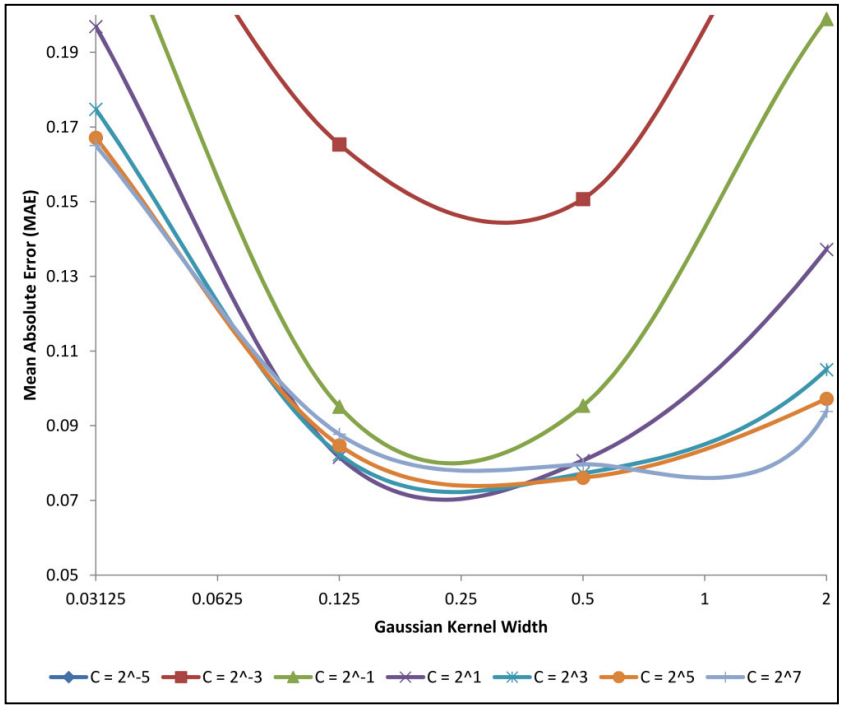


Figure 5.3(b): Coarse grid search result for the validation data

A narrower domain for the two values were obtained from the coarse grid search and then a fine grid search was performed in that domain to get more accurate values of the two parameters. From Figure 5.3, the narrow ranges for the two values were decided as:

$\sigma = [2^{-3} - 2^{-1}]$ and $C = [2^{1} - 2^{3}]$

The results of fine grid search are shown in Figure 5.4(a) and 5.4(b) for the training and validation data, respectively.

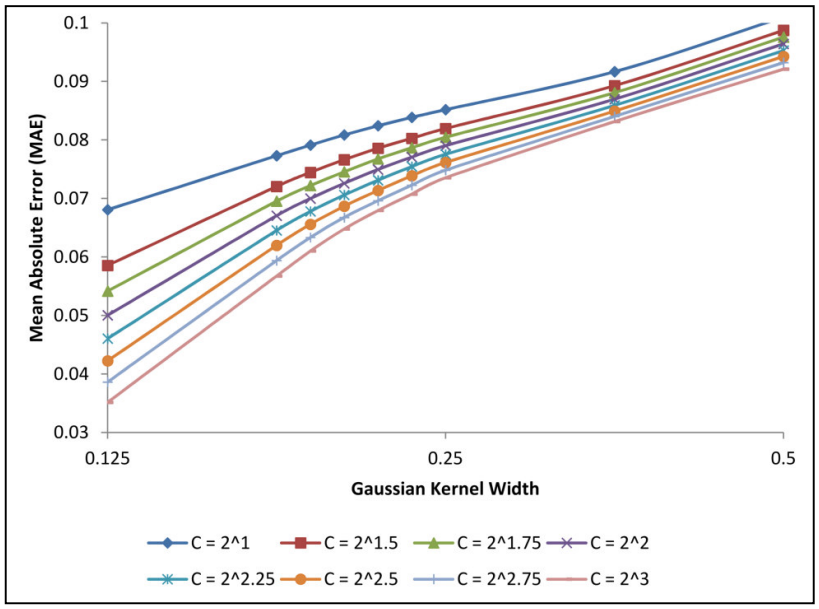


Figure 5.4(a): Fine grid search result for the training data

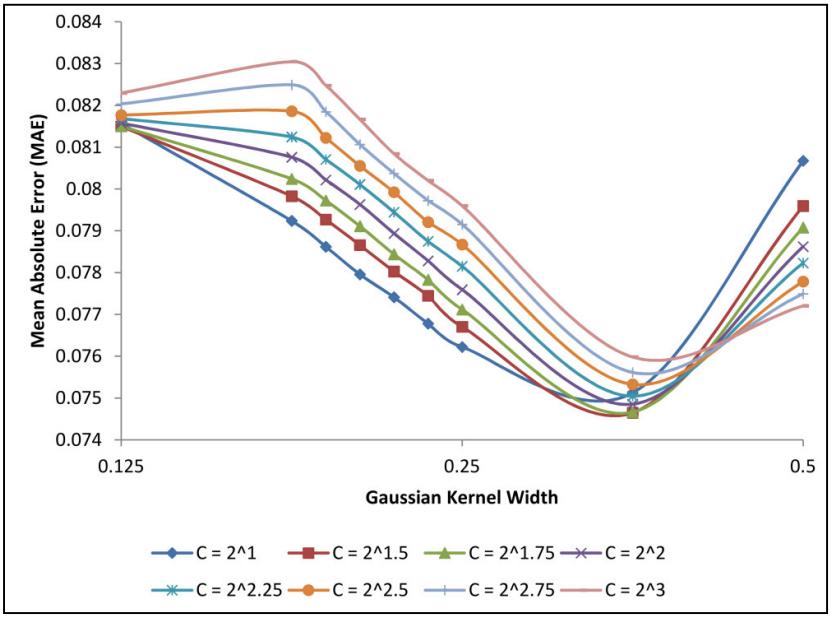


Figure 5.4(b): Fine grid search result for the validation data

It seems from Figure 5.4(b) that a kernel width value of 0.35 is the optimum. However, error for the training data is quite high at this value. The optimum parameters

were chosen as the ones producing acceptably lower value for both training and validation data. The optimum parameter values for the LS-SVR model are

$[C, \sigma] = [6.72, 0.2332]$ (5.4 b)

An LS-SVR model was also developed using the proposed analytical expressions (Cherkassky & Ma, 2004). As per these guidelines the value for regularization parameter is given by the following analytical expression (Equation 3.33):

$$C = \max\left(\left|\bar{y} + 3\sigma_y\right|, \left|\bar{y} - 3\sigma_y\right|\right) \quad (3.33)$$

While there is no proposed analytical expression for kernel width, the recommended range is given as $\sigma^d \approx (0.1 - 0.5) \Rightarrow \sigma \approx 0.4642 - 0.7937$ (d = number of input variables). Applying the above formula to the present industrial grinding process, the regularization parameter value was 1.4608. Figure 5.5 shows the LS-SVR model performance for the fixed regularization parameter (C = 1.4608) and different kernel width values in the range as suggested by Cherkassky & Ma (2004).

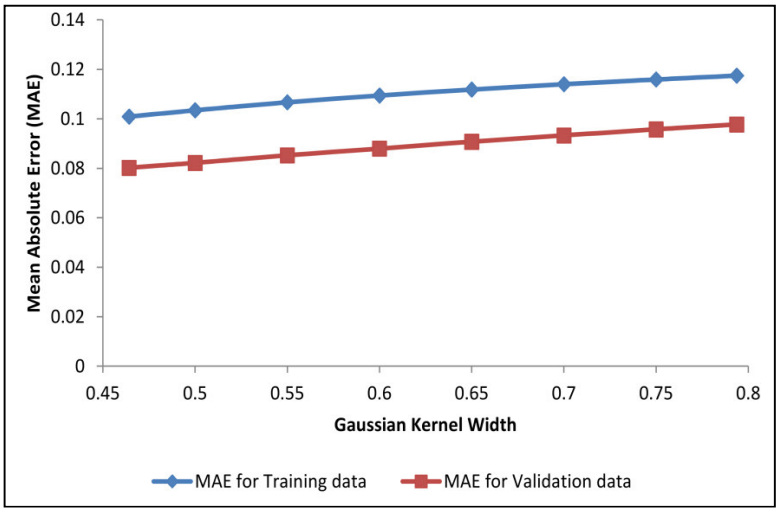


Figure 5.5: LS-SVR model performance using analytical tuning method

It can be observed that there is a monotonic increase of the error indicating that the optimum kernel width parameter lies outside the suggested range. The least error occurs at the lowest value of the kernel width (σ= 0.4642) which however, is not optimum.

For development of PCR and PCA-BPNN models, initially PCA was performed on the normalized values of entire 158 number of samples. The covariance matrix obtained for the 158×3 input data matrix is given below.

$$Covariance\ Matrix = \begin{pmatrix} 1 & -0.23973 & 0.361078 \\ -0.23973 & 1 & -0.45896 \\ 0.361078 & -0.45896 & 1 \end{pmatrix} \quad (5.5)$$

The eigen vectors obtained for the above covariance matrix are:

$$Eigen\ Vector\ Matrix = \begin{pmatrix} -0.279 & 0.81 & 0.514 \\ 0.584 & 0.568 & -0.58 \\ 0.762 & -0.139 & 0.632 \end{pmatrix} \quad (5.6)$$

The principal component analysis results based on the above eigen vectors are presented in Table 5.6:

Table 5.6: Results of PCA for the clinker grinding process

COMPONENTS	EIGEN VALUE	VARIANCE (%)	CUMULATIVE VARIANCE (%)
Principal component 1 (PC_1) $p_1 = 0.514x_1 - 0.58x_2 + 0.632x_3$	1.714	57.24	57.24
Principal component 2 (PC_2) $p_2 = 0.81x_1 + 0.568x_2 - 0.58x_3$	0.77	25.72	82.96
Principal component 3 (PC_3) $p_3 = -0.279x_1 + 0.584x_2 + 0.762x_3$	0.51	17.04	100

It can be observed in Table 5.6, that two principal components account for more than 80% of the total variance of the input dataset. Therefore, the input dimension was reduced from three actual input variables to two principal components p_1 and p_2 (latent variables).

For BPNN model development three types of training algorithms were used as mentioned in Chapter 4. Though gradient descent method has been the most used training algorithm for a back propagation neural network, it suffers from the problem of getting stuck in local minima. In order to avoid this problem, gradient descent with momentum (GDM) algorithm was used.

However, in cases where the value of the gradient (partial derivative of the error with respect to the weights) is very small, there is little change in weights according to the GDM algorithm which was the motivation for use of RP algorithm. The LM method has much better convergence characteristics than the other algorithms but has also more computational requirements. The detailed results of application of the aforementioned three algorithms for training of BPNN are shown in Table 5.7 and Figure 5.6.

LM algorithm is one of the fastest back propagation algorithms and it can be observed that this algorithm leads to faster convergence and the desired error goal for training data can be achieved with very less number of epochs. It may also be noted that for any algorithm the training error decreases with increase in the number of hidden layer neurons. A good model is judged by its ability to accurately estimate the output for both known and unknown data. Therefore, in Table 5.7 and Figure 5.6, the results are presented for different hidden layer neurons till the validation error showed a decreasing trend. The optimum number of neurons are attained once the validation error starts increasing (Figure 5.6).

For choosing the optimum number of hidden layer neurons, the error values produced with training as well as validation data were taken into consideration. It may be noted that the best result is produced by the model trained by resilient back propagation algorithm which produces the least validation error of 6.4%. Gradient descent with momentum algorithm although produces a comparable validation error, the error to training data is somewhat higher (7.8%). The percentage values mentioned here, are also known as mean absolute percentage error which is 100 times MAE.

Table 5.7: Back propagation neural network model performance for different training algorithms and different number of hidden layer neurons

TRAINING ALGORITHMS								
GRADIENT DESCENT WITH MOMENTUM (GDM)			LEVENBERG-MARQUARDT (LM)			RESILIENT BACK PROPAGATION (RP)		
No of Neurons in hidden layer	Mean Absolute Error (MAE)		No of Neurons in hidden layer	Mean Absolute Error (MAE)		No of Neurons in hidden layer	Mean Absolute Error (MAE)	
	Training	Validation		Training	Validation		Training	Validation
5	0.15111	0.11249	5	0.07284	0.06923	5	0.07786	0.06602
10	0.10045	0.08064	6	0.06516	0.07339	10	0.07956	0.07054
15	0.09041	0.07150	7	0.07117	0.07703	15	0.07059	0.06650
20	0.10137	0.09113	10	0.04028	0.09774	20	0.06804	0.06446
25	0.07819	0.06757	15	0.00483	0.12387	25	0.05953	0.08070
30	0.09134	0.07827						

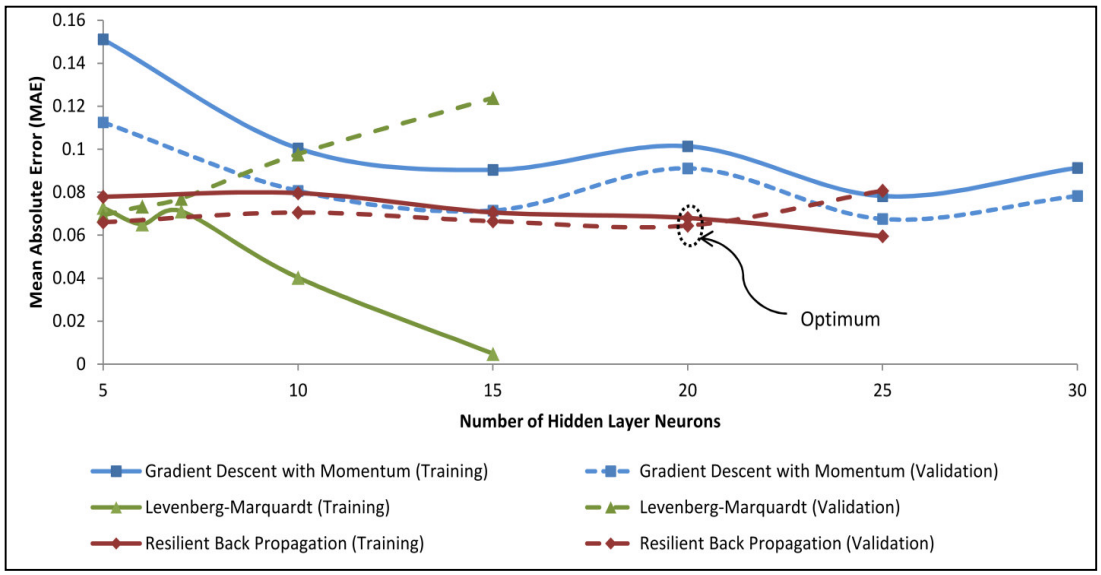


Figure 5.6: Mean absolute error as a function of number of neurons in back propagation neural network model

Validation error in case of LM algorithm increases quite fast with number of neurons which is a case of more accurate fitting to training data and less generalization with validation data. Also in case of LM algorithm the network converges quite fast with less number of neurons producing an error of 0.4% with the training data. However, the error for the validation data shows a sharp increase (more than 12%). One advantage of LM algorithm may be that though values are somewhat higher than RP algorithm at 7.2% and 6.9% respectively for the training and validation data, this is achieved at much less number of neurons than the other algorithms. From Table 5.7, the optimum back propagation network configuration can be decided as the one trained by RP algorithm with 20 number of hidden layer neurons and producing almost comparable error values (and also the least among all) for both known (training) and unknown (validation) data.

While RBF network model has the advantage of much less training time as compared to the back propagation network model, it suffers from the drawback of requiring more number of neurons than the BP network model for the same error goal. Ideally an RBF model should have as many hidden layer neurons as the number of training samples for

perfect fitting. But then it exhibits quite erratic behaviour to unknown data. Therefore, based on the criteria of producing low and almost comparable MAE values for both training and validation data, the network was designed by application of orthogonal least squares algorithm for different number of hidden layer neurons. The performance of RBF networks are produced for different spread value and different number of hidden layer neurons in Figure 5.7(a) and 5.7(b) respectively. The best RBF model was obtained as the one having 25 hidden layer neurons employing Gaussian basis function with a scaling parameter of 0.8.

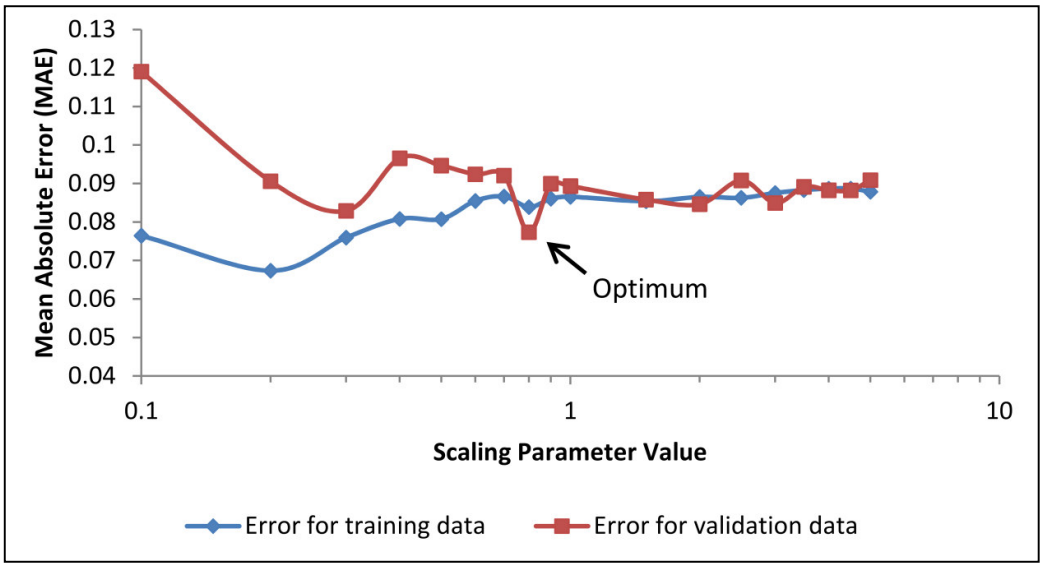


Figure 5.7(a): Mean absolute error as a function of scaling parameter in RBF neural network model (Constant number of hidden layer neurons)

The regression neural network has also as many hidden layer neurons as the number of training samples. The important design decision is the optimum selection of spread value. Figure 5.8 shows the MAE produced by the regression neural network (GRNN) to training and validation data at different values of spread parameter.

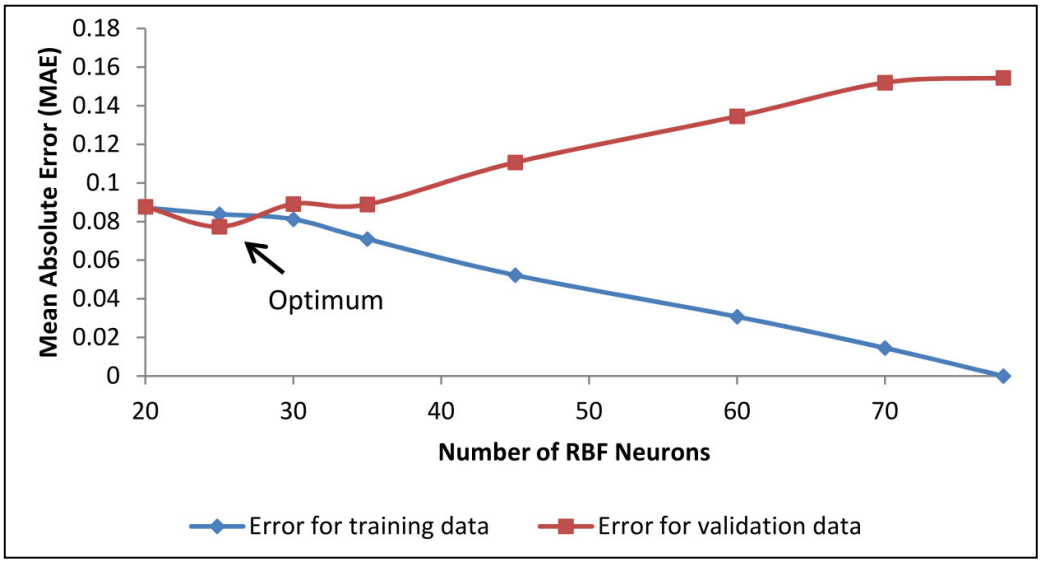


Figure 5.7(b): Mean absolute error as a function of number of RBF layer neurons in RBF neural network model (Constant scaling parameter)

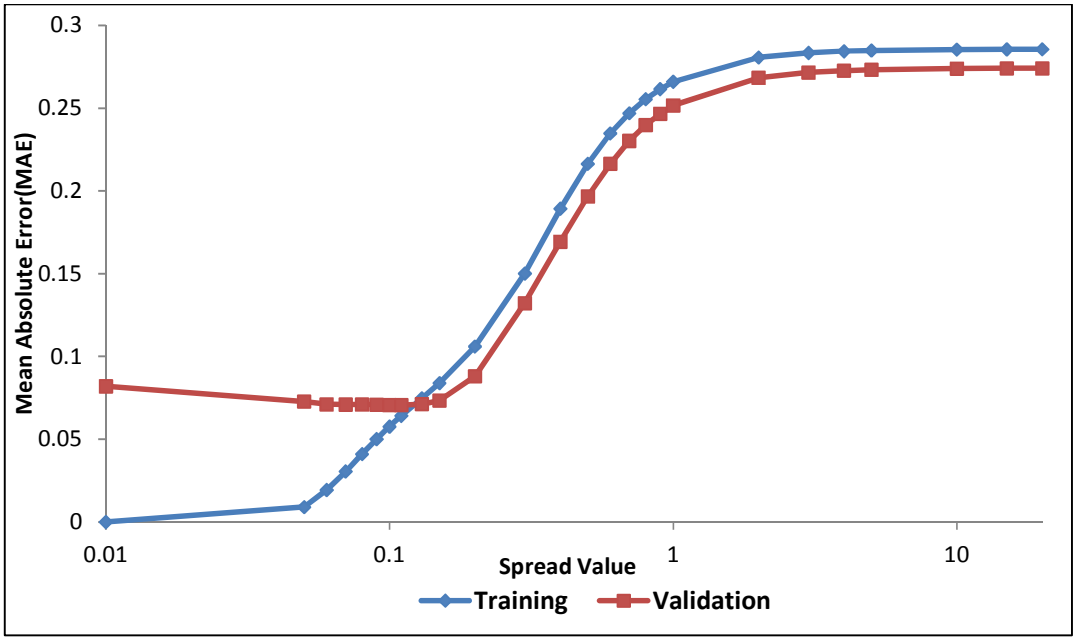


Figure 5.8: Mean absolute error value of regression network as a function of spread parameter in GRNN model

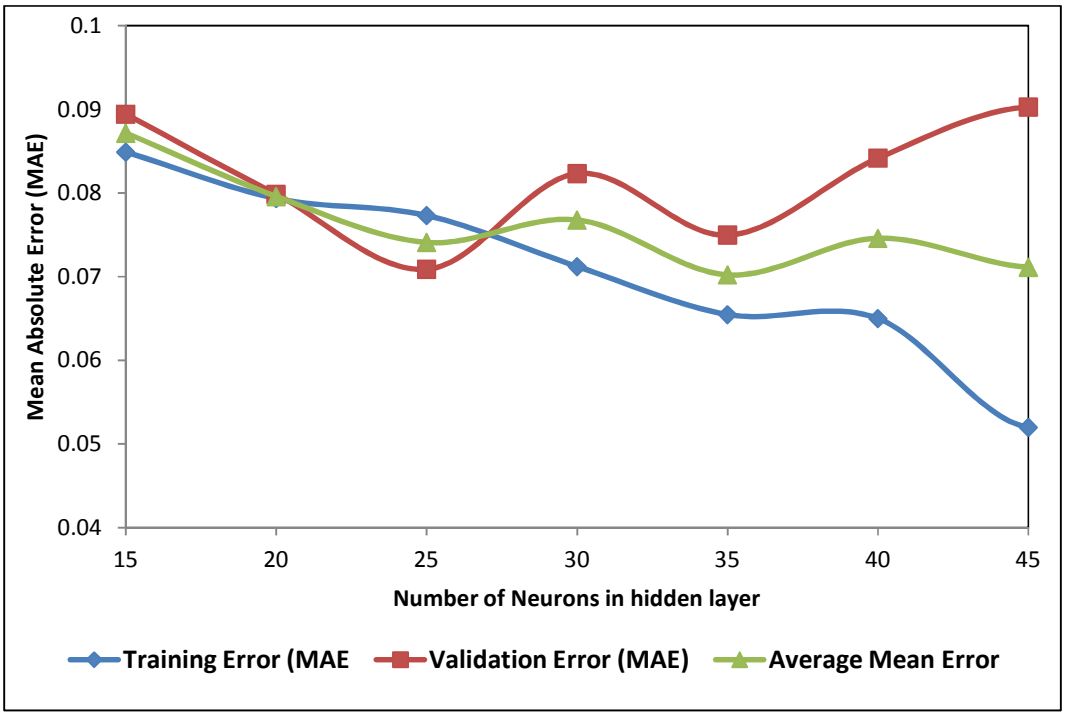


Figure 5.9: Training and Validation errors as a function of number of hidden layer neurons for the PCA-BPNN model

It can be observed from Figure 5.8 that as the value of smoothing parameter or spread value decreases the network results in more accurate fitting and as the spread parameter increases it results in less accurate fitting and better generalization. Since a good model is considered as one which estimates the output value quite accurately for both known and unknown inputs, the optimum spread value for the generalized regression network model was decided to be 0.11.

Gradient descent with momentum back propagation algorithm was used for training of the PCA-BPNN model where the model inputs are the two chosen latent variables p_1 and $p_2$instead of the three actual variables. The training and validation errors as a function of the number of hidden layer neurons are shown in Figure 5.9.

It can be observed in Figure 5.9 that the error for known data decreases with increase in the number of neurons. In fact, very low values for training error can be obtained if even higher number of neurons are used. But then, the performance of a model is judged

by its ability to predict outputs quite accurately for the unknown data. Use of very high number of neurons results in problem of over-fitting and poor generalization. Therefore the optimum number of neurons can be said to be the one producing almost comparable and of course sufficiently low error values both for known and unknown data. From Figure 5.9, it can be observed that the lowest average error is obtained with both 35 as well as 45 number of neurons. But the network with 45 neurons has the problem of overfitting and produces higher validation error. Therefore, the optimum number of the PCA based BPNN model was decided to be 35.

The performances of all the statistical models developed for the clinker grinding process are presented in Tables 5.8(a) and 5.8(b) for the training and validation data respectively.

Table 5.8 (a) Statistical Model Performance for training data in the grinding process

MODEL TYPE	MAE	RMSE	CORRELATION COEFFICIENT	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value	**0**	**0**	**1**	**100**	**1**	**0**
Linear Regression	0.1174	0.1571	0.8520	72.6015	0.7260	0.1266
Principal Component Regression	0.1285	0.1706	0.823	67.72	0.677	0.138
Quadratic Response Surface	0.1129	0.1511	0.8639	74.6474	0.7464	0.1266
Support Vector Regression (SVR_1: NSV = 56.9%)	0.0812	0.1201	0.9181	84.197	0.8399	0.0967
Support Vector Regression (SVR_2: NSV = 97.4%)	0.0627	0.1251	0.9117	83.002	0.8263	0.1011
LS-SVR_{GRID}	0.072	0.104	0.939	87.97	0.879	0.083
LS-SVR_{CM}	0.101	0.143	0.881	77.33	0.773	0.115

Table 5.8 (b) Statistical Model Performance for validation data in the grinding process

MODEL TYPE	MAE	RMSE	CORRELATION COEFFICIENT	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value	**0**	**0**	**1**	**100**	**1**	**0**
Linear Regression	0.0971	0.1273	0.8783	77.0649	0.7685	0.0892
Principal Component Regression	0.1124	0.1405	0.85	71.93	0.718	0.10
Quadratic Response Surface	0.0933	0.1282	0.8812	77.5266	0.7651	0.0891
Support Vector Regression (SVR_1: NSV = 56.9%)	0.0823	0.1131	0.9177	84.1133	0.8174	0.0773
Support Vector Regression (SVR_2: NSV = 97.4%)	0.0666	0.1003	0.9290	85.9939	0.8562	0.0695
LS-SVR_{GRID}	0.079	0.113	0.913	83.34	0.817	0.078
LS-SVR_{CM}	0.08	0.118	0.905	81	0.80	0.082

The performances of all the neural network models developed for the clinker grinding process are presented in Tables 5.9(a) and 5.9(b) for the training and validation data respectively.

The performances of all the fuzzy inference models developed for the clinker grinding process are presented in Tables 5.10(a) and 5.10(b) for the training and validation data respectively.

Table 5.9(a): Neural network model performance for training data in the grinding process

MODEL TYPE	MAE	RMSE	CORRELATION COEFFICIENT	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value	**0**	**0**	**1**	**100**	**1**	**0**
BPNN	0.068	0.135	0.896	79.75	0.797	0.108
RBFNN	0.084	0.119	0.918	84.26	0.843	0.095
GRNN	**0.064**	**0.097**	**0.947**	**89.46**	**0.894**	**0.078**
PCA-BPNN	0.0654	0.1342	0.8979	80.146	0.8	0.1077

Table 5.9(b): Neural network model performance for validation data in the grinding process

MODEL TYPE	MAE	RMSE	CORRELATION COEFFICIENT	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value	**0**	**0**	**1**	**100**	**1**	**0**
BPNN	**0.064**	**0.103**	**0.925**	**84.88**	0.726	**0.072**
RBFNN	0.077	0.114	0.906	82.12	0.812	0.079
GRNN	0.071	0.109	0.918	83.96	0.829	0.076
PCA-BPNN	0.0749	0.1086	0.915	83.171	0.8316	0.0759

Table 5.10(a): Fuzzy inference model performance for training data in the grinding process

MODEL TYPE	MAE	RMSE	CORRELATION COEFFICIENT	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value	**0**	**0**	**1**	**100**	**1**	**0**
Fuzzy Inference (Mamdani)	0.097	0.1314	0.9134	81.8077	0.8083	0.1089
Fuzzy Inference (Sugeno)	0.0766	0.1262	0.9118	83.0905	0.8230	0.1028
Neuro-fuzzy (ANFIS)	**0.0551**	**0.0945**	**0.9490**	**90.0769**	**0.9007**	**0.0754**

Table 5.10(b): Fuzzy inference model performance for validation data in the grinding process

MODEL TYPE	MAE	RMSE	CORRELATION COEFFICIENT	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value	**0**	**0**	**1**	**100**	**1**	**0**
Fuzzy Inference (Mamdani)	0.0939	0.1198	0.9065	80.8412	0.7951	0.0867
Fuzzy Inference (Sugeno)	0.0737	0.1053	0.9176	84.2005	0.8415	0.0739
Neuro-fuzzy (ANFIS)	**0.0626**	**0.1045**	**0.9208**	**84.6499**	**0.8438**	**0.0727**

Except linear and principal component regression models, all other models were developed so as to minimise the mean absolute error between the actual and model predicted output values.

However a critical question that remains unanswered in most of research works is: Is the lowest error value obtained for a particular model low enough so that the model is acceptable for practical use? Literature search revealed that there is rarely a straight forward answer to this question and it can be said that the criteria is process specific. While for some processes, errors in excess of 10 to 20% have been stated as acceptable, some have also mentioned modeling error not to exceed 10%.

Moreover, it has been observed that in most of the modeling works related to industrial processes, the models are assessed based on one or two performance criteria. Different researchers have used different model evaluation criteria. However, a model showing better value on one performance criterion may produce lower accuracy in another performance criteria. Therefore, in this work, the model performances are assessed by evaluating six different statistical parameters as described below.

The mean absolute error: $$MAE = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N} \tag{5.3}$$

The root mean squared error (RMSE): $$\sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}} \tag{5.7}$$

Correlation coefficient (R): $$\left(\frac{\sum_{i=1}^{N} (y_i - \bar{y}_i)(\hat{y}_i - \bar{\hat{y}}_i)}{\sqrt{\sum_{i=1}^{N} (y_i - \bar{y}_i)^2 \sum_{i=1}^{N} (\hat{y}_i - \bar{\hat{y}}_i)^2}} \right) \tag{5.8}$$

The variance account for (VAF): $$\left(1 - \frac{\mathrm{var}(y - \hat{y})}{\mathrm{var}(y)}\right) \times 100 \tag{5.9}$$

Nash-Sutcliffe efficiency (NSE): $$1 - \left[\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2} \right] \tag{5.10}$$

Theil's inequality coefficient (TIC): $$\frac{\sqrt{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}}{\sqrt{\sum_{i=1}^{N} y_i^{\,2}} + \sqrt{\sum_{i=1}^{N} \hat{y}_i^{\,2}}} \quad (5.11)$$

A good model should have MAE and RMSE values close to 0, VAF value close to 100 and R value close to unity. The Nash-Sutcliffe efficiency (Nash and Sutcliffe, 1970) which was initially proposed for evaluation of river flow forecasting models has since been applied for evaluation of various chemical process models such as membrane bio reactors (Mannina and Bella, 2012) and water quality prediction (Singh and Gupta, 2012). Similarly, Theil's inequality coefficient (TIC) has been successfully applied as a model evaluation parameters for TiO_2 production process (Hvala *et al*., 2005) and polymerization process (Li and Liu, 2011). NSE value ranges from -∞ to 1, 1 being the value for a perfect model. TIC value lies between 0 and 1 with 0 being the value for the case of perfect model.

An important question to be answered is what is the cut off prediction error criterion based on which it can be concluded that the best model developed is actually acceptable. In fact it depends on the application area of the model. MAE values of 0.1036 and 0.1757 for training and validation data, respectively, have been reported in soft sensing of ball mill output particle size (Wu and Yuan, 2009). Similarly, VAF values of 81 % and 83% have been reported as satisfactory (Gokceoglu and Zorlu, 2004; Yagiz and Gokceoglu, 2010). In modeling of grinding units estimation with around 6% error has been mentioned as acceptable (Casali *et al*., 1998) for soft sensing of particle size. Similarly, it has been reported in the literature that good models should have NSE values higher than 0 and TIC value less than 0.3 (Moriasi *et al*., 2007). In Tables 5.8-5.10, under each model evaluation criterion, the closest to the ideal value has been highlighted.

It can be observed that out of the two standard SVR models, SVR_2 model produces better performance. However, in case of SVR models, the model complexity increases with number of support vectors and it has been suggested that for optimum modeling performance, the hyper-parameters be so chosen that the number of support vectors is around 50% of the training samples. Therefore, though SVR_2 yields better results, it has the number of support vectors 97.4% of training samples. The optimum SVR model was searched in the range of 40 to 60% number of support vectors and it can be observed in Figure 5.2 that in this range, SVR_1 with the stated parameters is the optimum SVR model. The purpose of providing the performances of two types of LS-SVR models in Table 5.8 is not to conclude that analytical tuning formulae for computation of SVR hyper-parameters as proposed by Cherkassky and Ma (2004), should not be used. In fact, in the standard SVR model design, combination of grid search and analytical formulae were used for optimum SVR hyper-parameter selection. The comparison of the two LS-SVR models is only to suggest that in case of LS-SVR model, where only two unknown parameters are required to be determined, a rigorous grid search may result in better optimum values. The advantage of LS-SVR model using grid search is also quite obvious while comparing with the results of standard SVR in Table 5.8. The standard SVR model using a combination of analytical method and grid search tuning technique produced a minimum validation error of 8.2% compared to LS-SVR model where the error is less (7.9%).

It may be noted that the regression neural network produces the best result for the training data and the back propagation network yields the best result for the validation data. Since a model is evaluated on the basis of its generalization ability, the BPNN model trained with resilient back propagation algorithm can be considered better than the GRNN model. Moreover, the BPNN model uses 20 hidden layer neurons where as the GRNN has 79 patter layer neurons (equal to the number of training samples).

It can be observed in Tables 5.8-5.10 that though both RMSE and MAE represent the average error and both RMSE and MAE have the units of difference between actual and predicted value, for any model, the RMSE value is higher than MAE value. This is due to the fact that in RMSE, large errors get amplified because of squaring phenomenon. In case of training errors for BPNN and RBFNN models, BPNN produces less MAE despite higher RMSE. Similarly the ambiguity in model performance assessment based on one or two performance criteria is illustrated by observing the MAE and RMSE value of linear and quadratic models in Table 5.8(b). While quadratic model results in lower MAE, linear model shows marginally lower RMSE value. This observed variation in model performance is due to the fact that RMSE value is not only a function of average error but also a function of the distribution of error magnitudes. Therefore, MAE has been recommended as a better performance index than RMSE (Willmott and Matsuura, 2005).

Considering the prediction errors produced by different models, it can be concluded that the BPNN model can be accepted as a reasonably good model for the grinding process. Because, the BPNN model trained by resilient back propagation algorithm shows modelling error in the range of 6 to 7% for both training and validation data. Moreover, the BPNN model is also more accurate than the Mamdani and Sugeno type fuzzy inference models where the reported error values are 9.4% and 7.3%, respectively.

It may further be noted that the BPNN model has also better performance than the hybrid modeling approach of principal component analysis followed by BPNN modeling where the modeling error was 7.5%. Similar phenomenon is also observed by comparing the performance of linear and principal component regression models where linear regression model has better performance than PCR model. This shows that in the grinding process involving three input variables, dimensionality reduction by use of latent variables does not result in better modeling accuracy.

Among fuzzy inference models, the performance of the neuro-fuzzy model is clearly superior to all other types of models as observed in Table 5.10. While 5 types of linguistic variables were used for all the three input variables for the Mamdani and Sugeno models (Table 4.3), the neuro fuzzy approach achieves better result with less number of linguistic variables (3 numbers for each type of inputs). In addition to that, a total of 27 rules were used for the ANFIS model while 63 and 45 fuzzy *if-then* rules were framed for the Mamdani and Sugeno type models respectively.

On comparing the results of Tables 5.8 to 5.10, finally it can be concluded that both the BPNN model (validation error 6.4%) and the neuro-fuzzy inference model (validation error 6.26%) have the best performances as compared to all other model types. Both these models produce errors in the range of around 6% for the validation data (This percentage error which is also known as mean absolute percentage error is 100 times the mean absolute error). This error value has been recommended as acceptable by Casali *et al*., (1998) for soft sensing of particle size.

The predicted cement fineness values by various developed soft sensor models are presented in Figures 5.10 and 5.11 for the training and validation data respectively.

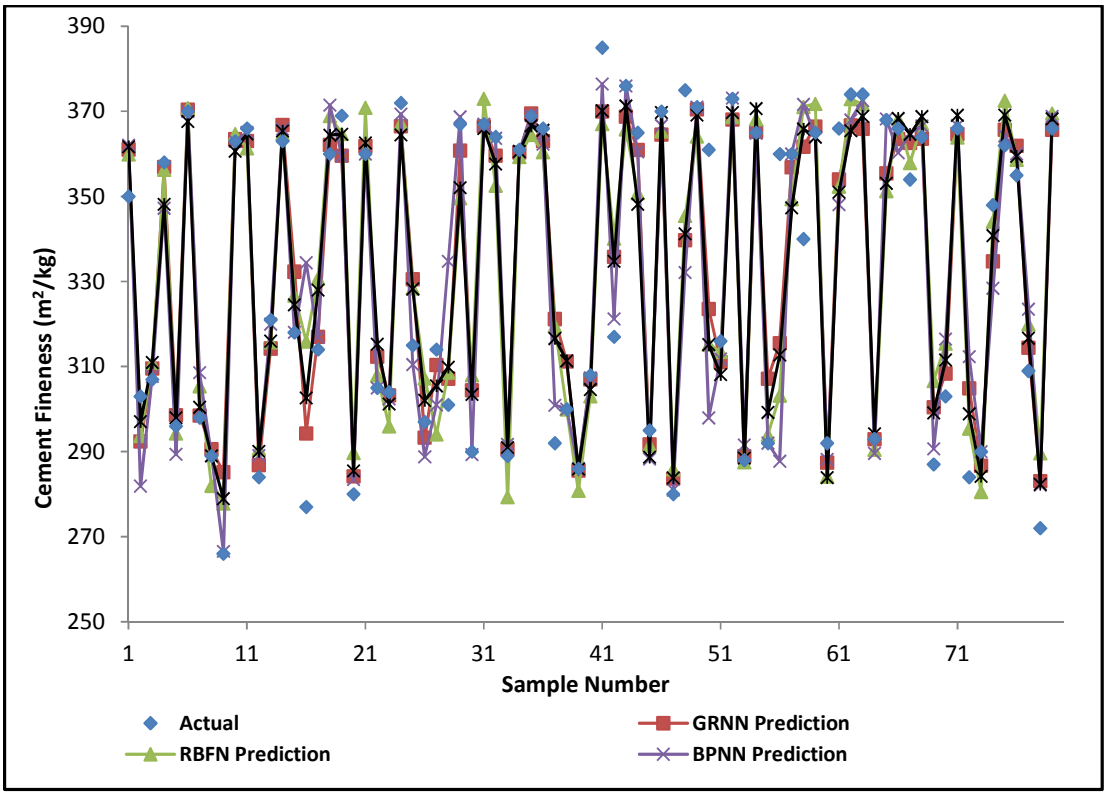


Figure 5.10 (a): Predicted values of cement fineness by different neural network models (Training data)

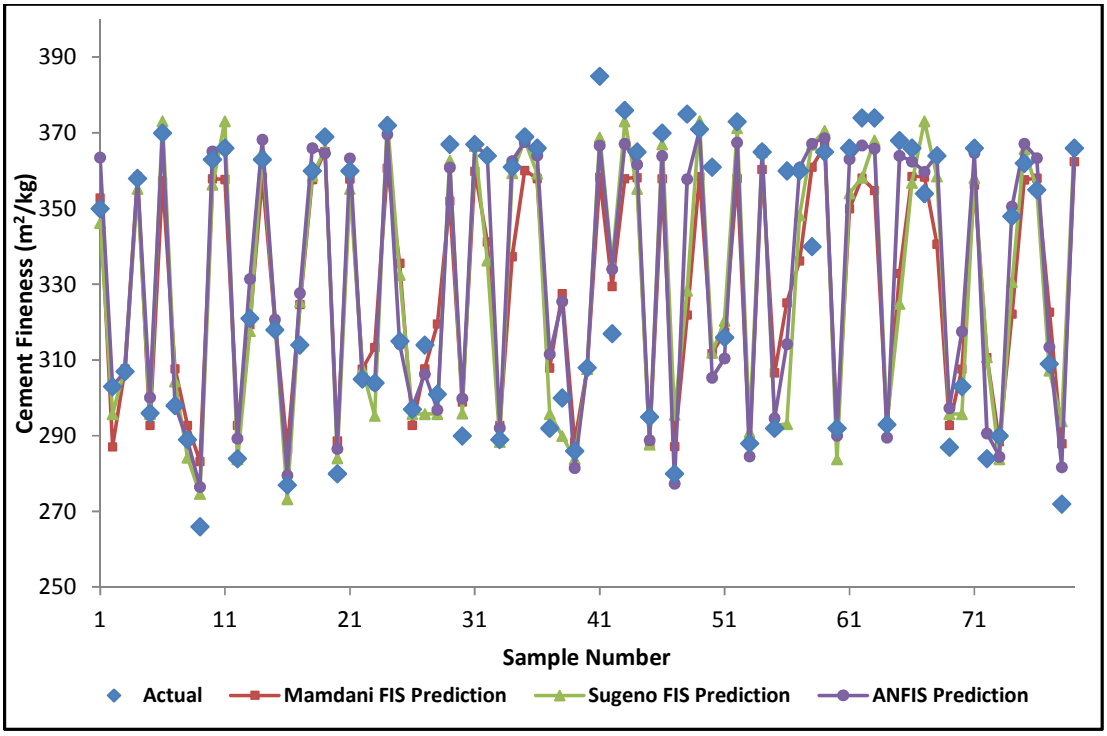
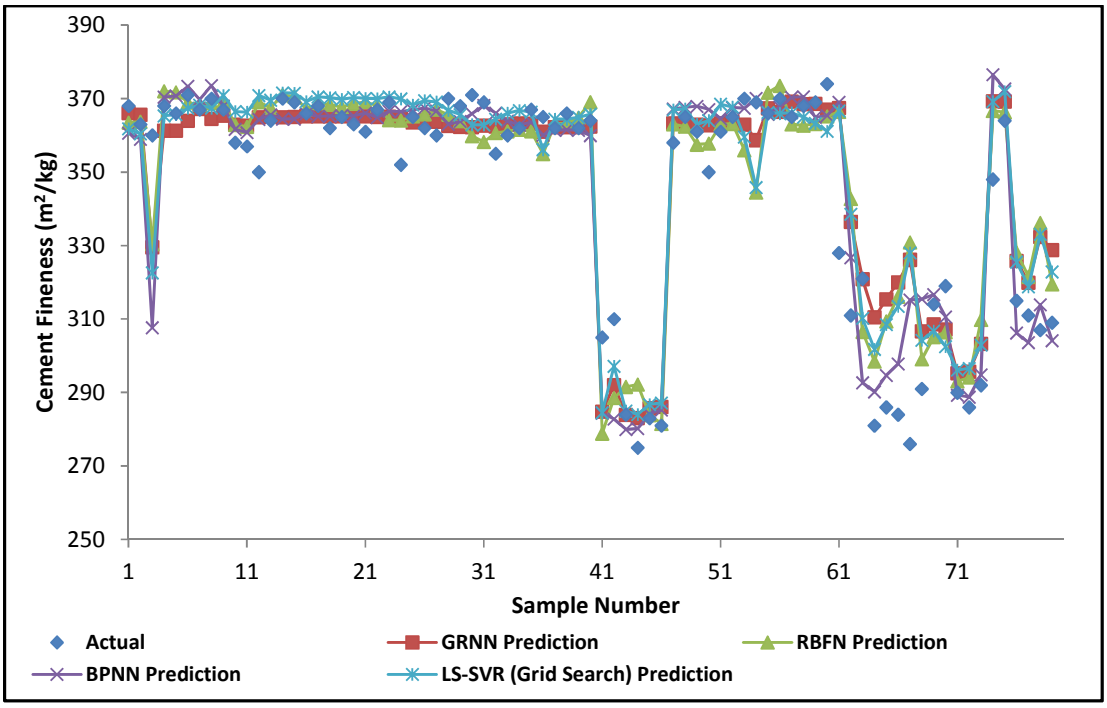


Figure 5.10(b): Predicted values of cement fineness by different fuzzy inference models (Training data)



5.11 (a): Predicted values of cement fineness by different neural network models (Validation data)

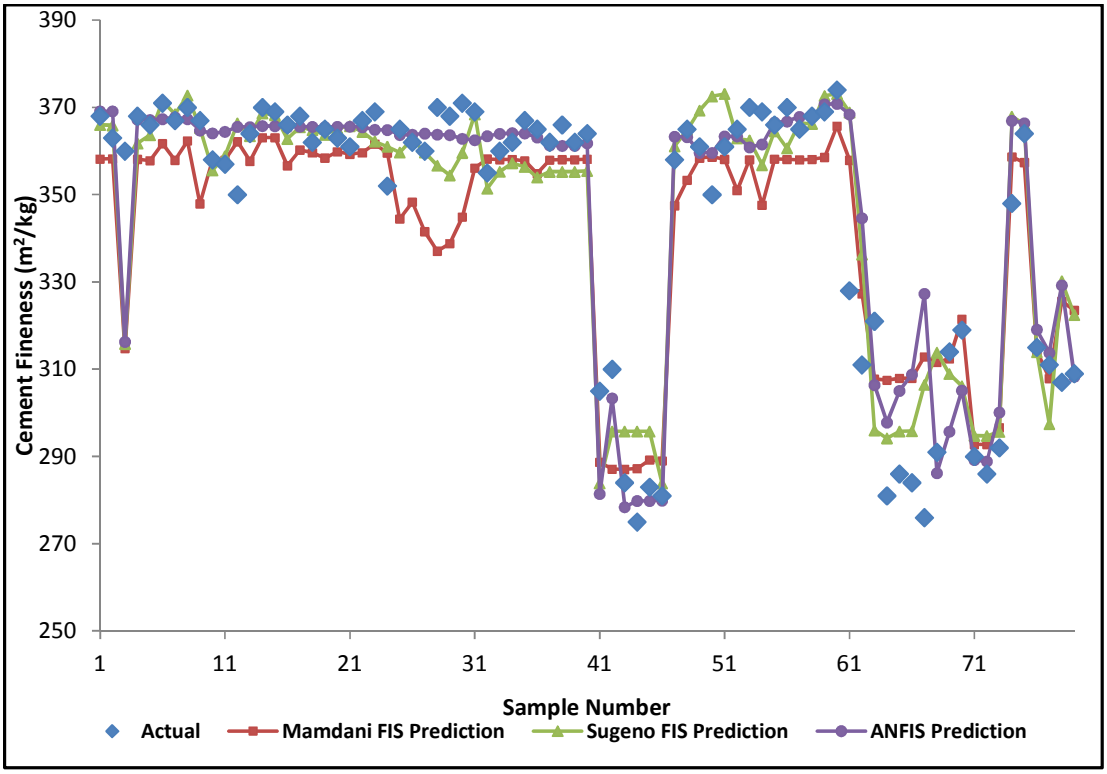


Figure 5.11 (b): Predicted values of cement fineness by different fuzzy inference models (Validation data)

Static Vs dynamic soft sensor

It is useful at this stage to discuss on the static or dynamic nature of the developed soft sensor. When input and output data at the same time instant are considered for model development, the soft sensor models are called as static. On the other hand, if the instantaneous output is modeled as a function of the inputs of the same as well as previous time instances, the model is said to be a dynamic one.

A static soft sensor has the form $y(t) = f(x_1(t), x_2(t), x_3(t))$ (5.12)

A dynamic soft sensor will have the form

$$y(t) = f(x_1(t), x_1(t-1),..., x_2(t), x_2(t-1),..., x_3(t), x_3(t-1),...) \quad (5.13)$$

In this work, since the instantaneous values of inputs and outputs were considered for modeling, the developed models can be considered static. However it will be improper to say that a dynamic model is always better than a static one. The reason is elaborated below.

Usually the dynamic soft sensors are developed by augmenting the instantaneous input values with time lagged values. The usefulness of a static or dynamic soft sensor largely depends on the process for which the inferential monitoring system is developed. Dynamic soft sensors are more useful than static soft sensors for process industries where dynamic process conditions are more common (e.g. regular grade changeover operations in polymer processes) or when significant time delay exists between inputs and outputs in which case input values of previous instances are also important . These two process conditions have no meaning in the clinker grinding process using vertical roller mill for the reasons explained below.

There are very few grade change over operations in the clinker grinding process. This essentially means the desired cement fineness value is more or less constant for the entire duration of grinding. Therefore, unsteady state in operation occurs only during start up and

shut down period and during normal operation the process operates mostly at steady state. Similarly, The effects of the inputs on the output are quite fast and the time delay factor is negligible.

The negligible time delay is a significant advantage of vertical roller mill over ball mill and also an important reason why VRM modeling is quite different than ball mill modeling. Particle residence time in the vertical roller mill is less than even a minute as compared to ball mills where particle residence time is in the order of 20 to 30 minutes (Sahasrabudhe *et al*., 2006). Therefore while the present output can be considered as a function of the instantaneous input, it is highly unreasonable to consider that the present output is a function of the input which is available a few minutes back.

Online Implementation

The designed soft sensor models can be implemented in the process for online estimation of cement fineness from input process data. It should be emphasized at this juncture that any data-driven model must not be used for predicting output from an input data which does not lie within the data range used for initial model development. This is because data-driven models do not extrapolate well. The online implementation of the model in the SIMULINK environment is shown in Figure 5.12 using the two best performing soft sensor models i.e. the BPNN model and the ANFIS model.

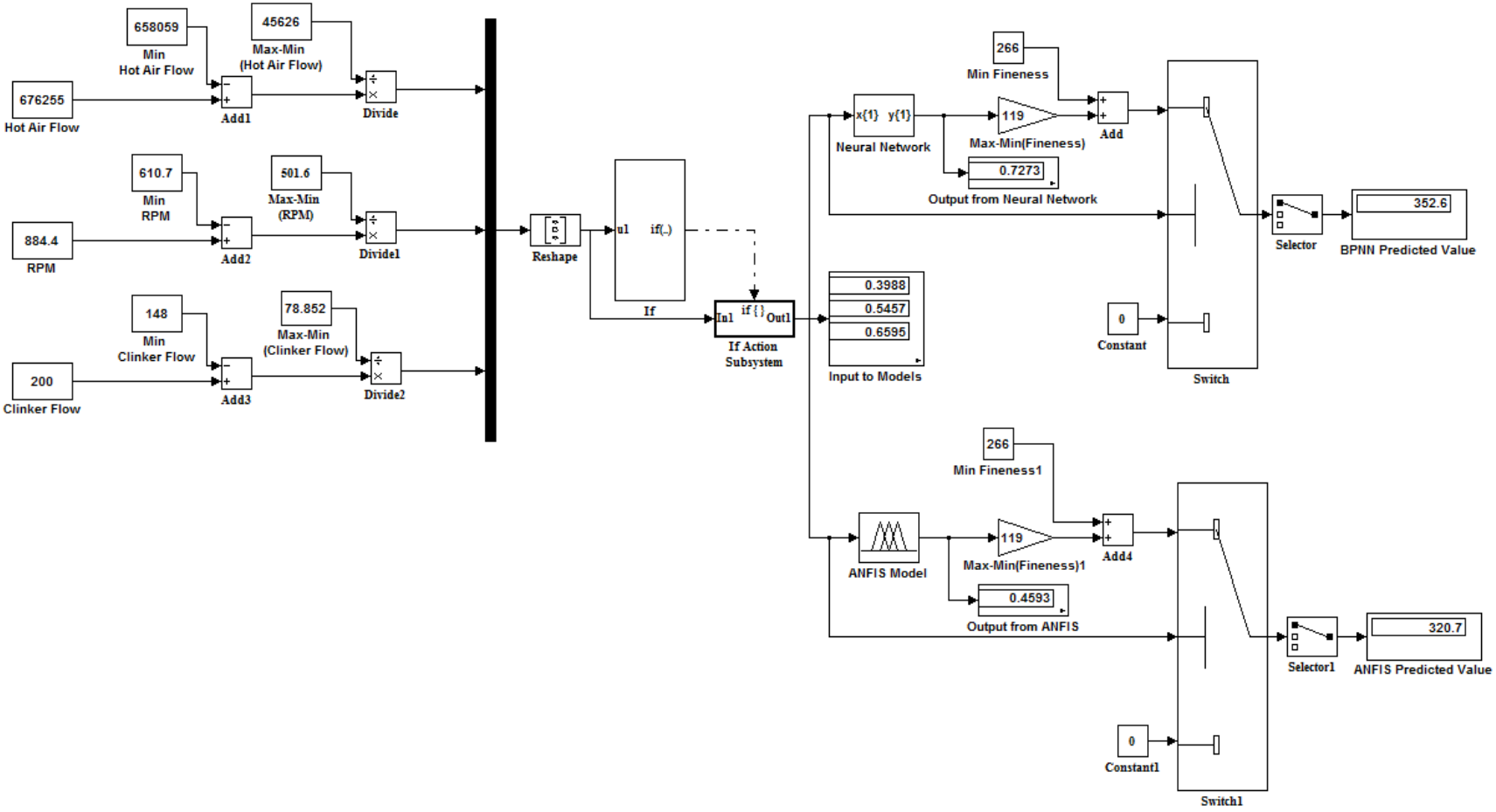


Figure 5.12(a): SIMULINK block diagram showing online monitoring of cement fineness with the neural network and fuzzy inference model for normal input data

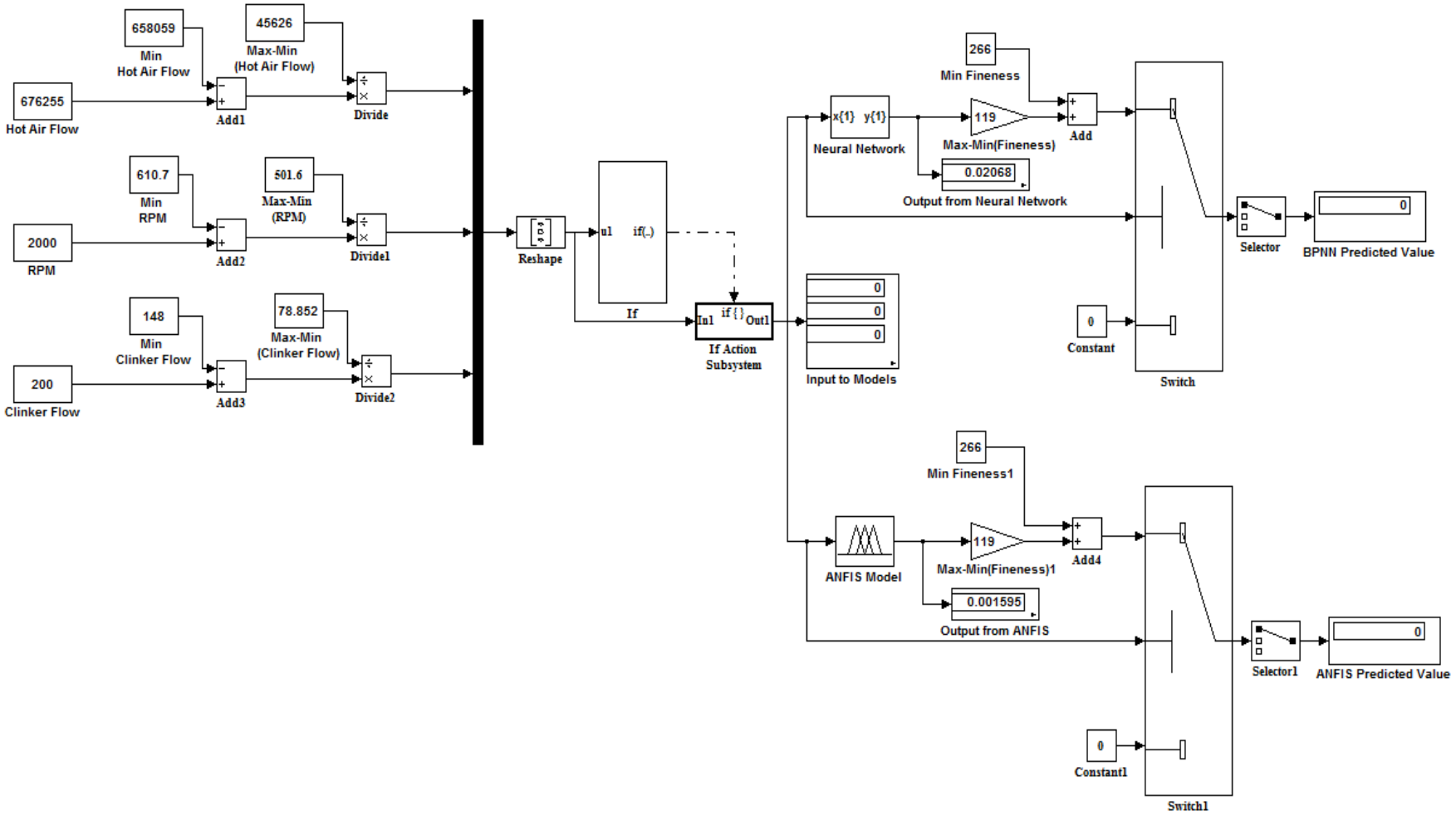


Figure 5.12 (b): SIMULINK block diagram showing online monitoring of cement fineness with the neural network and fuzzy inference model for abnormal (or outlier) input data

In Figure 5.12(a), the three blocks on the left receive the values of the input process variables as measured by the hardware sensors. At any instant, one or more measured input process variables may either be normal (values that fall within the range for model development)or abnormal (values that fall outside the range). The instantaneous values received are then normalized with their respective minimum-maximum values which subsequently pass onto the multiplexor. The row vector of input normalized data is transposed to a column vector by the 'Reshape' block to meet the syntax requirement of the models. However before passing on the input set to the model a range check has been introduced. The 'if' block checks if all the three normalized values are within the range 0 and 1. The condition supplied to the 'if' block is mentioned below:

$u1(1) \geq 0$ & $u1(1) \leq 1$ & $u1(2) \geq 0$ & $u1(2) \leq 1$ & $u1(3) \geq 0$ & $u1(3) \leq 1$

Here u1(1), u1(2) and u1(3) represent the three elements of the input u1 to the 'if' block.

Subsequently, if the above condition is satisfied the 'if action subsystem' block following the 'if' block passes the values to the two models. The output produced by the models are in the normalized form which are subsequently converted to the actual cement fineness or Blaine value. Figure 5.12(a) shows the simulation results for one set of input data which are normal values (i.e all normalized inputs lie within the range 0 and 1). Figure 5.12(b) explains what happens if one or more input values are abnormal or outlying values.

It can be seen in the block diagram that range for classifier RPM used for model development is from 610 to 1112. Figure 5.12(b) shows the simulation result for a classifier input value of 2000 which is outside the modeling range and hence in such a scenario, the output of the model cannot be relied upon. After normalization, the value for RPM is greater than one and the condition written in the 'if' block is not satisfied. Therefore the set of input values are prevented from flowing to the model which can be

noticed in the 'display' block preceding the model (all values are shown as 0). However, the models produce some output even for this set of 0 values which must be avoided in the final result display. This has been achieved by adding the 'switch' block which passes the model output to the 'display' if the inputs to the models are nonzero. Else a '0' is displayed as the final simulation result. The '0' is an indication that one or more input values lie outside the desired range to be used by the model.

Previously, in Figures 5.10 and 5.11, comparison of the performances of different models for training and validation datasets were presented only at the instances when laboratory values of cement fineness were available (total 158 values, 79 training values and 79 validation values). After showing the implementation strategy of the best chosen models in Figure 5.12, the continuously model predicted values of cement fineness for all the available input data (total 281 values as mentioned in Section 4.1.2) along with the intermittent reported laboratory values are presented in Figure 5.13.

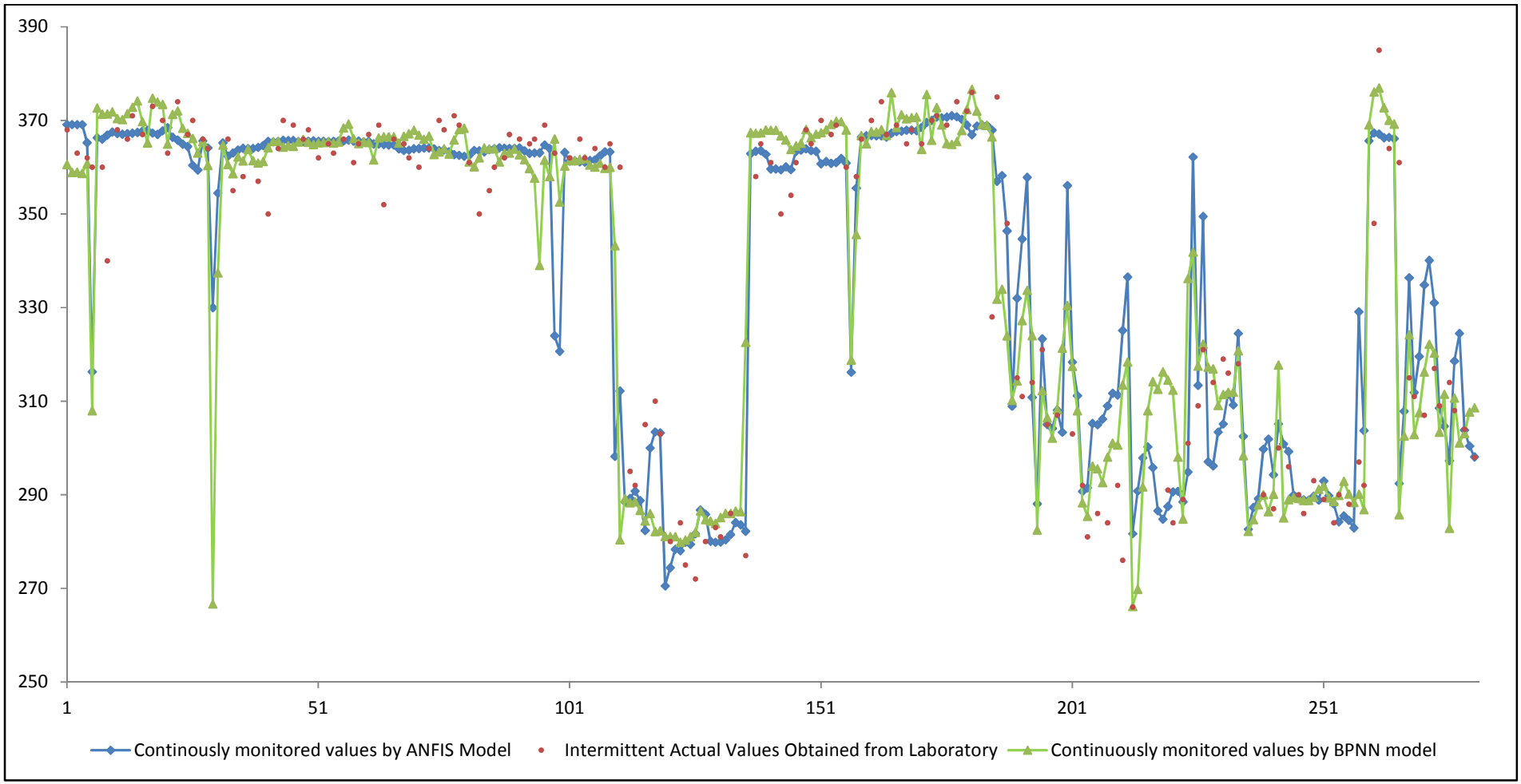


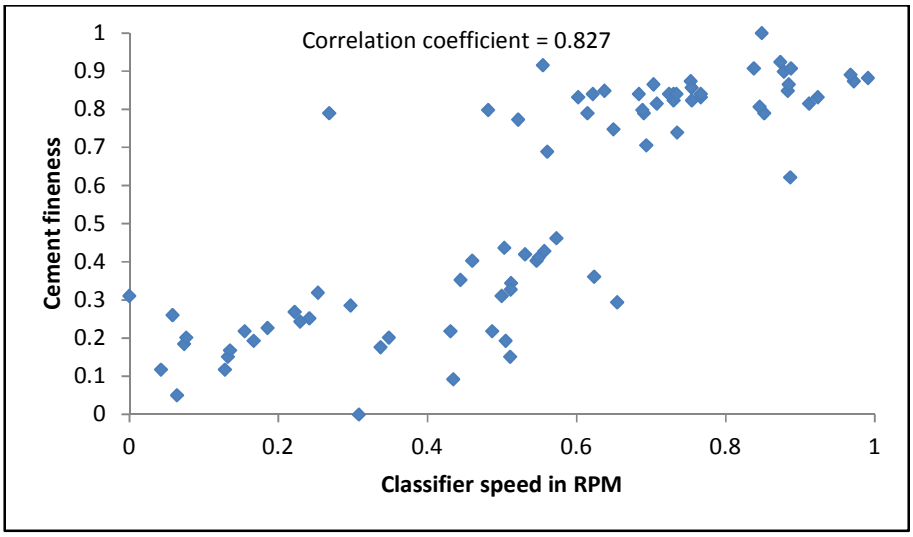
Figure 5.13: Continuously monitored values of cement fineness by back propagation neural network model and adaptive neuro-fuzzy inference model along with intermittent reported laboratory values

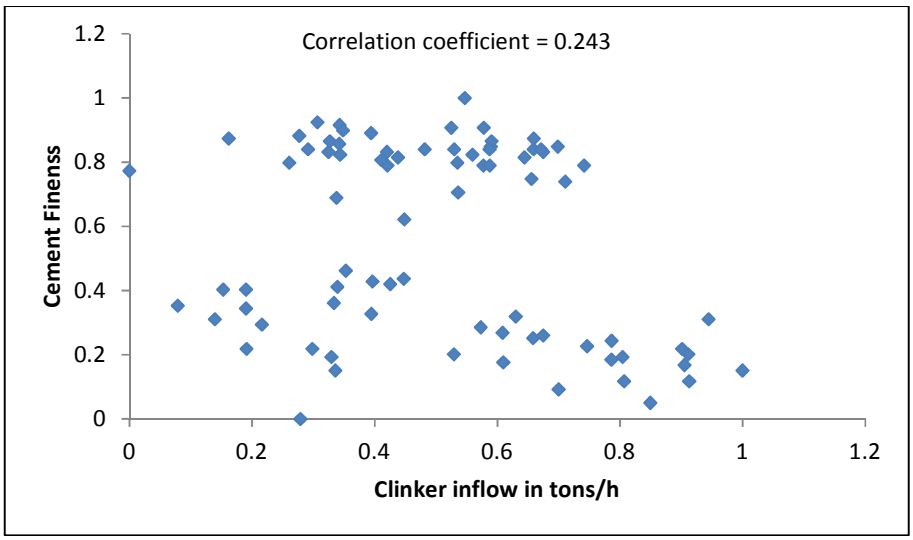
Closed loop study

Model predictive control strategy for controlling cement fineness (Blaine) in the clinker grinding process using ball mill has been proposed earlier (Martin and McGarel, 2001). However, due to of lack of online monitoring system, there is no control system in place for accurately controlling cement Blaine in cement industries. According to Bureau of Indian Standards (BIS), 53-grade ordinary portland cement (cement concrete achieving a compressive strength of 53 MPa in 28 days) must have a minimum fineness value of 225 m^2/kg. Many industries operate so as to maintain the fineness value in the range of 340 to 360 m^2/kg. However, there is no continuous control system in place to ensure that the fineness remains within a particular range. The developed model can be utilized for continuous monitoring and control of cement fineness.

An important issue is the selection of the manipulated variable. Since clinker inflow and hot air flow rate are the upstream values, these are mostly the disturbances for this process. Classifier RPM value is changed by the mill operator and can be considered as the manipulated variable. The linear regression model of the grinding process (Equation 4.1) indicates that among the three input variables, hot air flow has the least effect on the fineness. This phenomenon is further substantiated by making scatter plots of the output with each of the two disturbance inputs computing the correlation coefficient. The plots are shown in Figure 5.14.

Correlation coefficient has been used in the literature for influential variable selection. Higher value indicates strong correlation between the two variables and vice versa (Warne *et al*., 2004). Negative correlation indicates inverse relationship of the two variables.





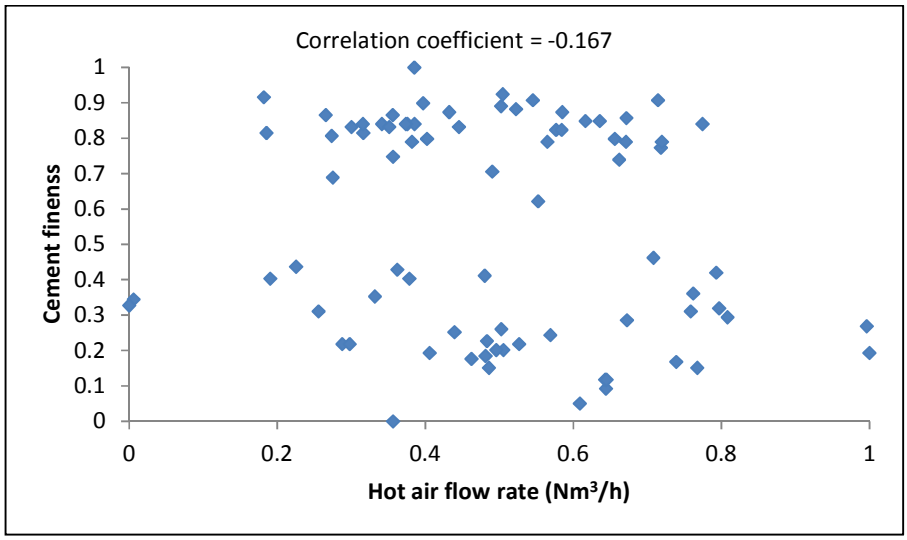


Figure 5.14: Scatter plots and correlation coefficient showing the dependence of the output with the three input variables

It can be observed that between the two disturbance variables, dependence of cement fineness on clinker inflow rate is stronger than dependence of cement fineness on hot air flow rate. Therefore, closed loop study was performed with clinker inflow rate as the disturbance variable. The model predictive control strategy to maintain cement fineness value within desired range is shown in Figure 5.15.

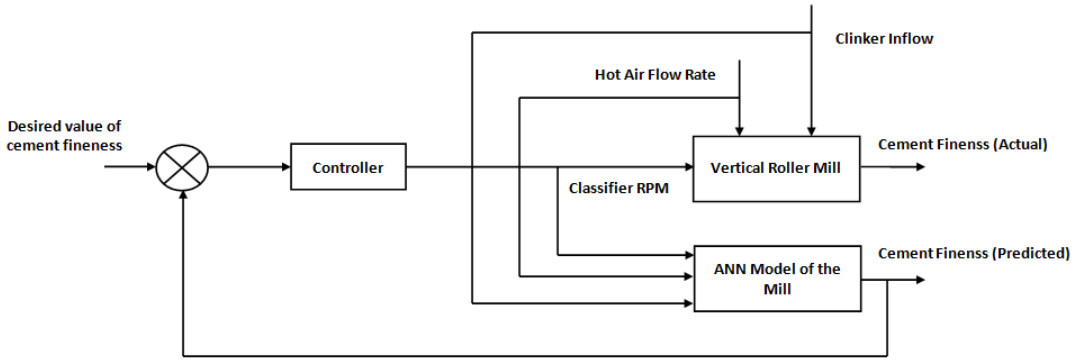


Figure 5.15: Closed loop structure for continuous monitoring and control of cement fineness

Here, the clinker grinding process consisting of the VRM and the associated inputs and outputs is shown. The values of the inputs are received from online physical sensors installed in the process. The block shown as ANN model in Figure 5.15 actually represents the entire structure shown in Figure 5.12. This ANN model performs sequentially the tasks of normalization of the actual inputs, comparison with the minimum and maximum values used for model design, computation of the output and conversion of normalized output to actual value in the unit of m^2/Kg. The continuously monitored value produced by the model and the set point value are supplied to the model predictive controller (MPC).

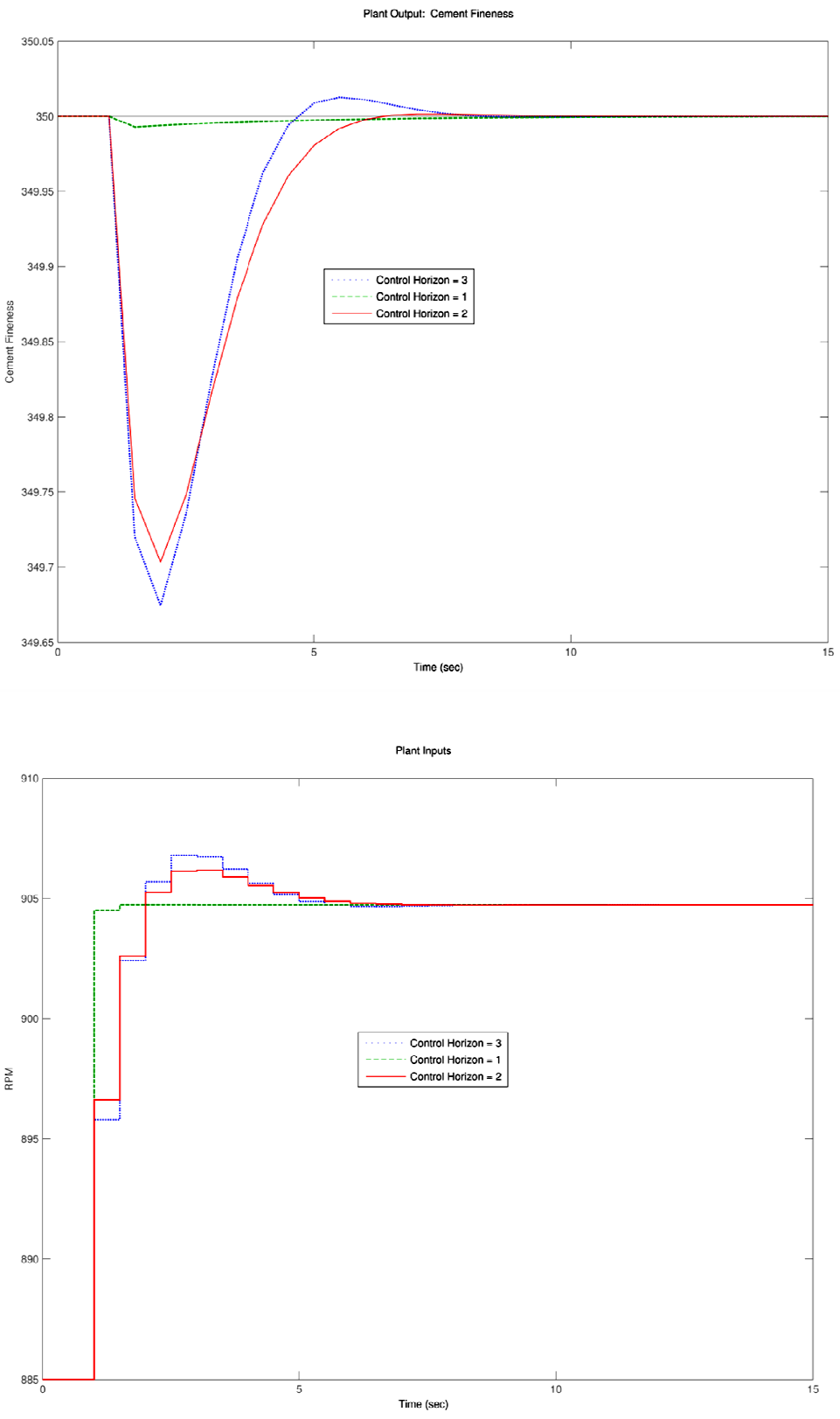


Figure 5.16: Cement fineness control for +10% step change of clinker inflow rate above the steady state value.

Two important parameters in MPC are the lengths of prediction and control horizons (Qin and Badgwell, 2003). Prediction horizon (*P*) refers to the time interval over which the output is predicted and control horizon (*M*) refers to the time interval over which the control moves are computed. Usually *P* is greater than M and the recommended values are *P* = 20 and *M* = 1 to 3 (Bequette, 2003).

In this work, the performance of the MPC was compared by using a prediction horizon of 20 and different values of control horizons ranging from 1 to 3. The simulated response of the changes in manipulated variable and the corresponding changes in output are shown in Figure 5.16 for a 10% step change in the load variable (clinker inflow rate) above a nominal steady state value of 200 tons per hour.

It can be observed that increase in control horizon does not bring any significant improvement. A higher control horizon results in increase in overshoot due to the larger magnitude of initial changes in the manipulated variables (Shridhar and Cooper, 1997). It can be concluded from Figure 5.16 that a control horizon of 1 gives better performance than other values.

5.3.2 Results for Clinkerization Process

As described in Section 4.2.3, the soft sensor model for rotary cement kiln was developed using the kiln data which were processed and divided using two different methods. The pair of datasets obtained are referred as dataset 1 and dataset 2. All models developed have nine inputs and eight outputs. The models are evaluated by measuring six different statistical performance indices. The formulae of these performance indices for a single output quality y_i have been presented in Equations 5.3 and 5.7 to 5.11. Therefore, for any model corresponding to eight outputs, eight values are obtained for a single performance index. The model performance values reported in the subsequent sections correspond to the average of the eight values. For example, for a particular model, for eight outputs, eight values of MAE were obtained. The value of MAE reported in a particular Table is the average of these eight MAE values. The same logic holds good for other statistical parameters as well.

5.3.2.1 Results for Kiln Dataset 1

Dataset 1 was formed by application of Hampel's method of univariate outlier detection followed by random division of the processed data to 156 number of training samples and 67 number of validation samples. Three types of feed forward neural network models were developed using the training dataset. The details of the three network structures have already been provided in Table 4.7. The performances of these models are presented in Tables 5.11(a) and 5.11(b) for the training and validation data respectively. The performances of the three neural network models in predicting the eight clinker quality parameters are shown in Figure 5.17.

One interesting observation is, as far as simulation of the networks for the training data is concerned, all three models produce satisfactory results (Table 5.11(a)). However, the performance of the GRNN model is superior to the other two models because of the use of large number of pattern layer neurons (Table 4.7). But the important aspect where

the BP network model lags behind the radial basis network and the regression network model is the generalization capability of the models (Table 5.11(b)) i.e. how well the models perform when they are supplied with data not used for the training. It is quite obvious from Table 5.11(b) that RBFN and GRNN clearly outperform the network model trained by back propagation method. The performance of a model is assessed by its ability for generalization.

Table 5.11(a): Neural network Model Performance for training data in clinkerization process (Dataset 1)

MODEL TYPE	MAE	RMSE	CORRELATION COEFFICIENT	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value	**0**	**0**	**1**	**100**	**1**	**0**
BPNN	0.0612	0.0822	0.87	75.93	0.7593	0.0955
RBFNN	0.065	0.0918	0.8397	70.55	0.7055	0.1033
GRNN	0.0351	0.061	0.9357	86.99	0.87	0.0683

Table 5.11(b): Neural network Model Performance for validation data in clinkerization process (Dataset 1)

MODEL TYPE	MAE	RMSE	CORRELATION COEFFICIENT	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value	**0**	**0**	**1**	**100**	**1**	**0**
BPNN	0.4033	0.817	-0.131	-2549.3	-27.65	0.592
RBFNN	0.1321	0.1696	0.188	-22.76	-0.542	0.2018
GRNN	0.1517	0.195	0.092	-51.94	-0.674	0.2228

Detailed analysis of the results of Table 5.11(b) reveals that the BPNN model has very low generalization capability. The high negative variance account for (VAF) values for the BPNN model is due to the fact that the model exhibits highly erratic behavior in estimating outputs from unknown inputs, resulting in much higher variance for the residuals ($y - \hat{y}$) than for the actual output. Clinker quality parameters are better predicted by RBFN than GRNN as evident from the values of different statistical performance indices in Table 5.11(b).

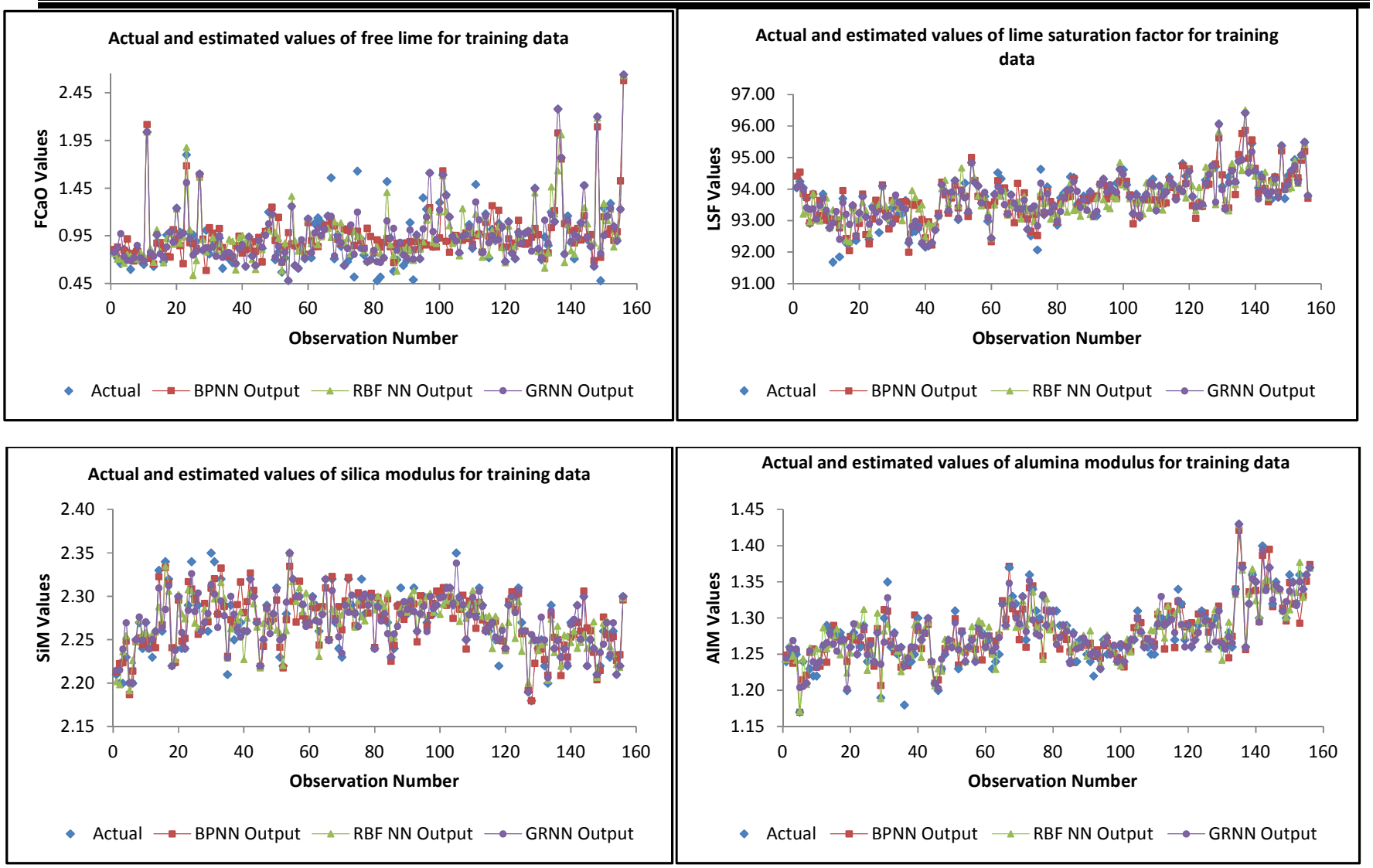


Figure 5.17(a): Prediction performances of the three neural networks for training data

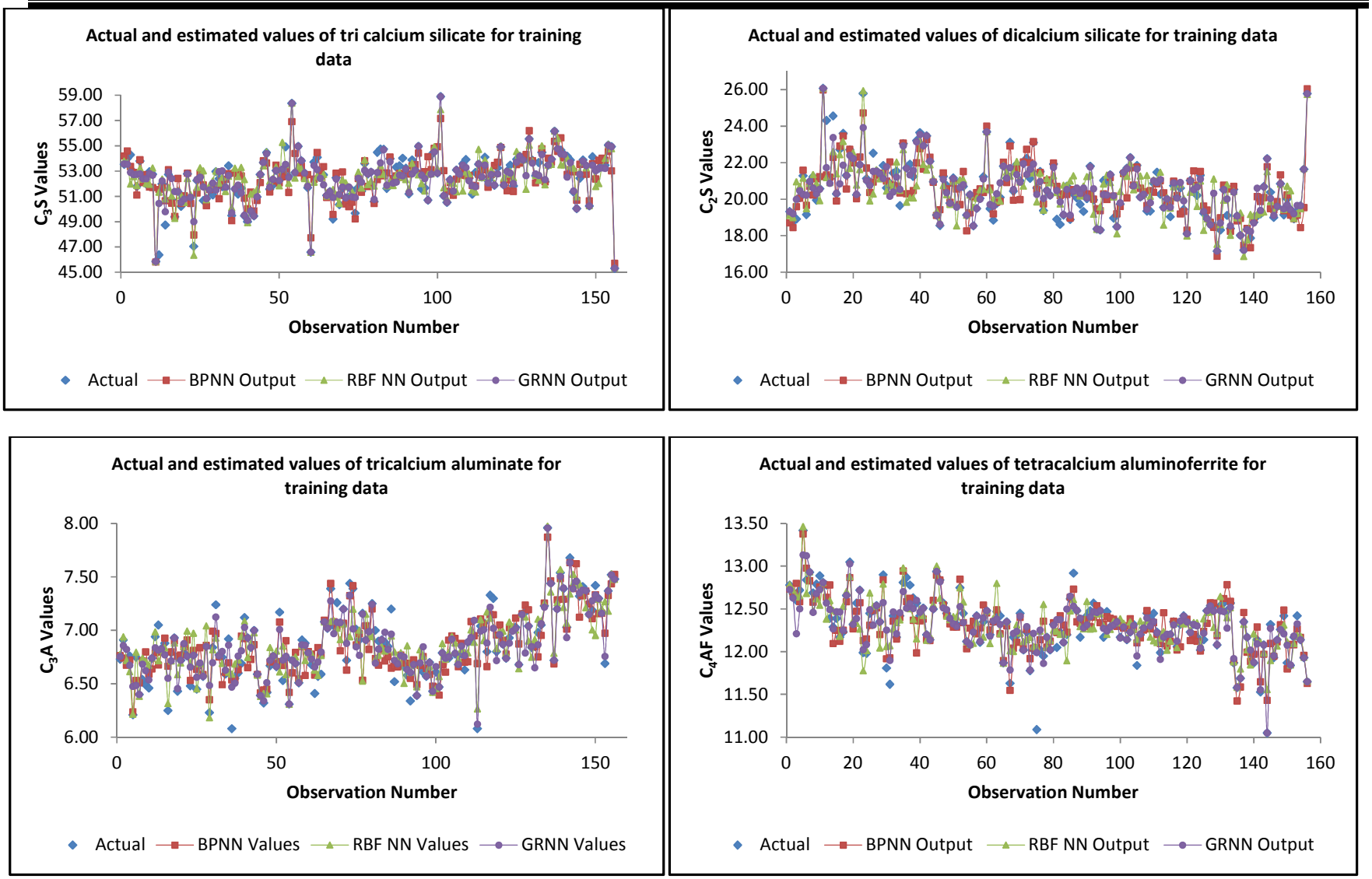


Figure 5.17(a): Prediction performances of the three neural networks for training data (Continued)

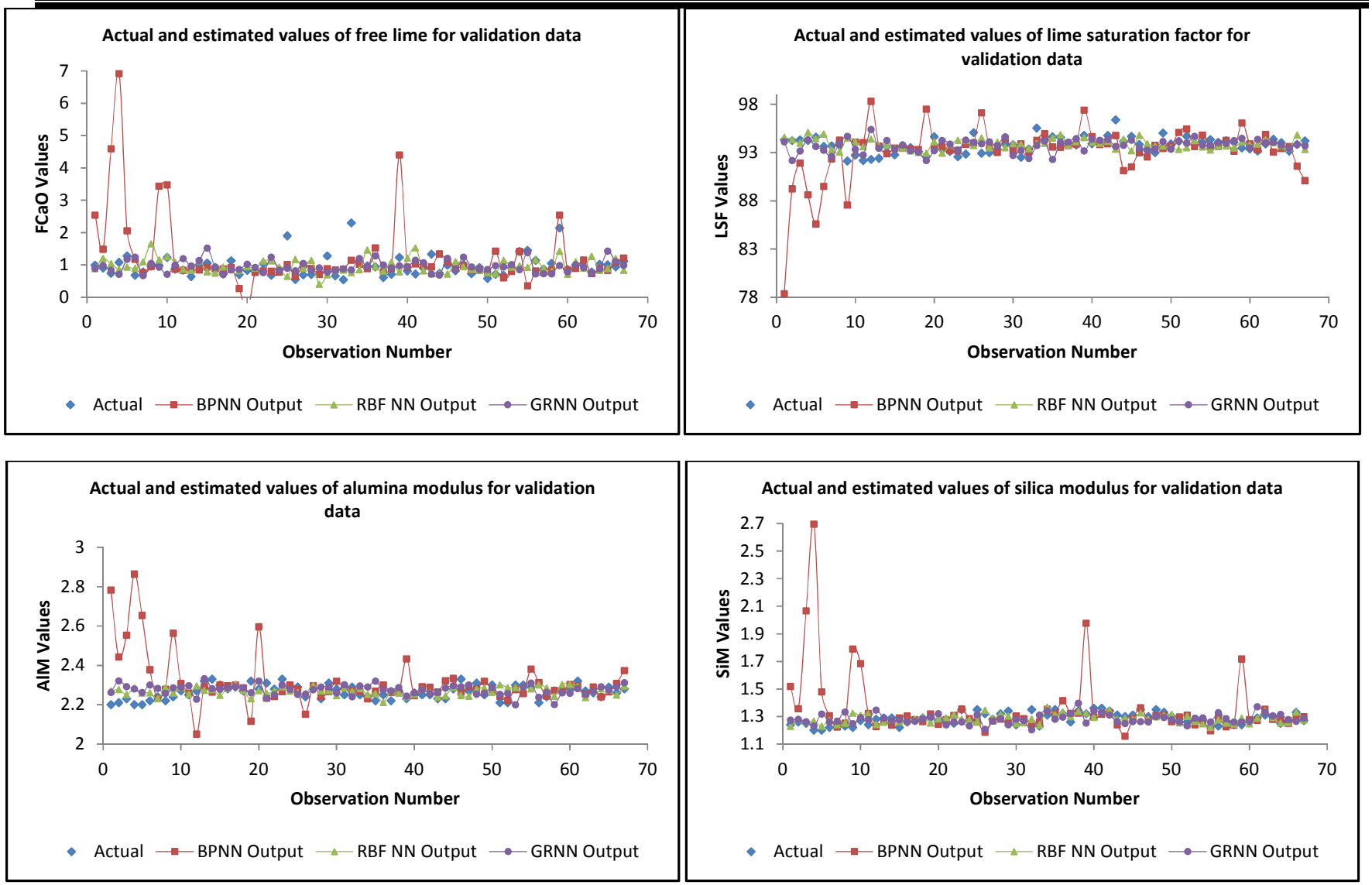


Figure 5.17(b): Prediction performances of the three neural networks for validation data

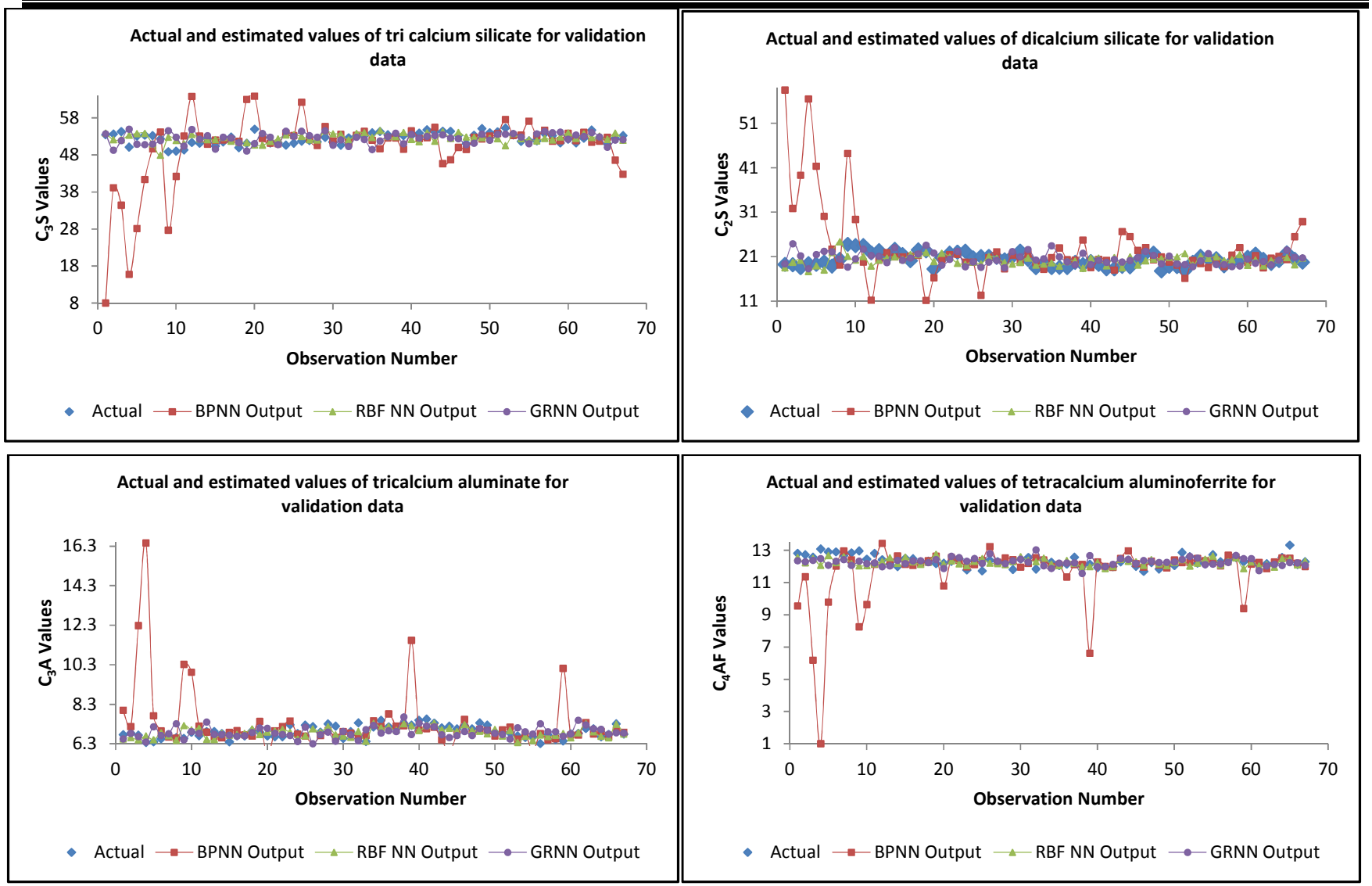


Figure 5.17(b): Prediction performances of the three neural networks for validation data (Continued)

However, the best performance may not be satisfactory. Even in case of RBFNN model, there is significant difference of performance with the training and validation data. This might be due to the fact that the statistical characteristics of the training data might have been significantly different than that of the validation data. This difference is due to the fact that, random sampling does not ensure uniform characteristics of the datasets. Moreover, there may be some flaws in the initial processed data itself due to the application of univariate outlier detection techniques to the multivariate data. These problems are addressed by applying multivariate outlier detection technique to the raw data and use of statistical techniques of subset selection instead of random data division. Application of closest distance to center (CDC) method of multivariate outlier detection resulted in a different set of processed data (Total number of observations are same) which was subsequently divided equally into training and validation subsets. This is referred to as Dataset 2.

5.3.2.2 Results for Kiln Dataset 2

The training set prepared by application of the Kennard-Stone method to the processed data, was used for data-driven model development of rotary cement kiln. It is mentioned in Section 2.3.2 that the model reported by Lin *et al*. (2007) for soft sensing of free lime has used different kiln operating variables as inputs. However, the input raw meal quality has not been considered in modeling. Therefore, before proceeding further with model development it was investigated whether the inclusion of raw meal quality as inputs improves the model performance or not. To address this issue, initially two types of multiple linear regression models for the kiln were developed, one only with kiln operating variables as inputs (Total 5 inputs - Column 2 of Table 4.5) and the other using kiln operating variables as well as raw meal quality as inputs (Total 9 inputs - Both columns 1 and 2 of Table 4.5). The linear regression equations are reported in Table 4.8.

The model performance with and without raw meal quality values as inputs are presented in Table 5.12.

Table 5.12: Multiple linear regression model performance with and without input material quality values as model inputs

NUMBER OF INPUTS	MODEL PERFORMANCE	
	MAE (Training set)	MAE (Validation set)
5 (Only kiln operating variables)	0.1056	0.096
9 (Kiln operating variables + Raw meal quality)	0.0899	0.0874

From Table 5.12 it is quite clear that inclusion of input quality as model inputs significantly improves the model performance (The MAE has decreased by 17% for training set and 10% for the validation set). However, it must be noted that as mentioned earlier the input quality values used for performance evaluation are the most recent values as obtained from the quality control laboratory as these are not continuously measured and hence instantaneous values are not available.

The available training and validation datasets as explained in Section 4.2.3 were subsequently used for development and analysis of the various data-driven soft sensor models of the rotary cement kiln. The datasets contain 9 input parameters and eight output (clinker quality) parameters. A total of 112 samples were used for training and rest 111 samples for validation.

The optimum structure of multilayer perceptron (BPNN) was determined by constructing neural network models with different number of hidden layer neurons followed by training with different algorithms so as to achieve the minimum error for the validation dataset. The results are shown in Figure 5.18.

For design of back propagation neural network model four different training algorithms were used: gradient descent with momentum (GDM), resilient back propagation (RP), Levenberg-Marquadt (LM) and conjugate gradient descent (CG). LM algorithm though quite fast and gives good fitting with less number of epochs, it suffers

from the problem of over fitting as evident from the very high error for the validation data. Both Gradient Descent with Momentum (GDM) and Resilient Back Propagation (RP) algorithms performed equally well with the validation data. However considering the marginal better result of GDM with training data, the optimum network structure was decided to be the one having 15 sigmoidal hidden layer neurons trained with gradient descent with momentum (GDM).

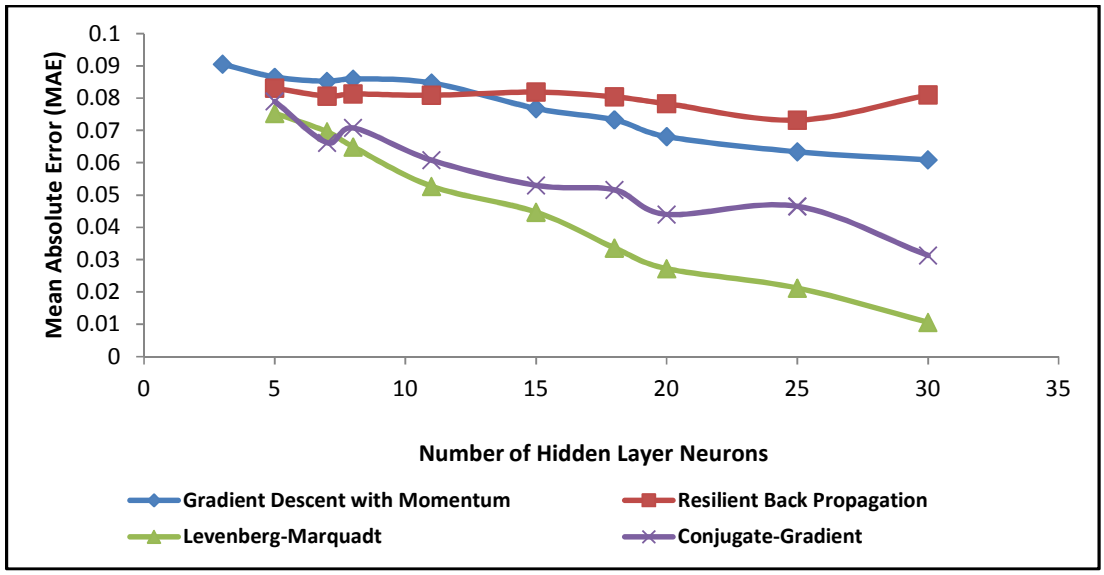


Figure 5.18 (a): Neural network model performance trained by different algorithms (Training data)

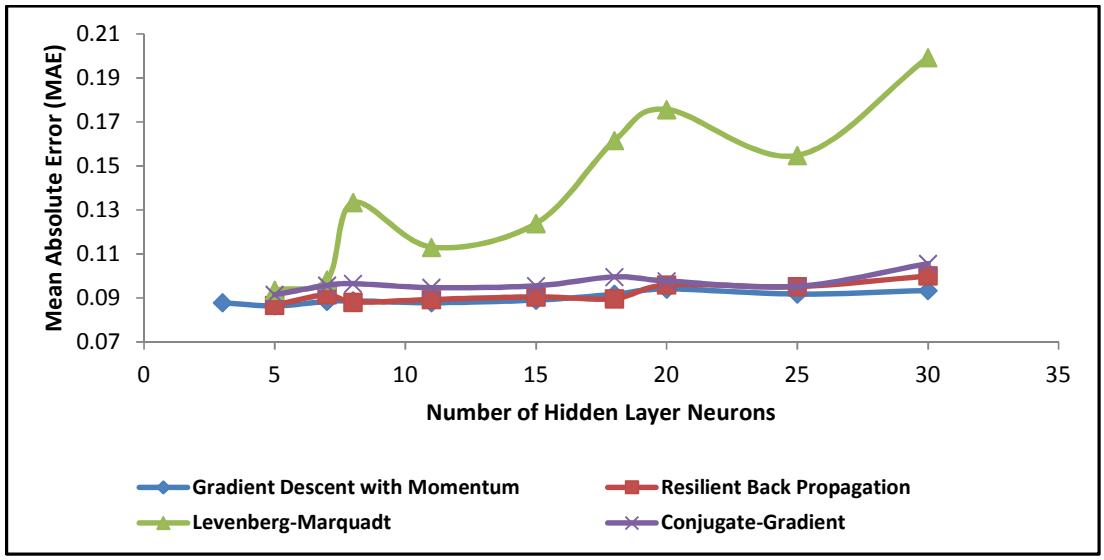


Figure 5.18 (b): Neural network model performance trained by different algorithms (Validation data)

Ideally an RBF model should have as many number of hidden layer neurons as the number of training samples (inputs) for perfect fitting. However, this perfect fitting with training data results in very poor generalization. Therefore, the aim was optimum selection of spread and number of neurons so that the network produces low and almost comparable error values for both training and validation data. Initially to determine the optimum spread parameter, networks with fixed number of neurons and different scaling parameters were created. Performances of these RBFNNs with training and cross validation data are shown in Figure 5.19(a). From this analysis, the optimum scaling parameter was decided as 2. Subsequently at this spread value, networks with different number of RBF neurons were created and tested. The results are shown in Figure 5.19(b).

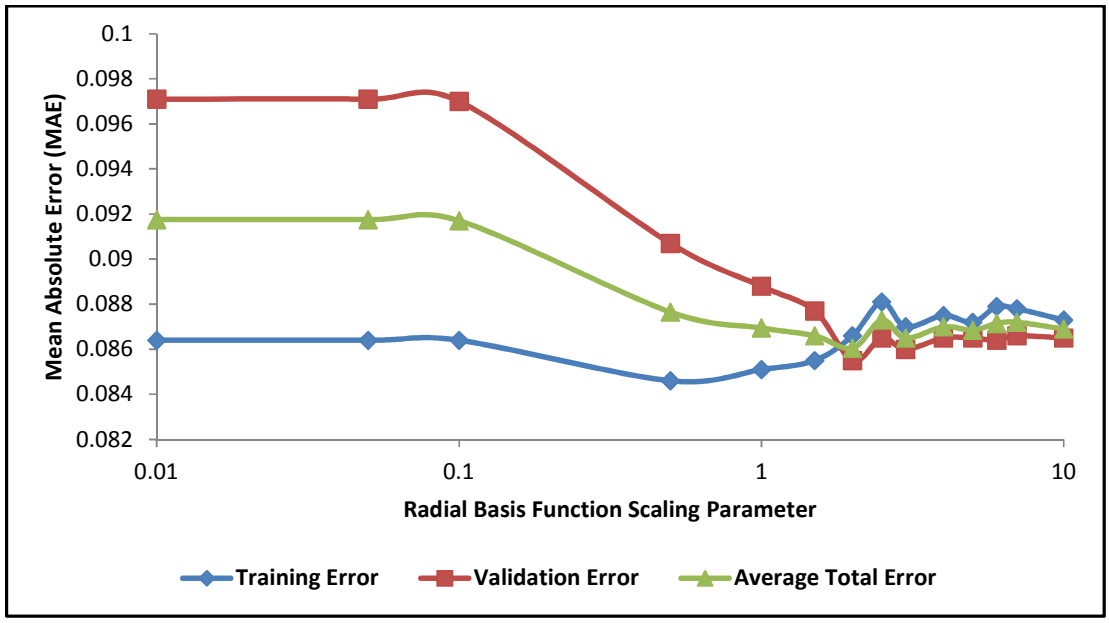


Figure 5.19(a): RBF neural network model performance as a function of scaling parameter at constant number of neurons (10) for clinkerization process

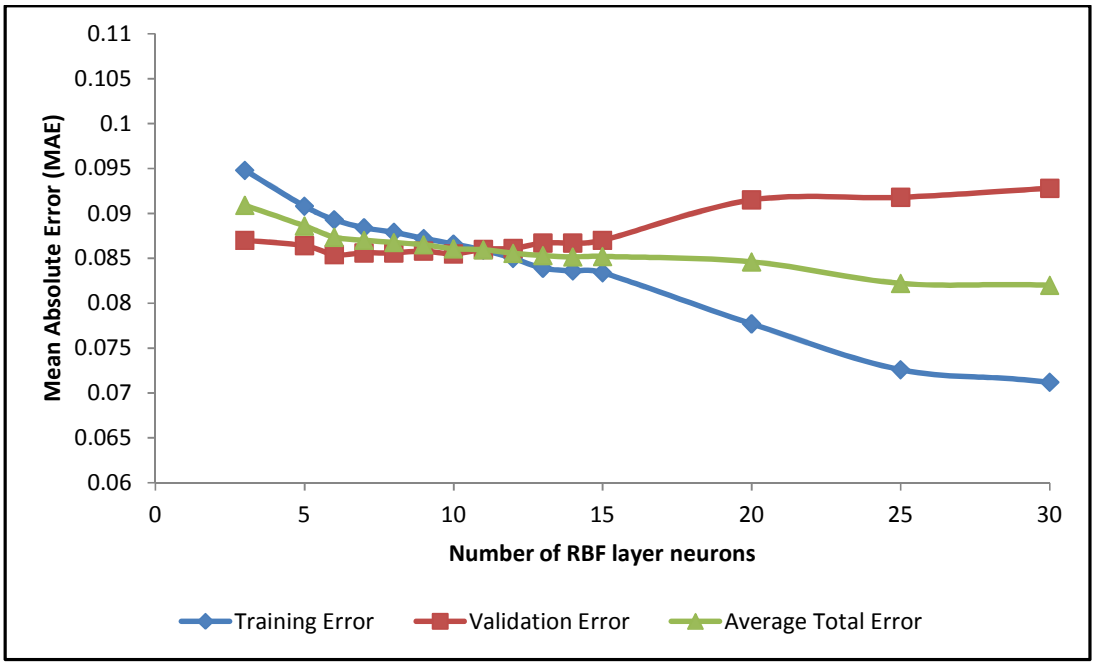


Figure 5.19(b): RBF neural network model performance as a function of number of neurons at constant scaling parameter value of 2 for clinkerization process

In the similar manner the optimum spread parameter of the regression network was determined. The results leading to optimum spread value are produced in Figure 5.20.

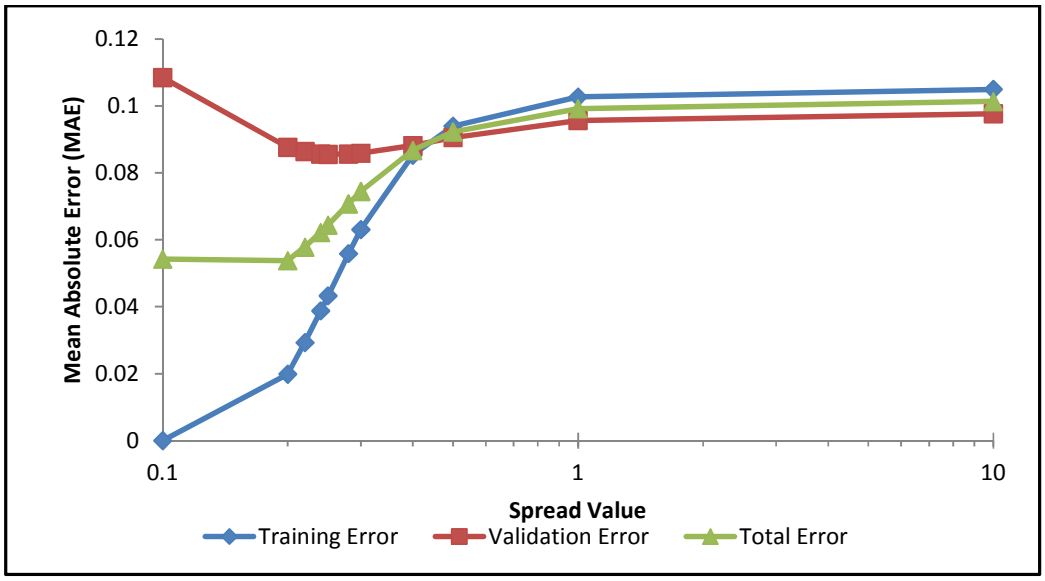


Figure 5.20: Regression neural network model performance as a function of spread value

The optimum structures of the three neural network models have already been presented in Table 4.7. The performances of these three neural network models were

assessed using the same statistical parameters which were used for the grinding process models. The results are reported in Tables 5.13.

Similarly using the procedure described in Section 4.2.3.2, Mamdani and Sugeno type fuzzy inference models of the kiln process were developed. The rules were framed based on the values of the training data and the optimum model in each case was chosen as the one that produced the least MAE for the validation dataset. A total of 147 rules were framed for the Mamdani model and 129 fuzzy *if-then* rules were framed for the Sugeno type model. The performances of the two fuzzy inference models are reported in Table 5.14.

The ideal model values for the different statistical parameters are mentioned in the top row. The closer the model values are to the ideal values, the better is the model fitness. The values in Tables 5.11-5.14 represent the average value for the eight output variables. NSE value ranges from -∞ to 1 and TIC value lies between 0 and 1. Good models should have NSE values higher than 0 and TIC value less than 0.3 (Moriasi *et al*., 2007).

The advantage of using multivariate outlier detection and structured subset selection method can be clearly observed by comparing the performances of the neural network models reported in Table 5.11 with the ones reported in Table 5.13. While the best performing RBF model produced an error of 13% for the validation data, in the second scenario, all models have validation error values significantly less than this. Logic for using six model evaluation parameters is also clearly visible for the clinkerization process. A comparison of the MAE and RMSE values of BPNN and RBFNN shows that while with respect to RMSE, RBFNN may seem better, but with respect to MAE, BPNN seems better. The reasons have already been discussed under the results for grinding process in Section 5.3.1.

Table 5.13(a): Neural network model performance for training data in clinkerization process (Dataset 2)

MODEL TYPE	MODEL DETAILS	MAE	RMSE	CORRELATION COEFFICIENT (R)	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value		**0**	**0**	**1**	**100**	**1**	**0**
BPNN	15 hidden neurons, trained by gradient descent with momentum (GDM)	0.077	0.113	0.589	34.567	0.338	0.180
RBFNN	14 neurons, Scaling parameter = 2	0.084	0.111	0.582	52.34	0.345	0.161
GRNN	Optimum Spread Parameter = 0.25	0.043	0.052	0.924	85.577	0.798	0.089

Table 5.13(b): Neural network model performance for validation data in clinkerization process (Dataset 2)

MODEL TYPE	MODEL DETAILS	MAE	RMSE	CORRELATION COEFFICIENT (R)	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value		**0**	**0**	**1**	**100**	**1**	**0**
BPNN	15 hidden neurons, trained by gradient descent with momentum (GDM)	0.089	0.126	0.335	12.445	0.038	0.207
RBFNN	14 neurons, Scaling parameter = 2	0.087	0.122	0.342	46.902	0.092	0.180
GRNN	Optimum Spread Parameter = 0.25	0.086	0.122	0.353	48.411	0.081	0.181

Table 5.14(a): Fuzzy inference Model Performance for training data in clinkerization process

MODEL TYPE	MODEL DETAILS	MAE	RMSE	CORRELATION COEFFICIENT (R)	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value		**0**	**0**	**1**	**100**	**1**	**0**
Fuzzy Inference (Mamdani)	5 fuzzy variables for each input (triangular and trapezoidal membership function) 5 fuzzy variables for each output (trapezoidal membership function) Total 147 rules	0.062	0.080	0.808	77.965	0.378	0.119
Fuzzy Inference (Sugeno)	5 fuzzy variables for each input (triangular and trapezoidal membership function) 9 fuzzy variables for each output (Zero order) Total 129 rules	0.054	0.090	0.807	71.028	0.211	0.133

Table 5.14(b): Fuzzy inference Model Performance for validation data in clinkerization process

MODEL TYPE	MODEL DETAILS	MAE	RMSE	CORRELATION COEFFICIENT (R)	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value		**0**	**0**	**1**	**100**	**1**	**0**
Fuzzy Inference (Mamdani)	5 fuzzy variables for each input (triangular and trapezoidal membership function) 5 fuzzy variables for each output (trapezoidal membership function) Total 147 rules	0.087	0.122	0.524	48.078	0.024	0.173
Fuzzy Inference (Sugeno)	5 fuzzy variables for each input (triangular and trapezoidal membership function) 9 fuzzy variables for each output (Zero order) Total 129 rules	0.081	0.118	0.571	49.943	0.089	0.167

Analyzing the results presented in Tables 5.13 and 5.14, it can be concluded that the two fuzzy inference models (Mamdani and Sugeno) and the regression neural network model have better performances than the BPNN and RBFNN models. The performances of these three models with the training and validation data are shown in Figure 5.21(a) and 5.21(b), respectively. To choose the best out of these three models, further analysis was done by studying the statistical parameters produced by these three models for each of the 8 outputs. The results are presented in Table 5.15.

Considering the average values it can be concluded that the regression neural network (GRNN) model has the best performance for the training data and Sugeno type fuzzy inference model (FISS) has the best performance for the validation data (The best average values are highlighted in Table 5.15). Performance of the models with respect to prediction of the individual outputs helps further in choosing the best model. Analyzing the model performance with validation results of Table 5.15 (b), it is clear that GRNN has the worst performance. While Mamdani type fuzzy inference system (FISM) shows poor prediction accuracy for SiM and C_3S, Sugeno type fuzzy inference system (FISS) shows low accuracy for LSF, SiM and C_3S. However, this lower accuracy comes with the advantage of better accuracy in predicting free lime.

While the significance of the different quality parameters in determining the cement quality has been mentioned in the process description, free lime content happens to be the most important quality parameter which ultimately decides whether the clinker is to be rejected or processed further in the grinding unit. Therefore, it was concluded that the FISS model has the best performance because it has the best overall prediction accuracy as well as best prediction accuracy for free lime content.

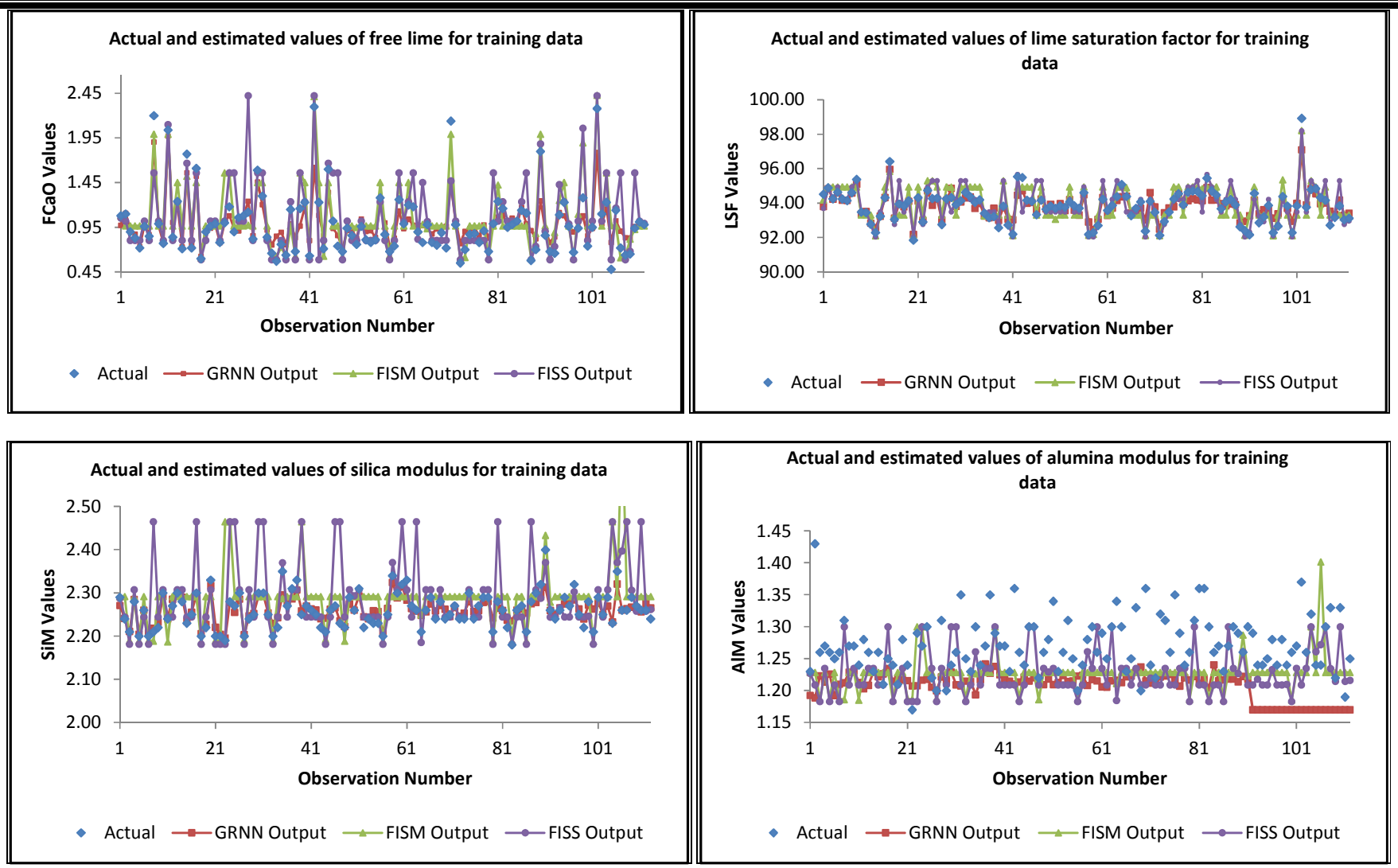
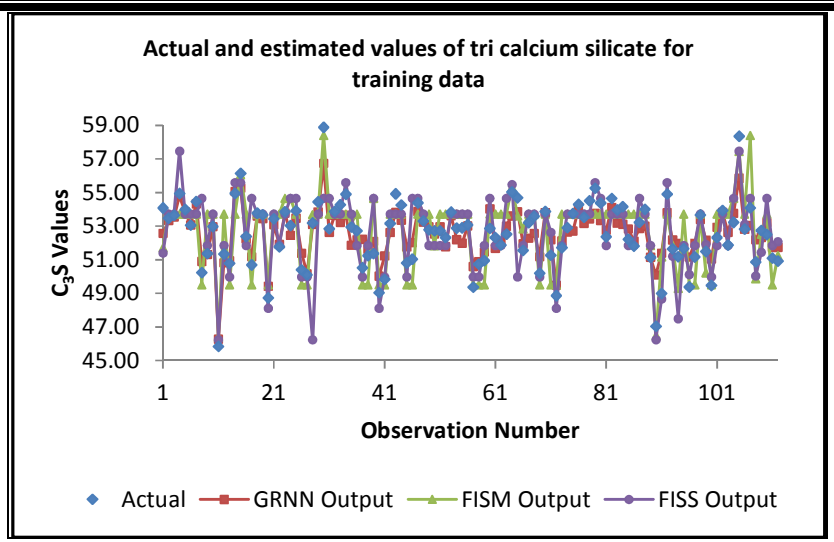
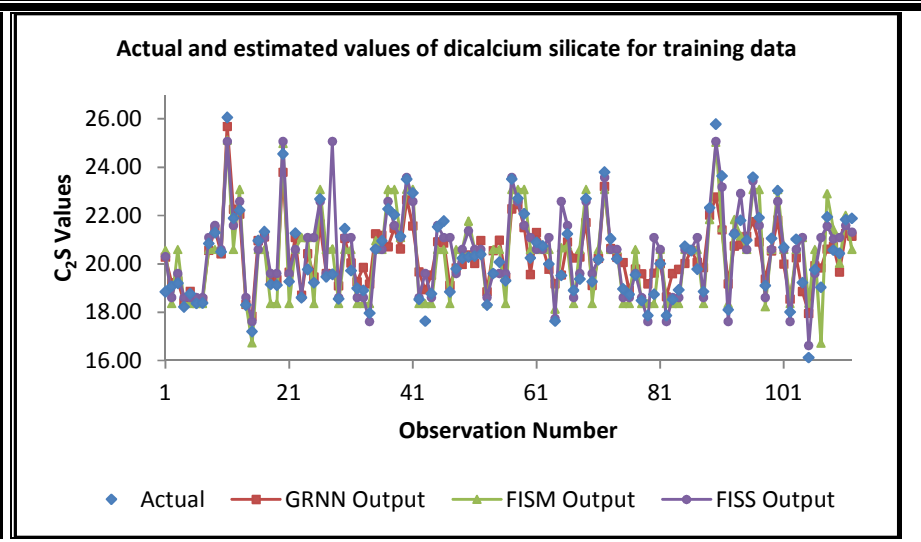
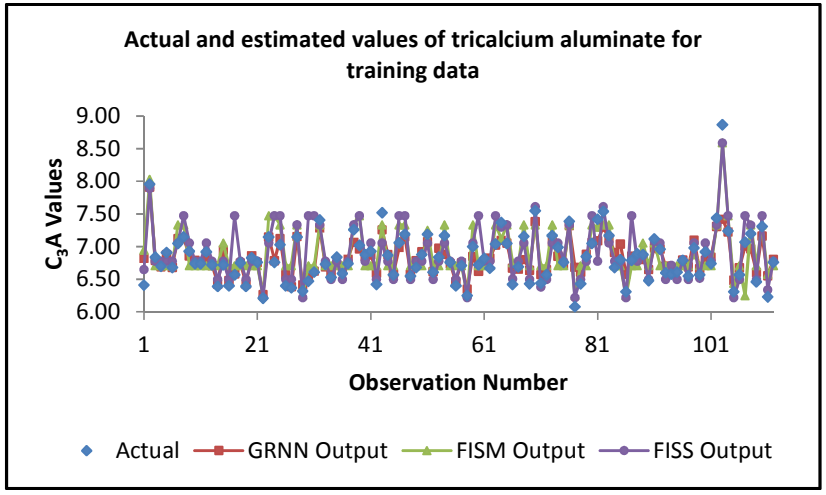


Figure 5.21(a): Performances of the three data-driven soft sensors with training data







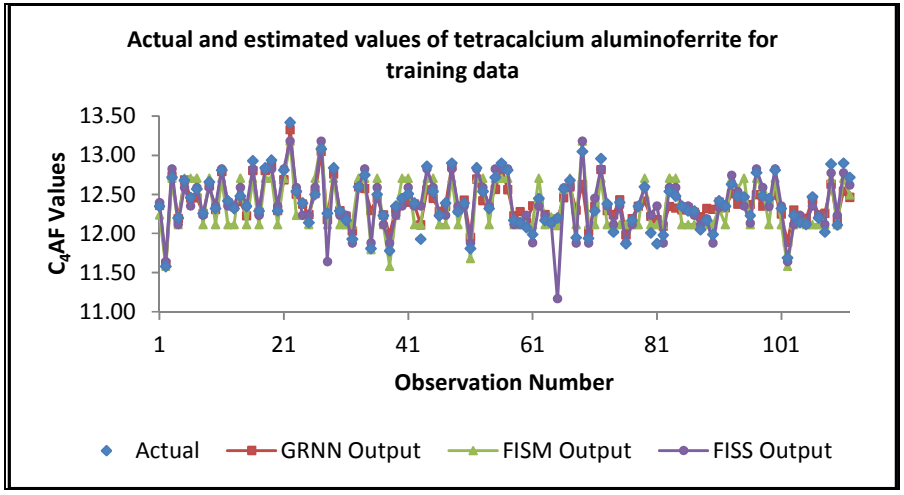


Figure 5.21(a): Performances of the three data-driven soft sensors with training data (Continued)

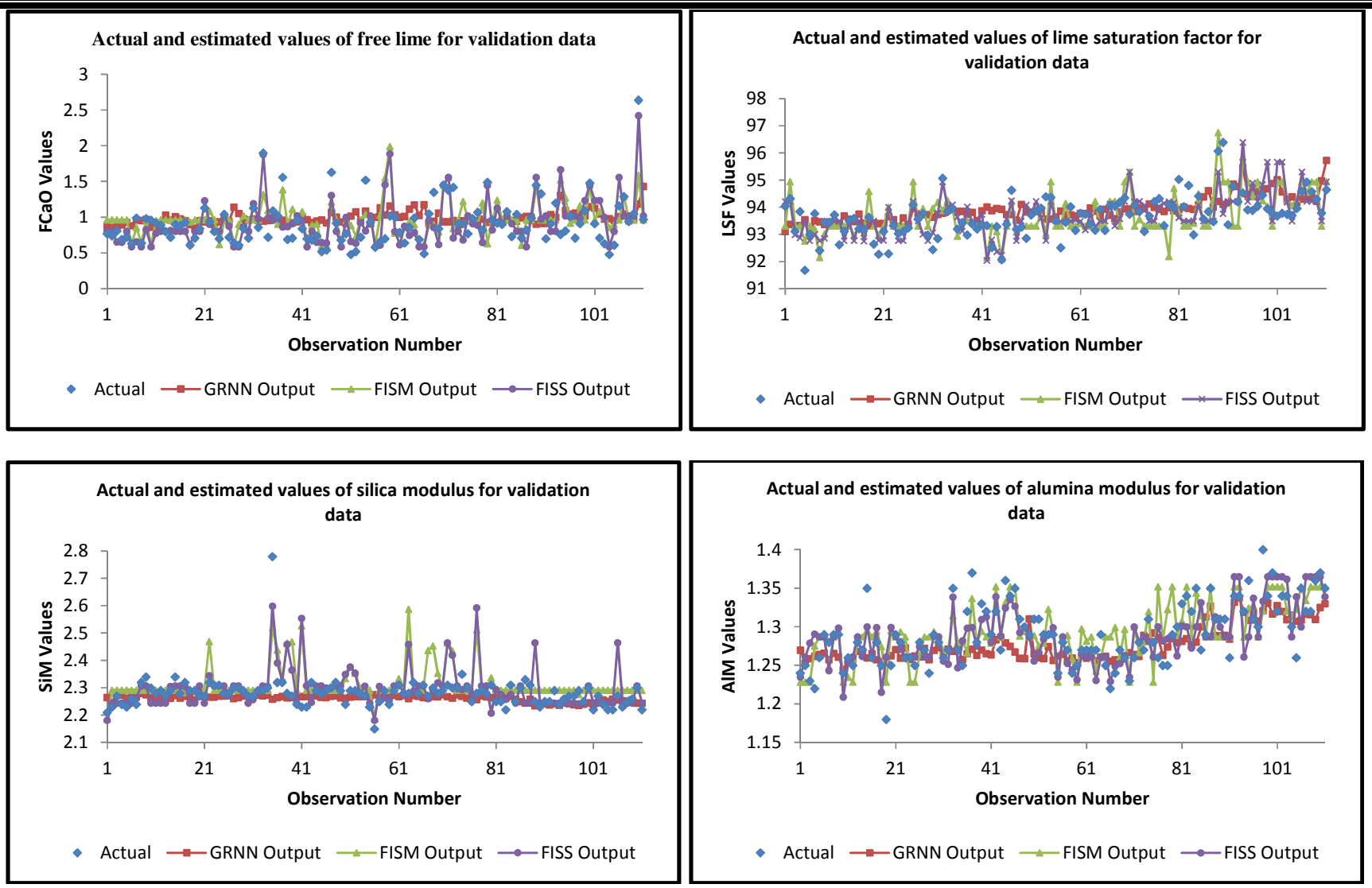
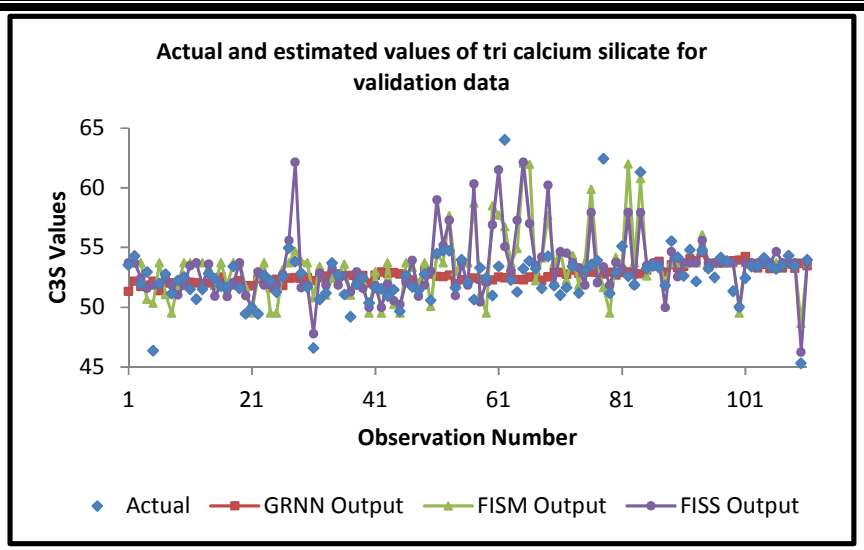
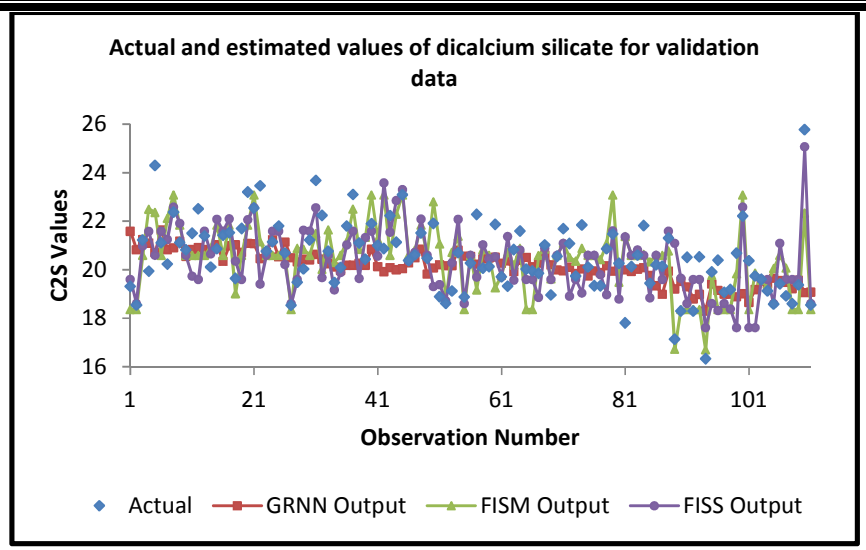
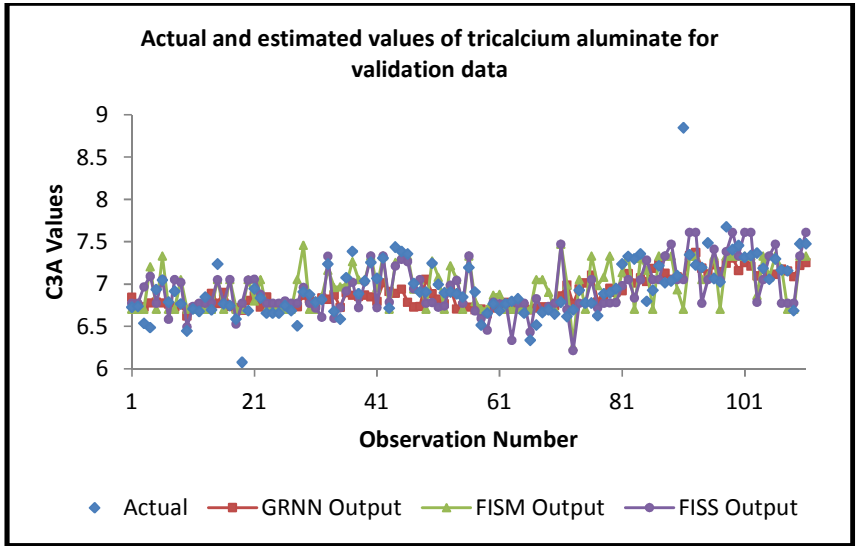


Figure 5.21(b): Performances of the three data-driven soft sensors with validation data







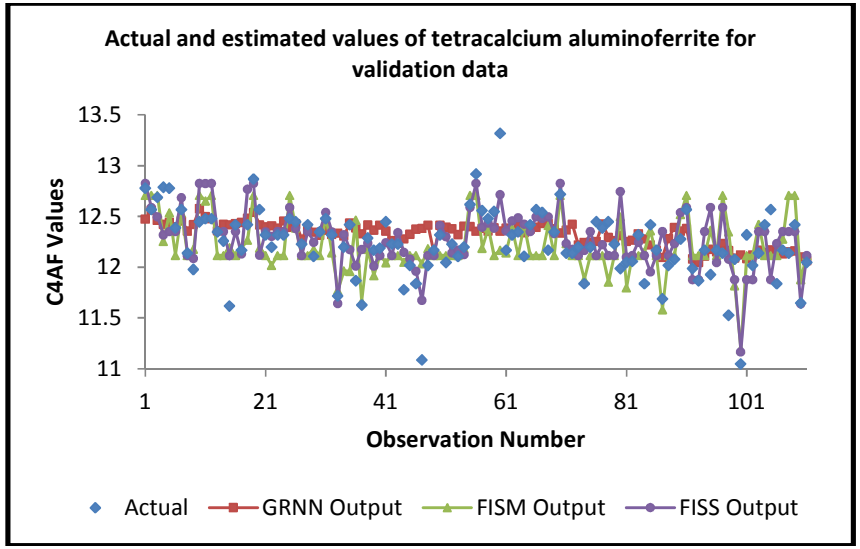


Figure 5.21(b): Performances of the three data-driven soft sensors with validation data (Continued)

Table 5.15 (a): Detailed model evaluation for training data in clinkerization process

Output Variable (Clinker Quality)	MODEL PERFORMANCE																	
	MAE			RMSE			R			VAF			NSE			TIC		
	GRNN	FISM	FISS	GRNN	FISM	FISS	GRNN	FISM	FISS	GRNN	FISM	FISS	GRNN	FISM	FISS	GRNN	FISM	FISS
FCaO	0.054	0.063	0.070	0.074	0.081	0.126	0.933	0.901	0.785	78.08	79.66	46.76	0.781	0.764	0.427	0.141	0.133	0.201
LSF	0.041	0.066	0.049	0.046	0.076	0.074	0.942	0.865	0.876	88.62	80.13	80.66	0.829	0.683	0.701	0.087	0.113	0.110
SiM	0.022	0.064	0.070	0.025	0.112	0.125	0.908	0.337	0.516	96.54	61.59	51.28	0.763	-2.085	-2.846	0.084	0.261	0.281
AlM	0.049	0.064	0.044	0.061	0.077	0.075	0.940	0.892	0.899	84.23	78.74	79.70	0.847	0.792	0.803	0.079	0.090	0.088
C_3S	0.036	0.050	0.048	0.040	0.063	0.077	0.929	0.843	0.769	92.16	85.73	78.69	0.807	0.643	0.474	0.058	0.078	0.095
C_2S	0.061	0.067	0.053	0.067	0.083	0.090	0.926	0.896	0.883	76.60	75.22	72.72	0.803	0.791	0.756	0.090	0.090	0.096
C_3A	0.041	0.062	0.056	0.050	0.079	0.090	0.891	0.848	0.828	81.86	79.33	73.46	0.744	0.684	0.589	0.120	0.126	0.141
C_4AF	0.044	0.061	0.039	0.055	0.070	0.065	0.922	0.884	0.899	86.49	83.29	84.92	0.809	0.747	0.787	0.053	0.061	0.056
Average	**0.043**	0.062	0.054	**0.052**	0.080	0.090	**0.924**	0.808	0.807	**85.57**	77.96	71.02	**0.798**	0.378	0.211	**0.089**	0.119	0.133

Table 5.15 (b): Detailed model evaluation for validation data in clinkerization process

Output Variable (Clinker Quality)	MODEL PERFORMANCE																	
	MAE			RMSE			R			VAF			NSE			TIC		
	GRNN	FISM	FISS	GRNN	FISM	FISS	GRNN	FISM	FISS	GRNN	FISM	FISS	GRNN	FISM	FISS	GRNN	FISM	FISS
FCaO	0.113	0.096	0.081	0.150	0.134	0.116	0.118	0.498	0.677	22.49	37.90	51.06	-0.063	0.151	0.360	0.307	0.266	0.233
LSF	0.078	0.078	0.074	0.100	0.099	0.102	0.408	0.550	0.507	67.59	64.64	62.69	0.009	0.030	-0.022	0.165	0.167	0.172
SiM	0.052	0.075	0.065	0.096	0.121	0.112	0.120	0.358	0.462	69.88	56.80	57.13	-0.087	-0.741	-0.478	0.235	0.239	0.230
AlM	0.099	0.090	0.087	0.131	0.114	0.116	0.621	0.708	0.695	47.03	53.76	51.41	0.281	0.457	0.432	0.143	0.115	0.120
C_3S	0.077	0.098	0.093	0.126	0.149	0.149	0.167	0.425	0.428	42.59	27.17	25.76	0.015	-0.370	-0.370	0.157	0.172	0.173
C_2S	0.108	0.093	0.099	0.145	0.121	0.137	0.345	0.647	0.546	30.40	47.84	33.39	0.031	0.326	0.128	0.164	0.130	0.150
C_3A	0.066	0.073	0.074	0.099	0.115	0.107	0.618	0.475	0.576	65.46	52.29	58.76	0.354	0.129	0.254	0.155	0.171	0.161
C_4AF	0.092	0.094	0.078	0.132	0.125	0.108	0.429	0.534	0.674	41.84	44.22	59.34	0.108	0.207	0.410	0.125	0.122	0.102
Average	0.086	0.087	**0.081**	0.122	0.122	**0.118**	0.353	0.524	**0.571**	48.41	48.08	**49.94**	0.081	0.024	**0.089**	0.181	0.173	**0.167**

The hybrid model was developed by first conducting PCA on the total normalized input data (223×9 data matrix). The covariance matrix is shown in Table 5.16. Eigen vectors of the covariance matrix are reported in Table 5.17. Each column under Table 5.17 represent one eigen vector. For nine inputs there are total nine eigen vectors. The eigen values for these nine eigen vectors are reported in Table 5.18. The magnitudes of the eigen-values represent the quantity of input data variance represented by the corresponding eigen vector. Based on these eigen values the principal components and their share in the total variance are presented in Table 5.19

Table 5.16: Covariance matrix for the normalized values of kiln inputs

1	-0.13371	0.085632	0.01712	0.001483	0.154242	-0.07059	0.040675	0.17405
-0.13371	1	-0.13703	0.182449	-0.17419	-0.19408	-0.04169	-0.25991	0.016876
0.085632	-0.13703	1	0.193291	0.048916	-0.10767	-0.07348	0.258803	-0.03782
0.01712	0.182449	0.193291	1	0.029276	-0.02808	-0.07784	0.198474	0.06579
0.001483	-0.17419	0.048916	0.029276	1	0.620826	-0.00124	0.558103	-0.03365
0.154242	-0.19408	-0.10767	-0.02808	0.620826	1	0.048149	0.135283	-0.05016
-0.07059	-0.04169	-0.07348	-0.07784	-0.00124	0.048149	1	-0.09817	0.155941
0.040675	-0.25991	0.258803	0.198474	0.558103	0.135283	-0.09817	1	0.168686
0.17405	0.016876	-0.03782	0.06579	-0.03365	-0.05016	0.155941	0.168686	1

Table 5.17: Eigen vectors for the symmetric matrix of Table 5.16

0.06833	-0.39575	0.078109	-0.27609	0.620246	-0.34388	0.472732	0.082192	-0.15232
-0.14435	-0.18159	0.603785	-0.05346	0.30559	0.577412	-0.16181	0.138089	0.326697
-0.00337	0.328436	0.483913	-0.38393	-0.28447	-0.33182	-0.04581	0.536217	-0.17288
0.101105	0.066541	-0.58072	-0.34844	0.156192	0.431122	-0.08013	0.550631	-0.08261
0.688389	-0.10108	0.218129	0.041812	-0.0022	0.239666	-0.18149	-0.15995	-0.59212
-0.48596	0.445256	0.0214	-0.27985	0.290069	0.15367	-0.07166	-0.40135	-0.46135
-0.03972	-0.28176	0.016535	-0.534	-0.53575	0.286615	0.427125	-0.28429	0.034181
-0.49453	-0.46604	0.01152	0.357439	-0.21286	0.085914	0.011191	0.302854	-0.51557
0.089897	0.439382	0.09743	0.398892	0.010145	0.28599	0.721933	0.15573	-0.05406

Table 5.18: Eigen values of the eigen vectors of Table 5.17

0.197	0.496	0.624	0.8302	1.0211	1.1196	1.2136	1.4165	2.0815

Based on the cumulative percentage variance criterion, the usual practice is to choose the number of principal components those account for a cumulative variance of more than 70% or 80%. Therefore, based on the results reported in Table 5.19, a total of five principal components were selected. These five principal components PC_1 to PC_5 are the linear combinations of the original input variables which are formulated based on their respective eigen vector values reported in Table 5.17. The resulting expressions of these five principal components are mentioned in Table 5.20.

Table 5.19: Principal components in the decreasing order of their contribution to data variance

PRINCIPAL COMPONENTS	VARIANCE	PERCENTAGE VARIANCE	CUMULATIVE PERCENTAGE VARIANCE
PC_1	2.0815	23.13	23.13
PC_2	1.4164	15.74	38.87
PC_3	1.214	13.49	52.36
PC_4	1.1196	12.44	64.8
PC_5	1.0211	11.35	76.15
PC_6	0.8302	9.22	85.37
PC_7	0.6242	6.93	92.3
PC_8	0.4959	5.51	97.81
PC_9	0.1972	2.19	100
Total	9.0001	100	

Table 5.20: Details of the five principal components

Principal Components	Expression
PC_1	$PC_1 = -0.1523x_1 + 0.3267x_2 - 0.1728x_3 - 0.0826x_4 - 0.5921x_5 - 0.4613x_6 + 0.0342x_7 - 0.5156x_8 - 0.054x_9$
PC_2	$PC_2 = 0.0822x_1 + 0.1381x_2 + 0.5362x_3 + 0.5506x_4 - 0.16x_5 - 0.4013x_6 - 0.2843 + 0.3028x_8 + 0.1557x_9$
PC_3	$PC_3 = 0.4727x_1 - 0.1618x_2 - 0.0458x_3 - 0.0801x_4 - 0.0815x_5 - 0.0717x_6 + 0.4271x_7 + 0.0112x_8 + 0.722x_9$
PC_4	$PC_4 = -0.3439x_1 + 0.5774x_2 - 0.3318x_3 + 0.4311x_4 + 0.2397x_5 + 0.1537x_6 + 0.2866x_7 + 0.086x_8 + 0.286x_9$
PC_5	$PC_5 = 0.6202x_1 + 0.3056x_2 - 0.2845x_3 + 0.1562x_4 - 0.0022x_5 + 0.29x_6 - 0.5357x_7 - 0.2128x_8 + 0.0101x_9$

Using the five principal components mentioned in Table 5.20, the input dataset dimension was reduced from 223×9 to 223×5. These 5 latent variables were subsequently used for developing BPNN model of the rotary kiln for estimation of clinker quality. Gradient descent with momentum was used as the training algorithm for design of the BPNN model from the 112×5 training data.

Networks with different number of hidden layer neurons were created and trained with training data so that the minimum MAE was attained. Subsequently, the different trained networks were tested with the validation data subset (111×5 validation data subset) and the MAE values were computed. The results are shown in Figure 5.22. Based on the results of Figure 5.22, the optimum number of neurons can be decided as either 18 or 15. Both these structures have the lowest validation errors with lower training error at 18 neurons. The optimum structure was decided as 18 hidden layer neurons where the network produces almost equal error for both training and validation data.

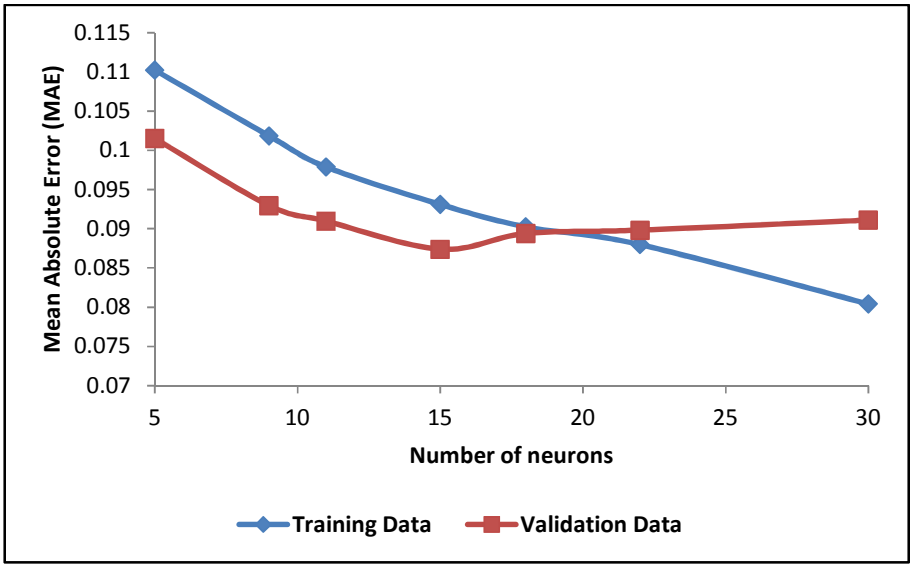


Figure 5.22: Results of BPNN model design from transformed input data

The statistical performance of the PCA - BPNN model is reported in Table 5.21.

Table 5.21: PCA - BPNN Model Performance for the clinkerization process

MODEL DETAILS	MAE	RMSE	CORRELATION COEFFICIENT (R)	VAF	NASH-SUTCLIFFE EFFICIENCY (NSE)	THIEL'S INEQUALITY COEFFICIENT (TIC)
Ideal Value	**0**	**0**	**1**	**100**	**1**	**0**
5 inputs 9 outputs Single hidden layer with 18 neurons Hyperbolic tangent activation function for hidden layer neurons Linear activation function for output layer neurons	Performance with training data					
	0.09	0.128	0.366	8.15	0.075	0.207
	Performance with validation data					
	0.089	0.126	0.293	7.933	0.027	0.208

A comparison of Table 5.21 and 5.13(b) indicates that application of PCA is meaningful to kiln data. Because the performance of the two BPNN models, one with nine actual input values and the other with five latent variables are almost at par with each other. However comparison of Table 5.21 with Table 5.14 reveals that the performance of the fuzzy model is still better in all the statistical performance indices. Therefore, the Sugeno type fuzzy model was subsequently tested in SIMULINK platform for assessing its online quality monitoring capability.

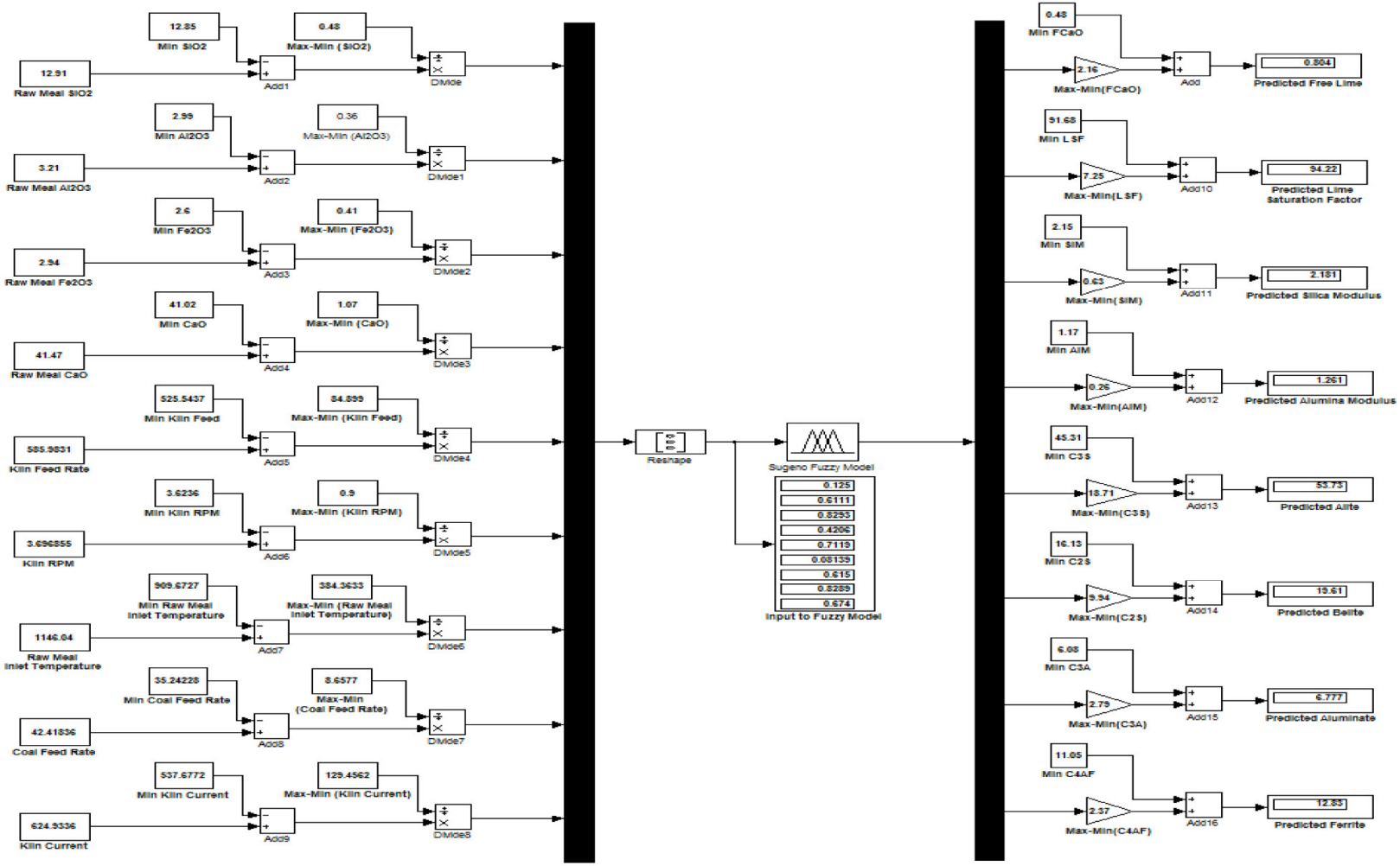


Figure 5.23: SIMULINK block diagram showing online monitoring of cement clinker quality with the fuzzy inference model

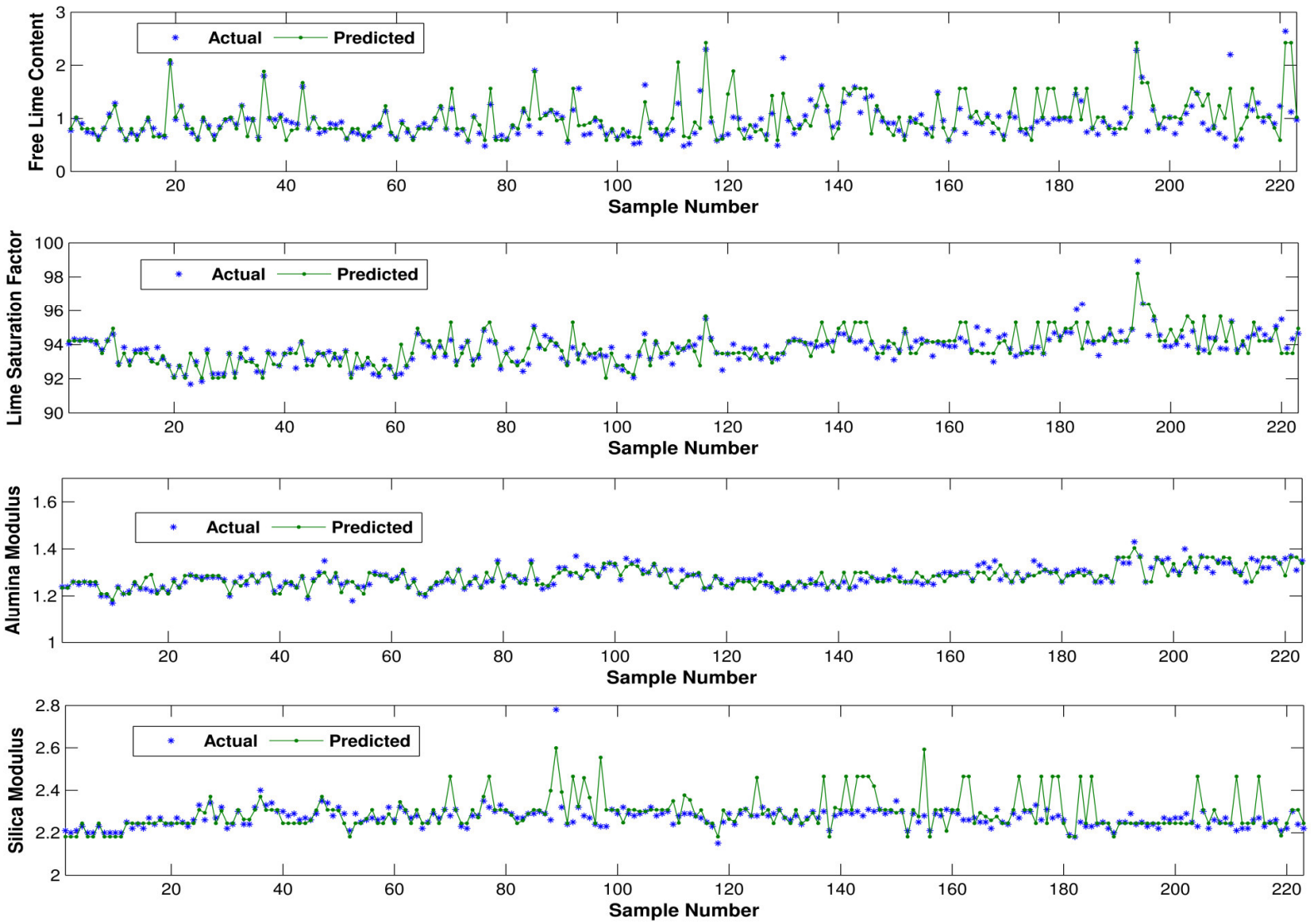


Figure 5.24: Continuously monitored value of clinker quality parameters by the Sugeno Fuzzy Inference Model

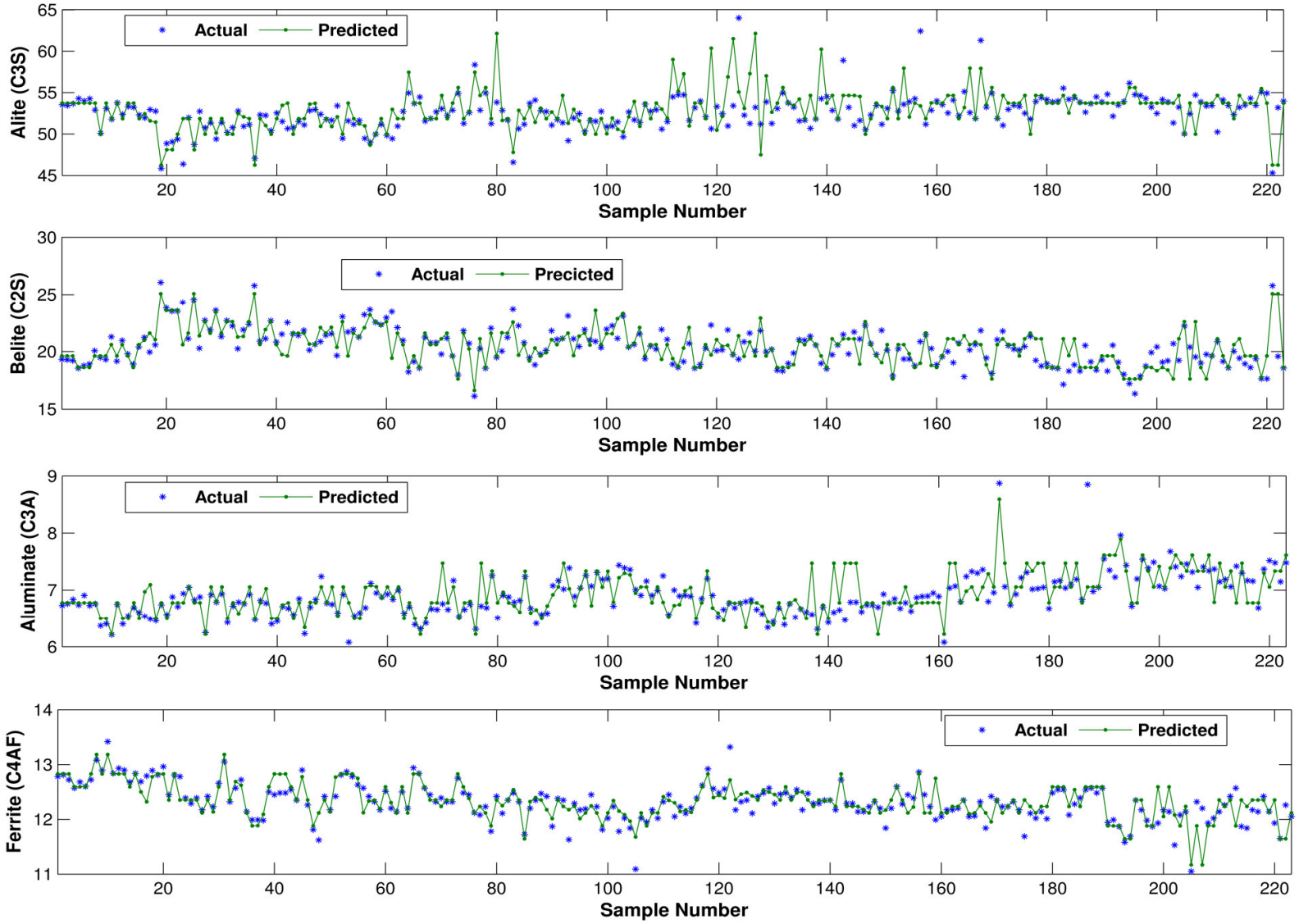


Figure 5.24: Continuously monitored value of clinker quality parameters by the Sugeno Fuzzy Inference Model (Continued)

Online Implementation

The model structure implemented in SIMULINK is shown in Figure 5.23.

The nine blocks on the left receive the values of the input process variables. Out of these nine input blocks, most recently obtained laboratory values for input raw meal quality are supplied to the first four blocks. The other five blocks receive the continuously measured values by the installed hardware sensors. The subsequent couple of blocks describe the process of normalization of each process value in the range 0 to 1. The fuzzy inference model requires a column vector of nine elements.

The normalized individual values are combined with the help of a multiplexor and then subsequently converted to a column vector using the reshape block. This column vector consisting of the normalized input values is the required input to the fuzzy model (a display unit is produced prior to the model to ascertain that the values supplied to the model are in proper format). The model with its membership functions, fuzzy variables and the rule base determines the outputs. The output from the model is a column vector containing 8 elements. These values are separated with the help of a demultiplexor and subsequently denormalized to produce the predicted values in their respective actual units. Figure 5.23 also shows the simulated results for one set of inputs. Finally, the continuously monitored cement clinker quality parameters using the block diagram of Figure 5.23 are shown in Figure 5.24 for all 223 number of observations.

Chapter - 6

Concluding Remarks

In most of the process industries, no hardware sensors are available for continuous monitoring of end product quality. Soft sensors or inferential sensors are industrial process models that use the information of easily monitored process variables and produce estimated values of quality variables. Because of the inherent complexities associated with most of the industrial process, the models are mostly derived from actual input-output process data. Till date, soft sensor development has been reported for product quality prediction in various process industries such as petroleum refinery and petrochemicals industry, polymer industry, fermentation and bio processes industry, metallurgical industry and pulp & paper industry. However, there have been very few applications of soft sensing techniques in cement manufacturing processes. Therefore, the focus of the present research work is development of soft sensors for continuous quality monitoring in cement manufacturing processes.

This chapter presents a summary of the work done followed by significant findings and finally the future scope of research.

6.1 Summary of the Work Done

Two processes were identified where there is no hardware sensor available for online monitoring of product quality. First, a physical process where cement clinker is ground in the vertical roller mill for production of cement. Here, the quality parameter is the product particle size which is measured offline by laboratory analysis. Second, a more complex chemical process occurring in the rotary cement kiln where the raw meal under high

temperature is converted to clinker. Here, the product quality is the chemical composition of the clinker formed which is again measured by offline laboratory sampling.

Actual input-output process data were collected from a cement plant in order to develop data driven soft sensors for estimation of cement fineness and clinker composition. The output of the clinker grinding process i.e. the cement fineness is a function of three input variables. The output of the clinkerization process i.e. the composition of the clinker is a function of nine input variables.

The data collected from the plant were first preprocessed for detection and removal of outliers. The grinding process where only three inputs are involved, univariate outlier detection techniques were applied for removal of abnormal data values. Nine inputs are involved in the clinkerization process taking place in the rotary cement kiln. Therefore, considering the multivariate nature of the data, both univariate as well as multivariate outlier detection techniques were applied. The different univariate outlier detection techniques applied were the three sigma edit rule, box plot method and the Hampel's identifier. For the input data of rotary cement kiln in addition to these three univariate outlier detection techniques, five types of multivariate outlier detection techniques were applied for data preprocessing. The different multivariate techniques applied are, the conventional techniques of Mahalanobis distance, hat matrix leverage value and robust techniques of minimum covariance determinant, smallest half volume and closest distance to center methods.

The total data sets obtained after data preprocesssing were divided equally into training and validation subsets for model development and performance evaluation. In the data division step, random division as well as statistical methods of Kennard-Stone algorithm and DUPLEX algorithms were followed for design of the training set.

For the grinding process, using the designed training data set, five types of statistical models (multiple linear regression, principal component regression, quadratic response surface, standard support vector regression and least square support vector regression), three types of feed forward neural network models (back propagation, radial basis function and regression), two types of fuzzy inference models (Mamdani and Takagi-Sugeno) and two types hybrid models (adaptive neuro fuzzy inference and PCA plus back propagation neural network) were developed.

Similarly, for the more complex clinkerization process, multiple linear regression, three types of feed forward neural network models (back propagation, radial basis function and regression), two types of fuzzy inference models (Mamdani and Takagi-Sugeno), and one hybrid model (PCA plus back propagation neural network) were developed. The performances of all developed models were analyzed by evaluating the model outputs for the unknown (validation) data.

The performance analysis was carried out in terms of six different statistical model performance indices. The best model in each of the two processes was decided as the one that gives the best desirable values in all the six statistical parameters. The best chosen model was implemented in the SIMULINK platform by means of interconnected block diagram structure to assess their continuous quality monitoring capability. For the grinding process, a simple model predictive control scheme has been developed to illustrate the potential of the model for closed loop control of cement fineness within the desired range.

6.2 Significant Observations and Findings

In the following sections the significant findings of this research work are presented. The findings are classified into three categories. In the first section, the general findings are presented which are followed by observations related to the grinding and clinkerization process in sections two and three respectively.

6.2.1 General Observations

1. Among the three univariate outlier detection techniques, the Hampel's method of outlier detection using the outlier resistant median and median absolute deviation instead of outlier sensitive mean and standard deviation produced better results than the three sigma edit rule and the box plot method.
2. In case of radial basis function neural network, with increase in the number of RBF neurons, accuracy of model fitting increases for the training data.
3. In case of regression and radial basis function neural network, with decrease in the spread parameter, the fitting to training data becomes more and more accurate.
4. In all types of neural network, fuzzy inference and support vector regression models, the optimum model parameters were decided by the method of cross validation. The optimum parameters are the ones that result in lower modeling error not only for the data used for model development (training data) but also for the unknown or validation data.
5. Among the statistical parameters for model evaluation, though RMSE and MAE both represent the average error between predicted and actual values, for any model, RMSE value is always higher than the MAE value.
6. During model evaluation, in a few cases, between two models, MAE value for one model was lower while the RMSE value for the other model was lower. In such cases the better model was chosen as the one producing lower MAE value since this has been reported in the statistical literature as a better performance parameter than the RMSE.

6.2.2 Soft Sensing of Cement Fineness

1. Two statistical techniques were applied for design of training set from the total data set. It was observed that the technique of Kennard-Stone maximal intra distance criterion performs better than the Duplex method.
2. Among statistical modeling techniques, the non-linear regression technique using the method of support vector machine performs better than the techniques of multiple linear regression, principal component regression and quadratic response surface methods.
3. Among the two types of support vector regression techniques studied, the technique of least square support vector regression performs better than the standard support vector regression technique.
4. In case of support vector regression models, the technique of rigorous grid search method results in better optimum values of SVR hyper-parameters than the values determined by using analytical formulae proposed in the literature.
5. Among the back propagation, radial basis function and regression neural network models designed for the grinding process, the back propagation model trained by resilient back propagation training algorithm produced the best result.
6. In the grinding process where only three input variables are involved, dimensionality reduction is not advisable. The performance of principal component regression and PCA-BPNN model which were formed using two latent variables as inputs, performed worse than multiple linear regression model and BPNN model respectively.
7. Overall, the hybrid model using adaptive neuro-fuzzy inference technique and the back propagation neural network produced better performance than the other

models. Any of these models can be implemented in the grinding process for online monitoring of cement fineness.

8. The predictive capability of the model can also be exploited for closed loop control of particle size in the grinding process.

6.2.3 Soft Sensing of Clinker Quality

1. Among the five multivariate outlier detection techniques which were applied to the kiln data, the performances of the three robust methods were better than the performances of the two classical methods. Among the three robust methods, the smallest half volume (SHV) method is very conservative and detects too many observations as outliers which may result in loss of important informations contained in the data. The performance of the closest distance to center (CDC) method was found to be the best because the data set resulting from application of CDC method has better or at par statistical characteristics as those resulting from other methods but with less number of observations detected as outliers.
2. Among the neural network models developed for the rotary kiln using the data set 1 (which was obtained by using univariate outlier detection and random data selection) the radial basis function neural network showed better performance than the back propagation and regression neural network models.
3. However, models developed using data set 2 (Obtained by using multivariate outlier detection and statistical method of training data selection), showed better performance with less complexity, than the models developed using data set 1.
4. Because of nine input variables and the multivariate nature of the input data, it is strongly recommended to use multivariate outlier detection technique instead of univariate outlier detection technique in the initial data preprocessing step.

5. Model performance improves when along with kiln operating variables, input material quality values are also included as input parameters.
6. Application of principal component analysis for reduction of number of input variables makes sense for kiln modeling because it was observed that the BPNN model developed from the nine actual input variables and the PCA-BPNN model developed from five latent variables, have almost similar performance.
7. Among the three types of neural network models and two types of fuzzy inference models, the Takagi-Sugeno fuzzy inference model outperformed all other models.

6.3 Major Contributions

1. A comparison of three univariate outlier detection techniques is performed on two sets of real industrial data, one set belongs to the clinker grinding process and the other set belongs to the pyro process (clinkerization process) in cement manufacturing.
2. Performance of five multivariate outlier detection techniques is compared on a real industrial data set for the pyro-process in cement manufacturing.
3. Two statistical techniques for training set design are compared.
4. Performances of three types of feed forward neural network models, and two types of fuzzy inference models are compared for the two industrial processes.
5. Data-driven soft sensors were designed for continuous online monitoring of cement fineness in the clinker grinding process using vertical roller mill and eight different clinker quality parameters in the pyroprocess using the rotary cement kiln for clinker production.
6. A model predictive control strategy has been developed using the mill classifier speed as the manipulated variable for maintaining the cement fineness within the desired value.

6.4 Future Scope of Research

This section comprises of two parts. In the first part the scope in the cement manufacturing process is presented followed by scope in the other processes.

6.4.1 Cement Manufacturing Process

1. It can be tried to improve the model performance further especially for the cement kiln. Probably along with instantaneous input values, time delayed input values can be included for model development. This time delay can be taken as the material residence time inside the kiln.
2. By including delayed inputs, dynamic soft sensor can be developed which can be utilized for online control of clinker quality.
3. Further, the hybrid modeling technique of adaptive neuro-fuzzy inference technique can be explored as a soft sensor model for the rotary cement kiln.

6.4.2 Other Industrial Processes

The different industrial processes for which soft sensors have been reported are presented in Chapter 2. The bulk of the soft sensors are reported for the petroleum refinery, polymer, fermentation and bioprocess industries. There are still many process industries for which application of soft sensors have not yet been adequately explored. A few such processes are presented below

1. In pulp and paper industry, soft sensors have been reported for prediction of Kappa number of pulp coming from digester. The final product i.e. paper is rejected based on the quality parameter of paper GSM (grams per square meter) and brightness. Research can be carried out to develop soft sensor for prediction of these quality for paper coming out of paper machine.
2. The final chemical composition of fertilizer (e.g. moisture, nitrogen etc) produced in the fertilizer plant.

3. In emission quality monitoring, while significant work has been done for NO_x monitoring, very few researches have been reported for SO_x monitoring. Presence of SO_x is likely to be significant in the stack gas when liquid petroleum products or petroleum coke are used as fuel.

Therefore, the research on soft sensor development for continuous online monitoring of quality variables can be extended to the process industries where, this has largely been unexplored so far.

References

Al-Shayji, K. A., & Liu, Y. A. (2002). Predictive modeling of large-scale commercial water desalination plants: data-based neural network and model-based process simulation. *Industrial & Engineering Chemistry Research*, *41*(25), 6460-6474.

Arauzo-Bravo, M. J., Cano-Izquierdo, J. M., Gomez-Sanchez, E., López-Nieto, M. J., Dimitriadis, Y. A., López-Coronado, J. (2004). Automatization of a penicillin production process with soft sensors and an adaptive controller based on neuro fuzzy systems. *Control Engineering Practice*, *12*(9), 1073-1090.

Atkinson, A. C. (1994). Fast very robust methods for the detection of multiple outliers. *Journal of the American Statistical Association*, *89*(428), 1329-1339.

Azadeh, A., Neshat, N., Kazemi, A., & Saberi, M. (2012). Predictive control of drying process using an adaptive neuro-fuzzy and partial least squares approach. *The International Journal of Advanced Manufacturing Technology*, *58*(5-8), 585-596.

Bequette, B. W. (2003). *Process control: modeling, design, and simulation*. Prentice Hall India, New Delhi.

Bhartiya, S., & Whiteley, J. R. (2001). Development of inferential measurements using neural networks. *ISA Transactions*, *40*(4), 307-323.

Billings, S. A., & Zheng, G. L. (1995). Radial basis function network configuration using genetic algorithms. *Neural Networks*, *8*(6), 877-890.

Binici, H., Aksogan, O., Cagatay, I. H., Tokyay, M., & Emsen, E. (2007). The effect of particle size distribution on the properties of blended cements incorporating GGBFS and natural pozzolan (NP). *Powder Technology*, *177*(3), 140-147.

Bogaerts, P., & Wouwer, A. V. (2003). Software sensors for bioprocesses. *ISA Transactions*, *42*(4), 547-558.

Bogaerts, P., & Wouwer, A. V. (2004). Parameter identification for state estimation—application to bioprocess software sensors. *Chemical Engineering Science*, *59*(12), 2465-2476.

Boulvin, M., Vande Wouwer, A., Lepore, R., Renotte, C., & Remy, M. (2003). Modeling and control of cement grinding processes. *Control Systems Technology, IEEE Transactions on*, *11*(5), 715-725.

Casali, A., Gonzalez, G., Torres, F., Vallebuona, G., Castelli, L., & Gimenez, P. (1998). Particle size distribution soft-sensor for a grinding circuit. *Powder Technology*, *99*(1), 15-21.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, *46*(1-3), 131-159.

Chen, S., Cowan, C. F., & Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis function networks. *Neural Networks, IEEE Transactions on*, *2*(2), 302-309.

Chen, F. Z., & Wang, X. Z. (1998). Software sensor design using Bayesian automatic classification and back-propagation neural networks. *Industrial & Engineering Chemistry Research*, *37*(10), 3985-3991.

Cherkassky, V., & Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, *17*(1), 113-126.

Chiang, L. H., Pell, R. J., & Seasholtz, M. B. (2003). Exploring process data with the use of robust outlier detection algorithms. *Journal of Process Control*, *13*(5), 437-449.

Cigizoglu, H. K., & Alp, M. (2006). Generalized regression neural network in modelling river sediment yield. *Advances in Engineering Software*, *37*(2), 63-68.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273-297.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Dai, X., Wang, W., Ding, Y., & Sun, Z. (2006). "Assumed inherent sensor" inversion based ANN dynamic soft-sensing method and its application in erythromycin fermentation process. *Computers & Chemical Engineering*, *30*(8), 1203-1225.

Dam, M., & Saraf, D. N. (2006). Design of neural networks using genetic algorithm for on-line property estimation of crude fractionator products. *Computers & Chemical Engineering*, *30*(4), 722-729.

Daszykowski, M., Serneels, S., Kaczmarek, K., Van Espen, P., Croux, C., & Walczak, B. (2007). TOMCAT: A MATLAB toolbox for multivariate calibration techniques. *Chemometrics and Intelligent Laboratory Systems*, *85*(2), 269-277.

Daszykowski, M., Kaczmarek, K., Vander Heyden, Y., & Walczak, B. (2007). Robust statistics in data analysis—a review: basic concepts. *Chemometrics and Intelligent Laboratory Systems*, *85*(2), 203-219.

De Groot, P. J., Postma, G. J., Melssen, W. J., & Buydens, L. M. C. (1999). Selecting a representative training set for the classification of demolition waste using remote NIR sensing. *Analytica Chimica Acta*, *392*(1), 67-75.

Dehghan, S., Sattari, G. H., Chehreh Chelgani, S., & Aliabadi, M. A. (2010). Prediction of uniaxial compressive strength and modulus of elasticity for Travertine samples using regression and artificial neural networks. *Mining Science and Technology (China)*, *20*(1), 41-46.

Desai, K., Badhe, Y., Tambe, S. S., & Kulkarni, B. D. (2006). Soft-sensor development for fed-batch bioreactors using support vector regression. *Biochemical Engineering Journal*, *27*(3), 225-239.

Dufour, P., Bhartiya, S., Dhurjati, P. S., & Doyle Iii, F. J. (2005). Neural network-based software sensor: training set design and application to a continuous pulp digester. *Control Engineering Practice*, *13*(2), 135-143.

Egan, W. J., & Morgan, S. L. (1998). Outlier detection in multivariate analytical chemical data. *Analytical Chemistry*, *70*(11), 2372-2379.

Elsharkwy, A. M., & Gharbi, R. B. C. (2000). Comparing classical and neural regression techniques in modeling crude oil viscosity. *Advances in Engineering Software*, *32*(3), 215-224.

Facco, P., Doplicher, F., Bezzo, F., & Barolo, M. (2009). Moving average PLS soft sensor for online product quality estimation in an industrial batch polymerization process. *Journal of Process Control*, *19*(3), 520-529.

Feudale, R. N., Woody, N. A., Tan, H., Myles, A. J., Brown, S. D., & Ferré, J. (2002). Transfer of multivariate calibration models: a review. *Chemometrics and Intelligent Laboratory Systems*, *64*(2), 181-192.

Filzmoser, P., Maronna, R., & Werner, M. (2008). Outlier identification in high dimensions. *Computational Statistics & Data Analysis*, *52*(3), 1694-1711.

Fortuna, L., Rizzo, A., Sinatra, M., & Xibilia, M. G. (2003). Soft analyzers for a sulfur recovery unit. *Control Engineering Practice*, *11*(12), 1491-1500.

Fortuna, L., Graziani, S., & Xibilia, M. G. (2005). Soft sensors for product quality monitoring in debutanizer distillation columns. *Control Engineering Practice*, *13*(4), 499-508.

Fortuna, L., Graziani, S., Rizzo, A., & Xibilia, M. G. (2007). *Soft sensors for monitoring and control of industrial processes*. Springer Science & Business Media.

Galicia, H. J., He, Q. P., & Wang, J. (2011). A reduced order soft sensor approach and its application to a continuous digester. *Journal of Process Control*, *21*(4), 489-500.

Ge, Z., & Song, Z. (2010). Nonlinear soft sensor development based on relevance vector machine. *Industrial & Engineering Chemistry Research*, *49*(18), 8685-8693.

Ge, Z., & Song, Z. (2014). Ensemble independent component regression models and soft sensing application. *Chemometrics and Intelligent Laboratory Systems*, *130*, 115-122.

Ghodsi, A., & Schuurmans, D. (2003). Automatic basis selection techniques for RBF networks. *Neural Networks*, *16*(5), 809-816.

Gnanadesikan, R., & Kettenring, J. R. (1972). Robust estimates, residuals, and outlier detection with multiresponse data. *Biometrics*, 81-124.

Gokceoglu, C., & Zorlu, K. (2004). A fuzzy model to predict the uniaxial compressive strength and the modulus of elasticity of a problematic rock. *Engineering Applications of Artificial Intelligence*, *17*(1), 61-72.

Goh, A. T. (1999). Soil laboratory data interpretation using generalized regression neural network. *Civil Engineering Systems*, *16*(3), 175-195.

Gonzaga, J. C. B., Meleiro, L. A. C., Kiang, C., & Maciel Filho, R. (2009). ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process. *Computers & Chemical Engineering*, *33*(1), 43-49.

Gunn, S. R. (1998). Support vector machines for classification and regression. *ISIS Technical Report*, *14*.

Gunn, S. R. (1998). MATLAB support vector machine toolbox. *Image Speech and Intelligent Systems Research Group, University of Southampton, UK*. Available online at: http://www.isis.ecs.soton.ac.uk/resources/svminfo/.

Gunther, J. C., Conner, J. S., & Seborg, D. E. (2009). Process monitoring and quality variable prediction utilizing PLS in industrial fed-batch cell culture. *Journal of Process Control*, *19*(5), 914-921.

Gurumoorthy, A., & Kosanovich, K. A. (1998). Improving the prediction capability of radial basis function networks. *Industrial & Engineering Chemistry Research*, *37*(10), 3956-3970.

Hadi, A. S. (1992). Identifying multiple outliers in multivariate data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54, 761-771.

Herbst, J. A., & Pate, W. T. (1999). Object components for comminution system soft sensor design. *Powder Technology*, *105*(1), 424-429.

Hewlett, P. (2003). *Lea's Chemistry of Cement And Concrete*. Butterworth-Heinemann.

Hoaglin, D. C., & Welsch, R. E. (1978). The hat matrix in regression and ANOVA. *The American Statistician*, *32*(1), 17-22.

Hodge, V. J., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, *22*(2), 85-126.

Hoo, K. A., Tvarlapati, K. J., Piovoso, M. J., & Hajare, R. (2002). A method of robust multivariate outlier replacement. *Computers & Chemical Engineering*, *26*(1), 17-39.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification. Technical report. Taipei:National Taiwan University, Available online at https://www.cs.sfu.ca/people/Faculty/teaching/726/spring11/svmguide.pdf.

Huang, C. L., & Wang, C. J. (2006). A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications*, *31*(2), 231-240.

Hvala, N., Strmčnik, S., Šel, D., Milanič, S., & Banko, B. (2005). Influence of model validation on proper selection of process models—an industrial case study. *Computers & Chemical Engineering*, *29*(7), 1507-1522.

Ikonen, E., Najim, K., & Kortela, U. (2000). Neuro-fuzzy modelling of power plant flue-gas emissions. *Engineering Applications of Artificial Intelligence*, *13*(6), 705-717.

Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *Systems, Man and Cybernetics, IEEE Transactions on*, *23*(3), 665-685.

Jang, J. S. R., & Sun, C. T. (1997). *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. Prentice-Hall, Inc..

Jolliffe, I. (2005). *Principal component analysis*. John Wiley & Sons, Ltd.

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

Kaneko, H., Arakawa, M., & Funatsu, K. (2009). Development of a new soft sensor method using independent component analysis and partial least squares. *AIChE journal*, *55*(1), 87-98.

Kaneko, H., & Funatsu, K. (2013). Nonlinear regression method with variable region selection and application to soft sensors. *Chemometrics and Intelligent Laboratory Systems*, *121*, 26-32.

Kano, M., & Nakagawa, Y. (2008). Data-based process monitoring, process control, and quality improvement: Recent developments and applications in steel industry. *Computers & Chemical Engineering*, *32*(1), 12-24.

Kennard, R. W., & Stone, L. A. (1969). Computer aided design of experiments. *Technometrics*, *11*(1), 137-148.

Kim, M., Lee, Y. H., Han, I. S., & Han, C. (2005). Clustering-based hybrid soft sensor for an industrial polypropylene process with grade changeover operation. *Industrial & Engineering Chemistry Research*, *44*(2), 334-342.

Kim, B., Kwon, M., & Kwon, S. H. (2009). Modeling of plasma process data using a multi-parameterized generalized regression neural network. *Microelectronic Engineering*, *86*(1), 63-67.

Kisi, O., & Kerem Cigizoglu, H. (2007). Comparison of different ANN techniques in river flow prediction. *Civil Engineering and Environmental Systems*, *24*(3), 211-231.

Ko, Y. D., & Shang, H. (2011). A neural network-based soft sensor for particle size distribution using image analysis. *Powder Technology*, *212*(2), 359-366.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464-1480.

Kresta, J. V., Marlin, T. E., & MacGregor, J. F. (1994). Development of inferential process models using PLS. *Computers & Chemical Engineering*, *18*(7), 597-611.

Lakshminarayan, K., Harp, S. A., & Samad, T. (1999). Imputation of missing data in industrial databases. *Applied Intelligence*, *11*(3), 259-275.

Lee, D. S., & Park, J. M. (1999). Neural network modeling for on-line estimation of nutrient dynamics in a sequentially-operated batch reactor. *Journal of Biotechnology*, *75*(2), 229-239.

Lee, H. W., Lee, M. W., & Park, J. M. (2007). Robust adaptive partial least squares modeling of a full-scale industrial wastewater treatment process. *Industrial & Engineering Chemistry Research*, *46*(3), 955-964.

Li, C., Ye, H., & Wang, G. (2004, October). Nonlinear time series modeling and prediction using RBF network with improved clustering algorithm. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on* (Vol. 4, pp. 3513-3518). IEEE.

Li, J., & Liu, X. (2011). Melt index prediction by RBF neural network optimized with an MPSO-SA hybrid algorithm. *Neurocomputing*, *74*(5), 735-740.

Li, S., & Tan, M. (2010). Tuning SVM parameters by using a hybrid CLPSO–BFGS algorithm. *Neurocomputing*, *73*(10), 2089-2096.

Lin, B., Recke, B., Knudsen, J. K., & Jørgensen, S. B. (2007). A systematic approach for soft sensor development. *Computers & Chemical Engineering*, *31*(5), 419-425.

Lin, S. W., Ying, K. C., Chen, S. C., & Lee, Z. J. (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications*, *35*(4), 1817-1824.

Lin, S. W., Lee, Z. J., Chen, S. C., & Tseng, T. Y. (2008). Parameter determination of support vector machine and feature selection using simulated annealing approach. *Applied Soft Computing*, *8*(4), 1505-1512.

Lin, B., & Jørgensen, S. B. (2011). Soft sensor design by multivariate fusion of image features and process measurements. *Journal of Process Control*, *21*(4), 547-553.

Linko, S., Luopa, J., & Zhu, Y. H. (1997). Neural networks as 'software sensors' in enzyme production. *Journal of Biotechnology*, *52*(3), 257-266.

Liu, J. (2007). On-line soft sensor for polyethylene process with multiple production grades. *Control Engineering Practice*, *15*(7), 769-778.

Liu, H., Shah, S., & Jiang, W. (2004). On-line outlier detection and data cleaning. *Computers & Chemical Engineering*, *28*(9), 1635-1647.

Liu, G., Zhou, D., Xu, H., & Mei, C. (2010). Model optimization of SVM for a fermentation soft sensor. *Expert systems with Applications*, *37*(4), 2708-2713.

Liu, Y., Gao, Z., & Chen, J. (2013). Development of soft-sensors for online quality prediction of sequential-reactor-multi-grade industrial processes. *Chemical Engineering Science*, *102*, 602-612.

Ljung, L. (1998). *System identification* (pp. 163-173). Birkhäuser Boston.

Lu, S., Wang, J., Zou, Y., Pan, Y., Chu, J., & Sun, W. (2009, January). Secondary Variables Selection and Data Pretreatment in Soft-Sensing for the Farmland Water Potential. In *Intelligent Information Technology and Security Informatics, 2009. IITSI'09. Second International Symposium on* (pp. 110-114). IEEE.

Luo, J. X., & Shao, H. H. (2006). Developing soft sensors using hybrid soft computing methodology: a neurofuzzy system based on rough set theory and genetic algorithms. *Soft Computing*, *10*(1), 54-60.

Ma, M. D., Ko, J. W., Wang, S. J., Wu, M. F., Jang, S. S., Shieh, S. S., & Wong, D. S. H. (2009). Development of adaptive soft sensor based on statistical identification of key variables. *Control Engineering Practice*, *17*(9), 1026-1034.

Mannina, G., & Di Bella, G. (2012). Comparing two start-up strategies for MBRs: experimental study and mathematical modelling. *Biochemical Engineering Journal*, *68*, 91-103.

Marengo, E., Bobba, M., Robotti, E., & Liparota, M. C. (2006). Modeling of the polluting emissions from a cement production plant by partial least-squares, principal component regression, and artificial neural networks. *Environmental Science & Technology*, *40*(1), 272-280.

Marinaro, M., & Scarpetta, S. (2000). On-line learning in RBF neural networks: a stochastic approach. *Neural Networks*, *13*(7), 719-729.

Marjanovic, O., Lennox, B., Sandoz, D., Smith, K., & Crofts, M. (2006). Real-time monitoring of an industrial batch process. *Computers & Chemical Engineering*, *30*(10), 1476-1481.

Martin, G., & McGarel, S. (2001). Nonlinear mill control. *ISA Transactions*, *40*(4), 369-379.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 415-446.

Mitra, K., & Ghivari, M. (2006). Modeling of an industrial wet grinding operation using data-driven techniques. *Computers & Chemical Engineering*, *30*(3), 508-520.

Mohammadi, A. H., & Richon, D. (2007). Use of artificial neural networks for estimating water content of natural gases. *Industrial & Engineering Chemistry Research*, *46*(4), 1431-1438.

Morad, K., Young, B. R., & Svrcek, W. Y. (2005). Rectification of plant measurements using a statistical framework. *Computers & Chemical Engineering*, *29*(5), 919-940.

Moriasi, D. N., Arnold, J. G., Van Liew, M. W., Bingner, R. L., Harmel, R. D., & Veith, T. L. (2007). Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Trans. ASABE*, *50*(3), 885-900.

Mu, S., Zeng, Y., Liu, R., Wu, P., Su, H., & Chu, J. (2006). Online dual updating with recursive PLS model and its application in predicting crystal size of purified terephthalic acid (PTA) process. *Journal of Process Control*, *16*(6), 557-566.

Nash, J., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I—A discussion of principles. *Journal of Hydrology*, *10*(3), 282-290.

Ochoa, S., Yoo, A., Repke, J. U., Wozny, G., & Yang, D. R. (2007). Modeling and parameter identification of the simultaneous saccharification-fermentation process for ethanol production. *Biotechnology Progress*, *23*(6), 1454-1462.

Ohshima, M., & Tanigaki, M. (2000). Quality control of polymer production processes. *Journal of Process Control*, *10*(2), 135-148.

Pai, T. Y., Wan, T. J., Hsu, S. T., Chang, T. C., Tsai, Y. P., Lin, C. Y., & Yu, L. F. (2009). Using fuzzy inference system to improve neural network for predicting hospital wastewater treatment plant effluent. *Computers & Chemical Engineering*, *33*(7), 1272-1278.

Palani, S., Liong, S. Y., & Tkalich, P. (2008). An ANN application for water quality forecasting. *Marine Pollution Bulletin*, *56*(9), 1586-1597.

Papadokonstantakis, S., Machefer, S., Schnitzlein, K., & Lygeros, A. I. (2005). Variable selection and data pre-processing in NN modelling of complex chemical processes. *Computers & Chemical Engineering*, *29*(7), 1647-1659.

Park, S., & Han, C. (2000). A nonlinear soft sensor based on multivariate smoothing procedure for quality estimation in distillation columns. *Computers & Chemical Engineering*, *24*(2), 871-877.

Pearson, R. K. (2002). Outliers in process modeling and identification. *Control Systems Technology, IEEE Transactions on*, *10*(1), 55-63.

Peisheng, L., Youhui, X., Dunxi, Y., & Xuexin, S. (2005). Prediction of grindability with multivariable regression and neural network in Chinese coal. *Fuel*, *84*(18), 2384-2388.

Pelckmans, K., Suykens, J. A., Van Gestel, T., De Brabanter, J., Lukas, L., Hamers, B., & Vandewalle, J. (2002). LS-SVMlab: a matlab/c toolbox for least squares support vector machines. *Tutorial. KULeuven-ESAT. Leuven, Belgium*.

Perendeci, A., Arslan, S., Çelebi, S. S., & Tanyolaç, A. (2008). Prediction of effluent quality of an anaerobic treatment plant under unsteady state through ANFIS modeling with on-line input variables. *Chemical Engineering Journal*, *145*(1), 78-85.

Qiao, J., Fang, Z., & Chai, T. (2010, March). LS-SVR-Based Soft Sensor Model for Cement Clinker Calcination Process. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on* (Vol. 2, pp. 591-594). IEEE.

Qin, S. J., Yue, H., & Dunia, R. (1997). Self-validating inferential sensors with application to air emission monitoring. *Industrial & Engineering Chemistry Research*, *36*(5), 1675-1685.

Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, *11*(7), 733-764.

Qin, Y., Zhang, S., Zhu, X., Zhang, J., & Zhang, C. (2007). Semi-parametric optimization for missing data imputation. *Applied Intelligence*, *27*(1), 79-88.

Radhakrishnan, V. R., & Mohamed, A. R. (2000). Neural networks for the identification and control of blast furnace hot metal quality. *Journal of Process Control*, *10*(6), 509-524.

Rajer-Kanduč, K., Zupan, J., & Majcen, N. (2003). Separation of data on the training and test set for modelling: a case study for modelling of five colour properties of a white pigment. *Chemometrics and Intelligent Laboratory Systems*, *65*(2), 221-229.

Rallo, R., Ferre-Gine, J., Arenas, A., & Giralt, F. (2002). Neural virtual sensor for the inferential prediction of product quality from process variables. *Computers & Chemical Engineering*, *26*(12), 1735-1754.

Riani, M., Atkinson, A. C., & Cerioli, A. (2009). Finding an unknown number of multivariate outliers. *Journal of the Royal Statistical Society: series B (statistical methodology)*, *71*(2), 447-466.

Rousseeuw, P. J., & Leroy, A. M. (1987). *Robust regression and outlier detection*. John Wiley, New York.

Rousseeuw, P. J., & Van Zomeren, B. C. (1990). Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, *85*(411), 633-639.

Roy, N. K., Potter, W. D., & Landau, D. P. (2006). Polymer property prediction and optimization using neural networks. *Neural Networks, IEEE Transactions on*, *17*(4), 1001-1014.

Runkler, T. A., Gerstorfer, E., Schlang, M., Jünnemann, E., & Hollatz, J. (2003). Modelling and optimisation of a refining process for fibre board production. *Control Engineering Practice*, *11*(11), 1229-1241.

Sadeghian, M., & Fatehi, A. (2011). Identification, prediction and detection of the process fault in a cement rotary kiln by locally linear neuro-fuzzy technique. *Journal of Process Control*, *21*(2), 302-308.

Sahasrabudhe, R., Sistu, P., Sardar, G. A. U. T. A. M., & Gopinath, R. A. V. I. (2006). Control and optimization in cement plants. *Control Systems, IEEE*, *26*(6), 56-63.

Samanta, B. (2010). Radial basis function network for ore grade estimation. *Natural Resources Research*, *19*(2), 91-102.

Samarasinghe, S. (2006). *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. CRC Press.

Sarimveis, H., Alexandridis, A., Tsekouras, G., & Bafas, G. (2002). A fast and efficient algorithm for training radial basis function neural networks based on a fuzzy partition of the input space. *Industrial & Engineering Chemistry Research*, *41*(4), 751-759.

Sbarbaro, D., Ascencio, P., Espinoza, P., Mujica, F., & Cortes, G. (2008). Adaptive soft-sensors for on-line particle size estimation in wet grinding circuits. *Control Engineering Practice*, *16*(2), 171-178.

Shakil, M., Elshafei, M., Habib, M. A., & Maleki, F. A. (2009). Soft sensor for NO_x and O_2 using dynamic neural networks. *Computers & Electrical Engineering*, *35*(4), 578-586.

Sharifi, A., Aliyari Shoorehdeli, M., & Teshnehlab, M. (2012). Identification of cement rotary kiln using hierarchical wavelet fuzzy inference system. *Journal of the Franklin Institute*, *349*(1), 162-183.

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

Sharmin, R., Sundararaj, U., Shah, S., Vande Griend, L., & Sun, Y. J. (2006). Inferential sensors for estimation of polymer quality parameters: Industrial application of a PLS-based soft sensor for a LDPE plant. *Chemical Engineering Science*, *61*(19), 6372-6384.

Shi, J., & Liu, X. G. (2006). Product quality prediction by a neural soft-sensor based on MSA and PCA. *International Journal of Automation and Computing*, *3*(1), 17-22.

Shridhar, R., & Cooper, D. J. (1997). A tuning strategy for unconstrained SISO model predictive control. *Industrial & Engineering Chemistry Research*, *36*(3), 729-746.

Simmons, M., Gorby, L., & Terembula, J. (2005, May). Operational experience from the United States' first vertical roller mill for cement grinding. In *Cement Industry Technical Conference, 2005. Conference Record* (pp. 241-249). IEEE.

Singh, K. P., & Gupta, S. (2012). Artificial intelligence based modeling for predicting the disinfection by-products in water. *Chemometrics and Intelligent Laboratory Systems*, *114*, 122-131.

Snee, R. D. (1977). Validation of regression models: methods and examples. *Technometrics*, *19*(4), 415-428.

Sorrentino, F. (2011). Chemistry and engineering of the production process: State of the art. *Cement and Concrete Research*, *41*(7), 616-623.

Sotomayor, O. A., Park, S. W., & Garcia, C. (2002). Software sensor for on-line estimation of the microbial activity in activated sludge systems. *ISA Transactions*, *41*(2), 127-143.

Specht, D. F. (1991). A general regression neural network. *Neural Networks, IEEE Transactions on*, *2*(6), 568-576.

Stadler, K. S., Poland, J., & Gallestey, E. (2011). Model predictive control of a rotary cement kiln. *Control Engineering Practice*, *19*(1), 1-9.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*(3), 293-300.

Tian, H., Mao, Z., Wang, S., & Li, K. (2006, June). Application of genetic algorithm combined with BP neural network in soft sensor of molten steel temperature. In *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on* (Vol. 2, pp. 7742-7745). IEEE.

Topalov, A. V., & Kaynak, O. (2004). Neural network modeling and control of cement mills using a variable structure systems theory based on-line learning mechanism. *Journal of Process Control*, *14*(5), 581-589.

Tronci, S., Baratti, R., & Servida, A. (2002). Monitoring pollutant emissions in a 4.8 MW power plant through neural network. *Neurocomputing*, *43*(1), 3-15.

Vapnik, V., Golowich, S. E., & Smola, A. (1997). Support vector method for function approximation, regression estimation, and signal processing. *Advances in neural Information Processing Systems*, 281-287.

Vapnik, V. N., & Vapnik, V. (1998). *Statistical learning theory* (Vol. 2). New York: Wiley.

Verboven, S., & Hubert, M. (2005). LIBRA: a MATLAB library for robust analysis. *Chemometrics and Intelligent Laboratory Systems*, *75*(2), 127-136.

Vuk, T., Tinta, V., Gabrovšek, R., & Kaučič, V. (2001). The effects of limestone addition, clinker type and fineness on properties of Portland cement. *Cement and concrete Research*, *31*(1), 135-139.

Wang, H., Song, Z., & Wang, H. (2002). Statistical process monitoring using improved PCA with optimized sensor locations. *Journal of Process Control*, *12*(6), 735-744.

Wang, W., Xu, Z., Lu, W., & Zhang, X. (2003). Determination of the spread parameter in the Gaussian kernel for classification and regression. *Neurocomputing*, *55*(3), 643-663.

Wang, D., Liu, J., & Srinivasan, R. (2010). Data-driven soft sensor approach for quality prediction in a refining process. *Industrial Informatics, IEEE Transactions on*, *6*(1), 11-17.

Warne, K., Prasad, G., Rezvani, S., & Maguire, L. (2004). Statistical and computational intelligence techniques for inferential model development: a comparative evaluation and a novel proposition for fusion. *Engineering Applications of Artificial Intelligence*, *17*(8), 871-885.

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, *30*(1), 79.

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, *2*(1), 37-52.

Woodruff, D. L., & Rocke, D. M. (1993). Heuristic search algorithms for the minimum volume ellipsoid. *Journal of Computational and Graphical Statistics*, *2*(1), 69-95.

Wu, X., & Yuan, M. (2009, June). Soft-sensor modeling of product particle size in ball milling circuits based on fuzzy neural networks with particle swarm optimization. In *Information and Automation, 2009. ICIA'09. International Conference on* (pp. 1458-1461). IEEE.

Yagiz, S., & Gokceoglu, C. (2010). Application of fuzzy inference system and nonlinear regression models for predicting rock brittleness. *Expert Systems with Applications*, *37*(3), 2265-2272.

Yan, W., Shao, H., & Wang, X. (2004). Soft sensing modeling based on support vector machine and Bayesian model selection. *Computers & Chemical Engineering*, *28*(8), 1489-1498.

Yan, X. (2008). Modified nonlinear generalized ridge regression and its application to develop naphtha cut point soft sensor. *Computers & Chemical Engineering*, *32*(3), 608-621.

Yuan, S. F., & Chu, F. L. (2007). Fault diagnostics based on particle swarm optimisation and support vector machines. *Mechanical Systems and Signal Processing*, *21*(4), 1787-1798.

Zamprogna, E., Barolo, M., & Seborg, D. E. (2004). Estimating product composition profiles in batch distillation via partial least squares regression. *Control Engineering Practice*, *12*(7), 917-929.

Zamprogna, E., Barolo, M., & Seborg, D. E. (2005). Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis. *Journal of Process Control*, *15*(1), 39-52.

Zhao, Y. H. (2003, November). A soft sensor based on nonlinear principal component analysis. In *Machine Learning and Cybernetics, 2003 International Conference on* (Vol. 2, pp. 707-710). IEEE.

Zheng, L., & Yu, M. (2008, May). Monitoring NO_x emissions from coal fired boilers using generalized regression neural network. In *Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on* (pp. 1916-1919). IEEE.

Zeng, J. S., & Gao, C. H. (2009). Improvement of identification of blast furnace ironmaking process by outlier detection and missing value imputation. *Journal of Process Control*, *19*(9), 1519-1528.

Zheng, L., Zhou, H., Wang, C., & Cen, K. (2008). Combining support vector regression and ant colony optimization to reduce NO_x emissions in coal-fired utility boilers. *Energy & Fuels*, *22*(2), 1034-1040.

Zhong, L., & Du, H. (2012). Prediction of free lime content in cement clinker based on RBF neural network. *Journal of Wuhan University of Technology-Mater. Sci. Ed.*, *27*(1), 187-190.

Zhou, H., Pei Zhao, J., Gang Zheng, L., Lin Wang, C., & Fa Cen, K. (2012). Modeling NO_x emissions from coal-fired utility boilers using support vector regression with ant colony optimization. *Engineering Applications of Artificial Intelligence*, *25*(1), 147-158.

Appendix A

MATLAB Codes

1.1 Outlier detection by conventional Mahalanobis distance method

```
load kilndata;
input = [kilnfeed, rpm, intemp, coalfeed, current]; % Concatenation of the input variables
[m,n] = size(input);
average = mean(input);
for j = 1:n
   for i = 1:m
      X(i,j) = input(i,j) - average(j); % Mean centered input data matrix
   end
end
covariance = (X'*X)/(m-1); % Calculation of covariance matrix
maha = diag(X*inv(covariance)*X'); % Calculation of Mahalanobis distance
for i = 1:m
   maha1(i,1) = X(i,:)*inv(covariance)*X(i,:)';
end
outliermd = zeros(size(maha,1),1);
j = 1;
for i = 1:1144
   if maha(i)> 12.833 %Chi square cut off value
      outliermd(j,1) = i;
      j = j+1;
   end
end
outliermd(j:size(maha)) = []; % Vector of sample numbers detected as outliers
Xcorrect = input;
Xcorrect(outliermd,:) = []; % Corrected input matrix after the deletion of the rows detected
     as outliers
```

1.2 Outlier detection by hat matrix leverage value method

```
load kilndata;
input = [kilnfeed, rpm, intemp, coalfeed, current]; % Concatenation of the input variables
[m,n] = size(input);
average = mean(input);
for j = 1:n
   for i = 1:m
      X(i,j) = input(i,j) - average(j); % Mean centered input data matrix
   end
end
covariance = (X'*X)/(m-1); % Calculation of covariance matrix
maha = diag(X*inv(covariance)*X'); % Calculation of Mahalanobis distance
for i = 1:m
```

```
   maha1(i,1) = X(i,:)*inv(covariance)*X(i,:)';
end
H = X*inv(X'*X)*X';  % Hat Matrix
leverage = diag(H);  % Leverage Values
outlierlev = zeros(size(leverage,1),1);
j = 1;
for i = 1:1144
   if leverage(i)> 2*n/m
      outlierlev(j,1) = i;
      j = j+1;
   end
end
outlierlev(j:size(leverage)) = []; % Vector of sample numbers detected as outliers
Xcorrect = input;
Xcorrect(outlierlev,:) = []; % Corrected input matrix after the deletion of the rows detected
     as outliers
```

1.3 Outlier detection by smallest half volume method

```
load kilndata;
input = [kilnfeed, rpm, intemp, coalfeed, current]; % Concatenation of the input variables
clear kilnfeed rpm intemp coalfeed current;
distance = pdist2(input,input); % Calculation of distance matrix
sortdistance = sort(distance); % Each column of the distance matrix sorted in ascending
     order
[p,q] = size(distance);
modsortdistance = sortdistance;
modsortdistance(((p/2)+1):p,:)=[]; % Retaining only the first half elements of each column
     of the sorted matrix
summation = sum(modsortdistance); % Determine sum of each column
[minsum,index] = min(summation); % Get the minimum sum value and the index
testcolumn = distance(:,index); % retaining all distance values of indexth column
for i = 1:p/2
   [m,n] = min(testcolumn);
   sample(i,1) = n;  % Getting the minimum p/2 row numbers from the minimum distance
     column
   minval(i,1) = m;
   testcolumn(n) = max(testcolumn);
end
finaldata = input(sample,:); % Set of p/2 most similar observations
[m,n] = size(finaldata);
average = mean(finaldata); % Robust mean
for j = 1:n
   for i = 1:m
      P(i,j) = finaldata(i,j) - average(j); % Mean centered input data matrix
   end
end
covariance = (P'*P)/(m-1); % Calculation of robust covariance matrix
for j = 1:n
   for i = 1:p
      X(i,j) = input(i,j) - average(j); % Centering original data with robust mean
```

```
   end
end
maha = diag(X*inv(covariance)*X'); % Calculation of Mahalanobis distance
% for i = 1:m
%     maha1(i,1) = X(i,:)*inv(covariance)*X(i,:)';
% end
outliershv = zeros(size(maha,1),1);
j = 1;
for i = 1:p
   if maha(i)> 12.833
      outliershv(j,1) = i;
      j = j+1;
   end
end
outliershv(j:size(maha)) = []; % Vector of sample numbers detected as outliers
Xcorrect = input;
Xcorrect(outliershv,:) = []; % Corrected input matrix after the deletion of the rows
    detected as outliers
```

1.4 Outlier detection by closest distance to center method

```
load kilndata;
input = [kilnfeed, rpm, intemp, coalfeed, current]; % Concatenation of the input variables
clear kilnfeed rpm intemp coalfeed current;
[m,n] = size(input);
average = mean(input);
sigma = std(input);
for i = 1:m
   for j = 1:n
      X(i,j) = (input(i,j) - average(j))/sigma(j); % Scaling of input data matrix
   end
end
averageX = mean(X); % Center of the input data matrix
distance = pdist2(X,averageX); % Euclidean distance of each observation from the center
for i = 1:m/2
   [p,q] = min(distance);
   sample(i,1) = q;  % Getting the m/2 samples with minimum distance to the center
   minval(i,1) = p;
   distance(q) = max(distance);
end
finaldata = X(sample,:); % Set of m/2 most similar observations
[p,q] = size(finaldata);
average = mean(finaldata); % Robust mean
for j = 1:q
   for i = 1:p
      P(i,j) = finaldata(i,j) - average(j); % Mean centered input data matrix
   end
end
covariance = (P'*P)/(p-1); % Calculation of robust covariance matrix
for j = 1:n
   for i = 1:m
```

```
        Xm(i,j) = X(i,j) - average(j); % Centering original scaled data with robust mean
    end
end
maha = diag(Xm*inv(covariance)*Xm'); % Calculation of Mahalanobis distance
% for i = 1:m
%     maha1(i,1) = X(i,:)*inv(covariance)*X(i,:)';
% end
outliercdc = zeros(size(maha,1),1);
j = 1;
for i = 1:m
    if maha(i)> 12.833
        outliercdc(j,1) = i;
        j = j+1;
    end
end
outliercdc(j:size(maha)) = []; % Vector of sample numbers detected as outliers
Xcorrect = input;
Xcorrect(outliercdc,:) = []; % Corrected input matrix after the deletion of the rows
     detected as outliers
disp(['Original number of samples are: ',num2str(m)]);
disp(['Number of outliers detected are: ',num2str(size(outliercdc,1))]);
```

Appendix B

Fuzzy Rule Base

gf = Hot air flow rate through the mill
rpm = classifier speed
tph = clinker inflow rate
For the type of membership function used and the range of values for different linguistic variables, please refer to Chapter - 4.

1.1 Mamdani FIS Rule base for grinding process

1. If (gf is m) and (rpm is l) and (tph is m) then (output1 is l)
2. If (gf is m) and (rpm is l) and (tph is l) then (output1 is l)
3. If (gf is l) and (rpm is vs) and (tph is vl) then (output1 is s)
4. If (gf is vl) and (rpm is vs) and (tph is vl) then (output1 is s)
5. If (gf is vs) and (rpm is m) and (tph is s) then (output1 is s)
6. If (gf is vs) and (rpm is m) and (tph is s) then (output1 is m)
7. If (gf is l) and (rpm is m) and (tph is vs) then (output1 is l)
8. If (gf is l) and (rpm is m) and (tph is vs) then (output1 is vl)
9. If (gf is m) and (rpm is s) and (tph is l) then (output1 is s)
10. If (gf is s) and (rpm is vl) and (tph is s) then (output1 is vl)
11. If (gf is m) and (rpm is vl) and (tph is s) then (output1 is l)
12. If (gf is vl) and (rpm is s) and (tph is m) then (output1 is s)
13. If (gf is vl) and (rpm is s) and (tph is l) then (output1 is s)
14. If (gf is vl) and (rpm is m) and (tph is s) then (output1 is s)
15. If (gf is vl) and (rpm is m) and (tph is m) then (output1 is s)
16. If (gf is s) and (rpm is s) and (tph is s) then (output1 is vs)
17. If (gf is m) and (rpm is s) and (tph is s) then (output1 is vs)
18. If (gf is s) and (rpm is l) and (tph is l) then (output1 is l)
19. If (gf is s) and (rpm is l) and (tph is m) then (output1 is vl)
20. If (gf is l) and (rpm is l) and (tph is l) then (output1 is vl)
21. If (gf is vl) and (rpm is l) and (tph is l) then (output1 is l)
22. If (gf is l) and (rpm is vl) and (tph is l) then (output1 is vl)
23. If (gf is m) and (rpm is s) and (tph is vl) then (output1 is s)
24. If (gf is l) and (rpm is m) and (tph is s) then (output1 is m)
25. If (gf is l) and (rpm is l) and (tph is s) then (output1 is m)
26. If (gf is l) and (rpm is m) and (tph is m) then (output1 is m)
27. If (gf is l) and (rpm is l) and (tph is m) then (output1 is m)
28. If (gf is s) and (rpm is vl) and (tph is m) then (output1 is l)
29. If (gf is s) and (rpm is vl) and (tph is m) then (output1 is vl)
30. If (gf is s) and (rpm is m) and (tph is m) then (output1 is m)
31. If (gf is l) and (rpm is m) and (tph is l) then (output1 is vs)
32. If (gf is l) and (rpm is m) and (tph is l) then (output1 is s)
33. If (gf is s) and (rpm is m) and (tph is s) then (output1 is m)
34. If (gf is m) and (rpm is m) and (tph is s) then (output1 is m)
35. If (gf is l) and (rpm is vl) and (tph is m) then (output1 is l)

36. If (gf is l) and (rpm is vl) and (tph is m) then (output1 is vl)
37. If (gf is s) and (rpm is l) and (tph is s) then (output1 is vl)
38. If (gf is m) and (rpm is l) and (tph is s) then (output1 is vl)
39. If (gf is l) and (rpm is s) and (tph is l) then (output1 is vs)
40. If (gf is l) and (rpm is s) and (tph is l) then (output1 is s)
41. If (gf is l) and (rpm is s) and (tph is vl) then (output1 is vs)
42. If (gf is l) and (rpm is s) and (tph is vl) then (output1 is s)
43. If (gf is vl) and (rpm is s) and (tph is l) then (output1 is m)
44. If (gf is m) and (rpm is vl) and (tph is s) then (output1 is vl)
45. If (gf is m) and (rpm is vl) and (tph is m) then (output1 is vl)
46. If (gf is m) and (rpm is vs) and (tph is l) then (output1 is s)
47. If (gf is s) and (rpm is m) and (tph is s) then (output1 is s)
48. If (gf is s) and (rpm is vl) and (tph is l) then (output1 is vl)
49. If (gf is m) and (rpm is vl) and (tph is l) then (output1 is vl)
50. If (gf is l) and (rpm is l) and (tph is m) then (output1 is l)
51. If (gf is l) and (rpm is l) and (tph is l) then (output1 is l)
52. If (gf is l) and (rpm is s) and (tph is m) then (output1 is vl)
53. If (gf is l) and (rpm is s) and (tph is m) then (output1 is l)
54. If (gf is m) and (rpm is vs) and (tph is vl) then (output1 is s)
55. If (gf is l) and (rpm is m) and (tph is s) then (output1 is s)
56. If (gf is l) and (rpm is m) and (tph is m) then (output1 is s)
57. If (gf is l) and (rpm is vs) and (tph is vl) then (output1 is vs)
58. If (gf is m) and (rpm is vs) and (tph is vl) then (output1 is vs)
59. If (gf is m) and (rpm is l) and (tph is l) then (output1 is vl)
60. If (gf is s) and (rpm is l) and (tph is l) then (output1 is vl)
61. If (gf is s) and (rpm is m) and (tph is vs) then (output1 is s)
62. If (gf is s) and (rpm is m) and (tph is vs) then (output1 is m)
63. If (gf is m) and (rpm is s) and (tph is m) then (output1 is s)

1.2 Sugeno FIS Rule Base for grinding process

1. If (gf is m) and (rpm is l) and (tph is m) then (output1 is l)
2. If (gf is l) and (rpm is vs) and (tph is vl) then (output1 is lm)
3. If (gf is vs) and (rpm is m) and (tph is s) then (output1 is lm)
4. If (gf is l) and (rpm is m) and (tph is vs) then (output1 is vl)
5. If (gf is m) and (rpm is s) and (tph is l) then (output1 is s)
6. If (gf is m) and (rpm is vl) and (tph is s) then (output1 is el)
7. If (gf is vl) and (rpm is s) and (tph is m) then (output1 is s)
8. If (gf is vl) and (rpm is m) and (tph is s) then (output1 is vs)
9. If (gf is s) and (rpm is s) and (tph is s) then (output1 is es)
10. If (gf is s) and (rpm is l) and (tph is l) then (output1 is vl)
11. If (gf is l) and (rpm is l) and (tph is l) then (output1 is el)
12. If (gf is m) and (rpm is s) and (tph is vl) then (output1 is vs)
13. If (gf is l) and (rpm is m) and (tph is s) then (output1 is m)
14. If (gf is s) and (rpm is vl) and (tph is m) then (output1 is vl)
15. If (gf is s) and (rpm is m) and (tph is m) then (output1 is m)
16. If (gf is l) and (rpm is m) and (tph is l) then (output1 is es)
17. If (gf is s) and (rpm is m) and (tph is s) then (output1 is s)
18. If (gf is l) and (rpm is vl) and (tph is m) then (output1 is vl)
19. If (gf is s) and (rpm is l) and (tph is s) then (output1 is el)
20. If (gf is l) and (rpm is s) and (tph is l) then (output1 is vs)

21. If (gf is m) and (rpm is l) and (tph is l) then (output1 is vl)
22. If (gf is vs) and (rpm is m) and (tph is m) then (output1 is lm)
23. If (gf is vl) and (rpm is s) and (tph is l) then (output1 is lm)
24. If (gf is m) and (rpm is vl) and (tph is m) then (output1 is el)
25. If (gf is m) and (rpm is m) and (tph is s) then (output1 is m)
26. If (gf is m) and (rpm is vs) and (tph is l) then (output1 is s)
27. If (gf is l) and (rpm is l) and (tph is s) then (output1 is s)
28. If (gf is vl) and (rpm is l) and (tph is s) then (output1 is s)
29. If (gf is m) and (rpm is s) and (tph is m) then (output1 is s)
30. If (gf is l) and (rpm is vl) and (tph is l) then (output1 is el)
31. If (gf is m) and (rpm is l) and (tph is s) then (output1 is vl)
32. If (gf is s) and (rpm is s) and (tph is vl) then (output1 is vs)
33. If (gf is l) and (rpm is l) and (tph is m) then (output1 is vl)
34. If (gf is s) and (rpm is vl) and (tph is l) then (output1 is el)
35. If (gf is s) and (rpm is vl) and (tph is m) then (output1 is el)
36. If (gf is s) and (rpm is l) and (tph is m) then (output1 is el)
37. If (gf is l) and (rpm is s) and (tph is m) then (output1 is s)
38. If (gf is l) and (rpm is s) and (tph is vl) then (output1 is vs)
39. If (gf is s) and (rpm is m) and (tph is vs) then (output1 is lm)
40. If (gf is l) and (rpm is vs) and (tph is vl) then (output1 is vs)
41. If (gf is l) and (rpm is m) and (tph is m) then (output1 is m)
42. If (gf is m) and (rpm is vs) and (tph is vl) then (output1 is vs)
43. If (gf is s) and (rpm is vl) and (tph is s) then (output1 is el)
44. If (gf is m) and (rpm is vl) and (tph is l) then (output1 is el)
45. If (gf is l) and (rpm is vl) and (tph is s) then (output1 is el)

1.3 ANFIS Rule base for grinding process

1. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) then (output is out1mf1)
2. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf2) then (output is out1mf2)
3. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf3) then (output is out1mf3)
4. If (input1 is in1mf1) and (input2 is in2mf2) and (input3 is in3mf1) then (output is out1mf4)
5. If (input1 is in1mf1) and (input2 is in2mf2) and (input3 is in3mf2) then (output is out1mf5)
6. If (input1 is in1mf1) and (input2 is in2mf2) and (input3 is in3mf3) then (output is out1mf6)
7. If (input1 is in1mf1) and (input2 is in2mf3) and (input3 is in3mf1) then (output is out1mf7)
8. If (input1 is in1mf1) and (input2 is in2mf3) and (input3 is in3mf2) then (output is out1mf8)
9. If (input1 is in1mf1) and (input2 is in2mf3) and (input3 is in3mf3) then (output is out1mf9)
10. If (input1 is in1mf2) and (input2 is in2mf1) and (input3 is in3mf1) then (output is out1mf10)
11. If (input1 is in1mf2) and (input2 is in2mf1) and (input3 is in3mf2) then (output is out1mf11)

12. If (input1 is in1mf2) and (input2 is in2mf1) and (input3 is in3mf3) then (output is out1mf12)
13. If (input1 is in1mf2) and (input2 is in2mf2) and (input3 is in3mf1) then (output is out1mf13)
14. If (input1 is in1mf2) and (input2 is in2mf2) and (input3 is in3mf2) then (output is out1mf14)
15. If (input1 is in1mf2) and (input2 is in2mf2) and (input3 is in3mf3) then (output is out1mf15)
16. If (input1 is in1mf2) and (input2 is in2mf3) and (input3 is in3mf1) then (output is out1mf16)
17. If (input1 is in1mf2) and (input2 is in2mf3) and (input3 is in3mf2) then (output is out1mf17)
18. If (input1 is in1mf2) and (input2 is in2mf3) and (input3 is in3mf3) then (output is out1mf18)
19. If (input1 is in1mf3) and (input2 is in2mf1) and (input3 is in3mf1) then (output is out1mf19)
20. If (input1 is in1mf3) and (input2 is in2mf1) and (input3 is in3mf2) then (output is out1mf20)
21. If (input1 is in1mf3) and (input2 is in2mf1) and (input3 is in3mf3) then (output is out1mf21)
22. If (input1 is in1mf3) and (input2 is in2mf2) and (input3 is in3mf1) then (output is out1mf22)
23. If (input1 is in1mf3) and (input2 is in2mf2) and (input3 is in3mf2) then (output is out1mf23)
24. If (input1 is in1mf3) and (input2 is in2mf2) and (input3 is in3mf3) then (output is out1mf24)
25. If (input1 is in1mf3) and (input2 is in2mf3) and (input3 is in3mf1) then (output is out1mf25)
26. If (input1 is in1mf3) and (input2 is in2mf3) and (input3 is in3mf2) then (output is out1mf26)
27. If (input1 is in1mf3) and (input2 is in2mf3) and (input3 is in3mf3) then (output is out1mf27)

1.4 Sugeno FIS Rule base for the clinkerization process

1. If (SiO2 is vl) and (Al2O3 is vl) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is l) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is m)(C2S is s)(C3A is s)(C4AF is l) (1)
2. If (SiO2 is vl) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is m)(C3S is lm)(C2S is m)(C3A is s)(C4AF is um) (1)
3. If (SiO2 is vl) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is l) and (Inlet-Temperature is vs) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is m)(C2S is m)(C3A is s)(C4AF is um) (1)
4. If (SiO2 is vl) and (Al2O3 is l) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is s) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is es)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is m)(C3S is um)(C2S is vs)(C3A is lm)(C4AF is um) (1)

5. If (SiO2 is vl) and (Al2O3 is m) and (Fe2O3 is vl) and (CaO is l) and (Kiln-Feed is l) and (Kiln-RPM is vl) and (Inlet-Temperature is m) and (Coal-Feed is vl) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is vs)(Silica-Modulus is es)(Alumina-Modulus is es)(C3S is lm)(C2S is m)(C3A is es)(C4AF is el) (1)
6. If (SiO2 is s) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is l) and (Kiln-Feed is m) and (Kiln-RPM is vs) and (Inlet-Temperature is l) and (Coal-Feed is s) and (Kiln-Current is s) then (Free-Lime is vl)(Lime-Saturation-Factor is vs)(Silica-Modulus is vs)(Alumina-Modulus is s)(C3S is es)(C2S is el)(C3A is s)(C4AF is vl) (1)
7. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is es)(Silica-Modulus is vs)(Alumina-Modulus is vs)(C3S is s)(C2S is l)(C3A is vs)(C4AF is el) (1)
8. If (SiO2 is s) and (Al2O3 is m) and (Fe2O3 is l) and (CaO is s) and (Kiln-Feed is s) and (Kiln-RPM is vs) and (Inlet-Temperature is l) and (Coal-Feed is s) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is es)(Silica-Modulus is vs)(Alumina-Modulus is vs)(C3S is vs)(C2S is vl)(C3A is vs)(C4AF is vl) (1)
9. If (SiO2 is vl) and (Al2O3 is m) and (Fe2O3 is l) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is m) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is es)(Silica-Modulus is vs)(Alumina-Modulus is s)(C3S is s)(C2S is vl)(C3A is s)(C4AF is vl) (1)
10. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is vl) and (CaO is vs) and (Kiln-Feed is l) and (Kiln-RPM is s) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is vl) then (Free-Lime is s)(Lime-Saturation-Factor is lm)(Silica-Modulus is es)(Alumina-Modulus is vs)(C3S is s)(C2S is lm)(C3A is vs)(C4AF is el) (1)
11. If (SiO2 is s) and (Al2O3 is s) and (Fe2O3 is vs) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is s)(Lime-Saturation-Factor is vs)(Silica-Modulus is vs)(Alumina-Modulus is es)(C3S is lm)(C2S is um)(C3A is es)(C4AF is vl) (1)
12. If (SiO2 is m) and (Al2O3 is vs) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is es)(Silica-Modulus is es)(Alumina-Modulus is lm)(C3S is s)(C2S is l)(C3A is lm)(C4AF is vl) (1)
13. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is vl) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is m) and (Inlet-Temperature is s) and (Coal-Feed is l) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is vs)(Silica-Modulus is vs)(Alumina-Modulus is vs)(C3S is lm)(C2S is m)(C3A is vs)(C4AF is vl) (1)
14. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is vl) and (Kiln-Feed is m) and (Kiln-RPM is m) and (Inlet-Temperature is m) and (Coal-Feed is l) and (Kiln-Current is s) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is vs)(C3S is m)(C2S is s)(C3A is es)(C4AF is vl) (1)
15. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is l) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is vs)(Silica-Modulus is vs)(Alumina-Modulus is s)(C3S is lm)(C2S is um)(C3A is vs)(C4AF is vl) (1)
16. If (SiO2 is l) and (Al2O3 is vs) and (Fe2O3 is m) and (CaO is vs) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is m) then (Free-Lime is es)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is s)(C3S is um)(C2S is vs)(C3A is vs)(C4AF is vl) (1)
17. If (SiO2 is vs) and (Al2O3 is l) and (Fe2O3 is l) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is s) and (Inlet-Temperature is s) and (Coal-Feed is l) and (Kiln-

Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is vs)(C3S is m)(C2S is lm)(C3A is vs)(C4AF is vl) (1)

18. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vs) and (Kiln-RPM is s) and (Inlet-Temperature is l) and (Coal-Feed is s) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is m)(Silica-Modulus is lm)(Alumina-Modulus is s)(C3S is l)(C2S is es)(C3A is es)(C4AF is m) (1)
19. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is vl) and (CaO is l) and (Kiln-Feed is s) and (Kiln-RPM is m) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is vs)(C3S is m)(C2S is lm)(C3A is vs)(C4AF is vl) (1)
20. If (SiO2 is vl) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is vs)(C3S is lm)(C2S is m)(C3A is vs)(C4AF is vl) (1)
21. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is m) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is l)(Lime-Saturation-Factor is es)(Silica-Modulus is lm)(Alumina-Modulus is lm)(C3S is es)(C2S is el)(C3A is vs)(C4AF is lm) (1)
22. If (SiO2 is s) and (Al2O3 is s) and (Fe2O3 is vl) and (CaO is vl) and (Kiln-Feed is l) and (Kiln-RPM is vl) and (Inlet-Temperature is m) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is s)(Silica-Modulus is es)(Alumina-Modulus is s)(C3S is m)(C2S is lm)(C3A is s)(C4AF is vl) (1)
23. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is l) and (Kiln-Feed is m) and (Kiln-RPM is l) and (Inlet-Temperature is m) and (Coal-Feed is s) and (Kiln-Current is m) then (Free-Lime is es)(Lime-Saturation-Factor is es)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is s)(C2S is l)(C3A is lm)(C4AF is l) (1)
24. If (SiO2 is vl) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is l) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is s)(Lime-Saturation-Factor is vs)(Silica-Modulus is s)(Alumina-Modulus is vs)(C3S is s)(C2S is l)(C3A is vs)(C4AF is l) (1)
25. If (SiO2 is l) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is s) and (Kiln-Feed is s) and (Kiln-RPM is m) and (Inlet-Temperature is m) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is el)(Lime-Saturation-Factor is el)(Silica-Modulus is vs)(Alumina-Modulus is vl)(C3S is m)(C2S is vs)(C3A is m)(C4AF is s) (1)
26. If (SiO2 is m) and (Al2O3 is l) and (Fe2O3 is m) and (CaO is vl) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is es)(Silica-Modulus is s)(Alumina-Modulus is m)(C3S is vs)(C2S is el)(C3A is s)(C4AF is um) (1)
27. If (SiO2 is m) and (Al2O3 is l) and (Fe2O3 is l) and (CaO is l) and (Kiln-Feed is l) and (Kiln-RPM is s) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is l) then (Free-Lime is lm)(Lime-Saturation-Factor is m)(Silica-Modulus is es)(Alumina-Modulus is vs)(C3S is m)(C2S is lm)(C3A is vs)(C4AF is vl) (1)
28. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is s) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is es)(Silica-Modulus is lm)(Alumina-Modulus is m)(C3S is s)(C2S is l)(C3A is es)(C4AF is m) (1)
29. If (SiO2 is vl) and (Al2O3 is s) and (Fe2O3 is l) and (CaO is l) and (Kiln-Feed is l) and (Kiln-RPM is l) and (Inlet-Temperature is m) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is m)(C2S is s)(C3A is s)(C4AF is l) (1)

30. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is l) and (CaO is l) and (Kiln-Feed is l) and (Kiln-RPM is s) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is vl) then (Free-Lime is s)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is s)(C3S is lm)(C2S is m)(C3A is vs)(C4AF is vl) (1)
31. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is vl) and (Coal-Feed is l) and (Kiln-Current is vl) then (Free-Lime is s)(Lime-Saturation-Factor is vs)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is lm)(C2S is um)(C3A is s)(C4AF is vl) (1)
32. If (SiO2 is m) and (Al2O3 is vs) and (Fe2O3 is l) and (CaO is s) and (Kiln-Feed is vs) and (Kiln-RPM is s) and (Inlet-Temperature is m) and (Coal-Feed is s) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is s)(Alumina-Modulus is s)(C3S is um)(C2S is s)(C3A is vs)(C4AF is l) (1)
33. If (SiO2 is vl) and (Al2O3 is s) and (Fe2O3 is l) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is m) and (Coal-Feed is vl) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is lm)(Silica-Modulus is es)(Alumina-Modulus is lm)(C3S is m)(C2S is s)(C3A is s)(C4AF is l) (1)
34. If (SiO2 is m) and (Al2O3 is l) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is es)(Silica-Modulus is vs)(Alumina-Modulus is m)(C3S is vs)(C2S is vl)(C3A is lm)(C4AF is um) (1)
35. If (SiO2 is vl) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is l) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is es)(Alumina-Modulus is lm)(C3S is m)(C2S is s)(C3A is s)(C4AF is vl) (1)
36. If (SiO2 is l) and (Al2O3 is l) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is vs) and (Kiln-RPM is s) and (Inlet-Temperature is m) and (Coal-Feed is vs) and (Kiln-Current is s) then (Free-Lime is s)(Lime-Saturation-Factor is m)(Silica-Modulus is vs)(Alumina-Modulus is el)(C3S is m)(C2S is s)(C3A is l)(C4AF is s) (1)
37. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is s) and (Coal-Feed is vl) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is s)(C3S is lm)(C2S is m)(C3A is vs)(C4AF is l) (1)
38. If (SiO2 is s) and (Al2O3 is m) and (Fe2O3 is vl) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is s) then (Free-Lime is s)(Lime-Saturation-Factor is lm)(Silica-Modulus is es)(Alumina-Modulus is s)(C3S is m)(C2S is lm)(C3A is s)(C4AF is vl) (1)
39. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is l) and (Kiln-RPM is l) and (Inlet-Temperature is m) and (Coal-Feed is l) and (Kiln-Current is s) then (Free-Lime is s)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is s)(C3S is lm)(C2S is um)(C3A is vs)(C4AF is l) (1)
40. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is m)(C2S is lm)(C3A is s)(C4AF is vl) (1)
41. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is l) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is s) and (Kiln-Current is vs) then (Free-Lime is es)(Lime-Saturation-Factor is m)(Silica-Modulus is es)(Alumina-Modulus is lm)(C3S is um)(C2S is vs)(C3A is s)(C4AF is l) (1)
42. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is l) and (Kiln-Feed is vl) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-

Current is l) then (Free-Lime is s)(Lime-Saturation-Factor is es)(Silica-Modulus is s)(Alumina-Modulus is m)(C3S is s)(C2S is vl)(C3A is s)(C4AF is m) (1)

43. If (SiO2 is l) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is s) and (Kiln-RPM is vs) and (Inlet-Temperature is l) and (Coal-Feed is s) and (Kiln-Current is m) then (Free-Lime is lm)(Lime-Saturation-Factor is vs)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is vs)(C2S is vl)(C3A is s)(C4AF is um) (1)
44. If (SiO2 is vl) and (Al2O3 is m) and (Fe2O3 is l) and (CaO is vl) and (Kiln-Feed is l) and (Kiln-RPM is m) and (Inlet-Temperature is m) and (Coal-Feed is vl) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is es)(Alumina-Modulus is lm)(C3S is m)(C2S is s)(C3A is s)(C4AF is l) (1)
45. If (SiO2 is vl) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is vl) then (Free-Lime is el)(Lime-Saturation-Factor is um)(Silica-Modulus is vs)(Alumina-Modulus is s)(C3S is m)(C2S is s)(C3A is vs)(C4AF is um) (1)
46. If (SiO2 is vl) and (Al2O3 is s) and (Fe2O3 is s) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is l) and (Inlet-Temperature is s) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is vs)(Silica-Modulus is s)(Alumina-Modulus is l)(C3S is lm)(C2S is um)(C3A is m)(C4AF is lm) (1)
47. If (SiO2 is m) and (Al2O3 is vs) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is vl)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is s)(C3S is lm)(C2S is m)(C3A is vs)(C4AF is um) (1)
48. If (SiO2 is s) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is s) and (Inlet-Temperature is vl) and (Coal-Feed is s) and (Kiln-Current is vl) then (Free-Lime is s)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is m)(C2S is s)(C3A is s)(C4AF is um) (1)
49. If (SiO2 is vs) and (Al2O3 is l) and (Fe2O3 is vl) and (CaO is m) and (Kiln-Feed is l) and (Kiln-RPM is vs) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is es)(Alumina-Modulus is lm)(C3S is m)(C2S is lm)(C3A is s)(C4AF is vl) (1)
50. If (SiO2 is m) and (Al2O3 is vs) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is m) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is lm)(Lime-Saturation-Factor is lm)(Silica-Modulus is es)(Alumina-Modulus is s)(C3S is lm)(C2S is m)(C3A is es)(C4AF is um) (1)
51. If (SiO2 is l) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is s) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is vl) then (Free-Lime is vs)(Lime-Saturation-Factor is es)(Silica-Modulus is vs)(Alumina-Modulus is m)(C3S is lm)(C2S is um)(C3A is lm)(C4AF is um) (1)
52. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is vl) and (Inlet-Temperature is m) and (Coal-Feed is m) and (Kiln-Current is s) then (Free-Lime is es)(Lime-Saturation-Factor is vs)(Silica-Modulus is s)(Alumina-Modulus is l)(C3S is s)(C2S is um)(C3A is m)(C4AF is m) (1)
53. If (SiO2 is s) and (Al2O3 is s) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is l) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is s) and (Kiln-Current is s) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is s)(Alumina-Modulus is m)(C3S is m)(C2S is lm)(C3A is es)(C4AF is m) (1)
54. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is l) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is m) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is m)(Silica-Modulus is s)(Alumina-Modulus is s)(C3S is m)(C2S is s)(C3A is vs)(C4AF is um) (1)

55. If (SiO2 is s) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is l) and (Kiln-Feed is vl) and (Kiln-RPM is s) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is lm)(C2S is um)(C3A is lm)(C4AF is l) (1)
56. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is s) and (CaO is vl) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is m) and (Coal-Feed is vl) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is vs)(Silica-Modulus is s)(Alumina-Modulus is m)(C3S is lm)(C2S is l)(C3A is s)(C4AF is m) (1)
57. If (SiO2 is m) and (Al2O3 is vl) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is m) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is vs) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is m)(Silica-Modulus is es)(Alumina-Modulus is lm)(C3S is m)(C2S is s)(C3A is lm)(C4AF is l) (1)
58. If (SiO2 is l) and (Al2O3 is l) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is m) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is vl) then (Free-Lime is um)(Lime-Saturation-Factor is l)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is um)(C2S is vs)(C3A is s)(C4AF is um) (1)
59. If (SiO2 is l) and (Al2O3 is vs) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is m) and (Coal-Feed is l) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is m)(C2S is s)(C3A is s)(C4AF is um) (1)
60. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is l) and (Inlet-Temperature is m) and (Coal-Feed is m) and (Kiln-Current is s) then (Free-Lime is es)(Lime-Saturation-Factor is vs)(Silica-Modulus is vs)(Alumina-Modulus is um)(C3S is lm)(C2S is m)(C3A is lm)(C4AF is um) (1)
61. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is l) and (CaO is vs) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is lm)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is s)(C3S is m)(C2S is lm)(C3A is vs)(C4AF is um) (1)
62. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is vl) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is lm)(C2S is m)(C3A is el)(C4AF is m) (1)
63. If (SiO2 is s) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is l) and (Kiln-RPM is s) and (Inlet-Temperature is m) and (Coal-Feed is l) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is vs)(Silica-Modulus is s)(Alumina-Modulus is um)(C3S is lm)(C2S is um)(C3A is lm)(C4AF is m) (1)
64. If (SiO2 is l) and (Al2O3 is l) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is m) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is s) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is vl)(C3S is m)(C2S is lm)(C3A is um)(C4AF is lm) (1)
65. If (SiO2 is m) and (Al2O3 is l) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is l) and (Kiln-RPM is l) and (Inlet-Temperature is vl) and (Coal-Feed is l) and (Kiln-Current is vl) then (Free-Lime is s)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is m)(C2S is s)(C3A is s)(C4AF is l) (1)
66. If (SiO2 is s) and (Al2O3 is vs) and (Fe2O3 is s) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is lm)(Alumina-Modulus is m)(C3S is m)(C2S is m)(C3A is s)(C4AF is lm) (1)
67. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is vs) and (CaO is vs) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is s) and (Coal-Feed is vl) and (Kiln-

Current is l) then (Free-Lime is lm)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is s)(C2S is l)(C3A is s)(C4AF is um) (1)

68. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is s) and (Kiln-Current is m) then (Free-Lime is es)(Lime-Saturation-Factor is lm)(Silica-Modulus is s)(Alumina-Modulus is m)(C3S is m)(C2S is s)(C3A is s)(C4AF is m) (1)
69. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is vl) and (Inlet-Temperature is vl) and (Coal-Feed is s) and (Kiln-Current is s) then (Free-Lime is lm)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is s)(C2S is l)(C3A is s)(C4AF is m) (1)
70. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is s) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is um)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is s)(C3S is s)(C2S is um)(C3A is vs)(C4AF is l) (1)
71. If (SiO2 is vl) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is s) and (Kiln-RPM is m) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is vl) then (Free-Lime is vs)(Lime-Saturation-Factor is m)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is l)(C2S is s)(C3A is s)(C4AF is m) (1)
72. If (SiO2 is s) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is s) and (Kiln-RPM is l) and (Inlet-Temperature is m) and (Coal-Feed is vs) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is l)(C3S is lm)(C2S is m)(C3A is lm)(C4AF is lm) (1)
73. If (SiO2 is m) and (Al2O3 is vs) and (Fe2O3 is vs) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is lm)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is lm)(C2S is m)(C3A is vs)(C4AF is um) (1)
74. If (SiO2 is s) and (Al2O3 is m) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is l) and (Kiln-RPM is vl) and (Inlet-Temperature is m) and (Coal-Feed is l) and (Kiln-Current is m) then (Free-Lime is lm)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is lm)(C2S is m)(C3A is s)(C4AF is um) (1)
75. If (SiO2 is l) and (Al2O3 is vs) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is s)(Alumina-Modulus is l)(C3S is m)(C2S is lm)(C3A is lm)(C4AF is lm) (1)
76. If (SiO2 is l) and (Al2O3 is l) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is s)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is m)(C3S is m)(C2S is s)(C3A is lm)(C4AF is l) (1)
77. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is s) then (Free-Lime is s)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is lm)(C2S is m)(C3A is s)(C4AF is um) (1)
78. If (SiO2 is s) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is l) and (Kiln-Feed is l) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is l)(C3S is m)(C2S is s)(C3A is m)(C4AF is lm) (1)
79. If (SiO2 is s) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is m) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is um)(C3S is m)(C2S is m)(C3A is lm)(C4AF is m) (1)

80. If (SiO2 is s) and (Al2O3 is s) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is um)(C3S is m)(C2S is lm)(C3A is lm)(C4AF is um) (1)
81. If (SiO2 is l) and (Al2O3 is l) and (Fe2O3 is s) and (CaO is vl) and (Kiln-Feed is l) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is vl) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is l)(C3S is m)(C2S is s)(C3A is m)(C4AF is lm) (1)
82. If (SiO2 is l) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is lm)(C2S is m)(C3A is s)(C4AF is um) (1)
83. If (SiO2 is m) and (Al2O3 is vl) and (Fe2O3 is s) and (CaO is l) and (Kiln-Feed is l) and (Kiln-RPM is m) and (Inlet-Temperature is l) and (Coal-Feed is s) and (Kiln-Current is s) then (Free-Lime is s)(Lime-Saturation-Factor is m)(Silica-Modulus is es)(Alumina-Modulus is m)(C3S is m)(C2S is s)(C3A is lm)(C4AF is l) (1)
84. If (SiO2 is m) and (Al2O3 is l) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is s)(Lime-Saturation-Factor is m)(Silica-Modulus is es)(Alumina-Modulus is m)(C3S is m)(C2S is s)(C3A is lm)(C4AF is l) (1)
85. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is m) and (Inlet-Temperature is l) and (Coal-Feed is s) and (Kiln-Current is s) then (Free-Lime is vs)(Lime-Saturation-Factor is m)(Silica-Modulus is es)(Alumina-Modulus is l)(C3S is um)(C2S is vs)(C3A is m)(C4AF is m) (1)
86. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is l) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is s) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is lm)(C2S is m)(C3A is s)(C4AF is um) (1)
87. If (SiO2 is m) and (Al2O3 is l) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is l) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is vl) then (Free-Lime is lm)(Lime-Saturation-Factor is um)(Silica-Modulus is vs)(Alumina-Modulus is vl)(C3S is m)(C2S is vs)(C3A is um)(C4AF is lm) (1)
88. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is vl) then (Free-Lime is s)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is m)(C3S is lm)(C2S is m)(C3A is s)(C4AF is lm) (1)
89. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is lm)(Silica-Modulus is s)(Alumina-Modulus is s)(C3S is m)(C2S is s)(C3A is vs)(C4AF is um) (1)
90. If (SiO2 is m) and (Al2O3 is l) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is m) and (Kiln-RPM is m) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is el)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is vl)(C3S is es)(C2S is el)(C3A is m)(C4AF is s) (1)
91. If (SiO2 is l) and (Al2O3 is vs) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is vs)(C3S is m)(C2S is lm)(C3A is es)(C4AF is l) (1)
92. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-

Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is el)(C2S is lm)(C3A is s)(C4AF is um) (1)

93. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is es)(Lime-Saturation-Factor is es)(Silica-Modulus is s)(Alumina-Modulus is l)(C3S is s)(C2S is vl)(C3A is m)(C4AF is m) (1)
94. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is es)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is s)(C2S is vl)(C3A is s)(C4AF is l) (1)
95. If (SiO2 is s) and (Al2O3 is s) and (Fe2O3 is l) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is s)(Silica-Modulus is es)(Alumina-Modulus is m)(C3S is lm)(C2S is m)(C3A is m)(C4AF is vl) (1)
96. If (SiO2 is s) and (Al2O3 is m) and (Fe2O3 is l) and (CaO is l) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is es)(Alumina-Modulus is s)(C3S is m)(C2S is lm)(C3A is s)(C4AF is vl) (1)
97. If (SiO2 is vl) and (Al2O3 is m) and (Fe2O3 is l) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is m)(C2S is s)(C3A is s)(C4AF is l) (1)
98. If (SiO2 is l) and (Al2O3 is vs) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is vl) then (Free-Lime is es)(Lime-Saturation-Factor is vs)(Silica-Modulus is s)(Alumina-Modulus is m)(C3S is lm)(C2S is l)(C3A is lm)(C4AF is m) (1)
99. If (SiO2 is s) and (Al2O3 is s) and (Fe2O3 is s) and (CaO is vs) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is vs)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is lm)(C2S is um)(C3A is s)(C4AF is l) (1)
100. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is m) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is s)(C3S is m)(C2S is s)(C3A is s)(C4AF is l) (1)
101. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is lm)(Silica-Modulus is el)(Alumina-Modulus is lm)(C3S is lm)(C2S is lm)(C3A is vs)(C4AF is um) (1)
102. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is s) and (Kiln-Feed is m) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is vl) then (Free-Lime is m)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is vl)(C3S is s)(C2S is l)(C3A is m)(C4AF is es) (1)
103. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is l) and (Inlet-Temperature is m) and (Coal-Feed is vl) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is vs)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is lm)(C2S is um)(C3A is s)(C4AF is um) (1)
104. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is vl)(C3S is m)(C2S is lm)(C3A is s)(C4AF is um) (1)

105. If (SiO2 is vl) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is lm)(C2S is um)(C3A is s)(C4AF is um) (1)
106. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is m) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is lm)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is um)(C2S is s)(C3A is s)(C4AF is m) (1)
107. If (SiO2 is s) and (Al2O3 is s) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is vs)(Silica-Modulus is vs)(Alumina-Modulus is m)(C3S is lm)(C2S is um)(C3A is s)(C4AF is um) (1)
108. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is m)(C2S is m)(C3A is s)(C4AF is um) (1)
109. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is l) and (Kiln-Feed is m) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is vs)(Silica-Modulus is vs)(Alumina-Modulus is m)(C3S is lm)(C2S is um)(C3A is s)(C4AF is um) (1)
110. If (SiO2 is l) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is m) and (Coal-Feed is l) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is lm)(C2S is m)(C3A is s)(C4AF is um) (1)
111. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is l) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is vl) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is vs)(Silica-Modulus is vs)(Alumina-Modulus is m)(C3S is lm)(C2S is um)(C3A is lm)(C4AF is um) (1)
112. If (SiO2 is s) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is lm)(Lime-Saturation-Factor is vs)(Silica-Modulus is vs)(Alumina-Modulus is m)(C3S is s)(C2S is l)(C3A is lm)(C4AF is um) (1)
113. If (SiO2 is s) and (Al2O3 is l) and (Fe2O3 is s) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is es)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is m)(C3S is m)(C2S is lm)(C3A is lm)(C4AF is um) (1)
114. If (SiO2 is s) and (Al2O3 is s) and (Fe2O3 is l) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is l) and (Inlet-Temperature is vl) and (Coal-Feed is m) and (Kiln-Current is s) then (Free-Lime is m)(Lime-Saturation-Factor is lm)(Silica-Modulus is s)(Alumina-Modulus is s)(C3S is lm)(C2S is m)(C3A is s)(C4AF is vl) (1)
115. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is s)(Silica-Modulus is vs)(Alumina-Modulus is um)(C3S is s)(C2S is um)(C3A is m)(C4AF is m) (1)
116. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is m) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is um)(Lime-Saturation-Factor is m)(Silica-Modulus is s)(Alumina-Modulus is um)(C3S is lm)(C2S is m)(C3A is lm)(C4AF is es) (1)
117. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is s) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-

Current is s) then (Free-Lime is l)(Lime-Saturation-Factor is m)(Silica-Modulus is s)(Alumina-Modulus is l)(C3S is lm)(C2S is m)(C3A is m)(C4AF is s) (1)

118. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is l) and (Kiln-Current is l) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is lm)(Alumina-Modulus is l)(C3S is lm)(C2S is m)(C3A is m)(C4AF is s) (1)
119. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is vs)(Silica-Modulus is s)(Alumina-Modulus is l)(C3S is el)(C2S is um)(C3A is m)(C4AF is lm) (1)
120. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is s) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is lm)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is m)(C2S is lm)(C3A is s)(C4AF is um) (1)
121. If (SiO2 is l) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is s)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is um)(C3S is lm)(C2S is um)(C3A is lm)(C4AF is lm) (1)
122. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is m) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is es)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is l)(C3S is lm)(C2S is m)(C3A is m)(C4AF is lm) (1)
123. If (SiO2 is l) and (Al2O3 is s) and (Fe2O3 is s) and (CaO is s) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is es)(Lime-Saturation-Factor is lm)(Silica-Modulus is s)(Alumina-Modulus is m)(C3S is um)(C2S is s)(C3A is s)(C4AF is m) (1)
124. If (SiO2 is m) and (Al2O3 is s) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is l) then (Free-Lime is lm)(Lime-Saturation-Factor is vs)(Silica-Modulus is vs)(Alumina-Modulus is lm)(C3S is es)(C2S is vl)(C3A is s)(C4AF is l) (1)
125. If (SiO2 is l) and (Al2O3 is vs) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is vl) then (Free-Lime is s)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is lm)(C3S is lm)(C2S is um)(C3A is s)(C4AF is el) (1)
126. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is s) and (Kiln-Feed is l) and (Kiln-RPM is vl) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is s) then (Free-Lime is m)(Lime-Saturation-Factor is lm)(Silica-Modulus is s)(Alumina-Modulus is m)(C3S is lm)(C2S is um)(C3A is s)(C4AF is m) (1)
127. If (SiO2 is m) and (Al2O3 is m) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is m) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is s) and (Kiln-Current is m) then (Free-Lime is es)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is m)(C3S is lm)(C2S is m)(C3A is s)(C4AF is m) (1)
128. If (SiO2 is m) and (Al2O3 is l) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is s) and (Kiln-RPM is s) and (Inlet-Temperature is m) and (Coal-Feed is s) and (Kiln-Current is vl) then (Free-Lime is s)(Lime-Saturation-Factor is m)(Silica-Modulus is vs)(Alumina-Modulus is l)(C3S is m)(C2S is s)(C3A is um)(C4AF is m) (1)
129. If (SiO2 is m) and (Al2O3 is l) and (Fe2O3 is m) and (CaO is m) and (Kiln-Feed is vl) and (Kiln-RPM is l) and (Inlet-Temperature is l) and (Coal-Feed is m) and (Kiln-Current is m) then (Free-Lime is vs)(Lime-Saturation-Factor is s)(Silica-Modulus is s)(Alumina-Modulus is m)(C3S is m)(C2S is m)(C3A is lm)(C4AF is m) (1)

List of Publications

Journal Articles

1. Pani, A., & Mohanta, H. K. (2009). Application of soft sensors in process monitoring and control: A review. *The IUP Journal of Science & Technology*, *5*(4), 7-20.
2. Pani, A. K., & Mohanta, H. K. (2011). A survey of data treatment techniques for soft sensor design. *Chemical Product and Process Modeling*, *6*(1).
3. Pani, A. K., Vadlamudi, V. K., & Mohanta, H. K. (2013). Development and comparison of neural network based soft sensors for online estimation of cement clinker quality. *ISA Transactions*, *52*(1), 19-29.
4. Pani, A. K., & Mohanta, H. K. (2014). Soft sensing of particle size in a grinding process: Application of support vector regression, fuzzy inference and adaptive neuro fuzzy inference techniques for online monitoring of cement fineness. *Powder Technology*, *264*, 484-497.
5. Pani, A. K., & Mohanta, H. K. (2015). Online monitoring and control of particle size in the grinding process using least square support vector regression and resilient back propagation neural network. *ISA Transactions* (*In Press*)
6. Pani, A. K., & Mohanta, H. K. Inferential sensing of cement clinker quality using robust multivariate outlier detection and soft computing techniques. (*Communicated*)

Conference Proceedings

1. Pani A. K., & Mohanta H. K., (2011, February) Importance of Data Analysis and Treatment for Soft Sensor Design: Application to Continuous Rotary Cement Kiln. 2nd Conference on Advances in Chemical Engineering, Thapar University, *AChemE - 2011*.
2. Pani, A. K., Vadlamudi, V., Bhargavi, R. J., & Mohanta, H. K. (2011, July). Neural Network Soft Sensor Application in Cement Industry: Prediction of Clinker Quality Parameters. In *Process Automation, Control and Computing (PACC), 2011 International Conference on* (pp. 1-6). IEEE.
3. Vadlamudi V. K., Pani A. K., Bhargavi R. J., Mohanta H. K, (2011, December). Development of Fuzzy Logic Based Soft Sensor for Prediction of Free Lime Content in the Cement Clinker. Indian Chemical Engineering Congress *CHEMCON – 2011*.
4. Pani A. K., Vadlamudi V. K., Bhargavi R. J., Mohanta H. K., An RBF Neural Network based Soft Sensor Development for Prediction of Cement Clinker Properties. Indian Chemical Engineering Congress *CHEMCON – 2011*.
5. Pani, A. K., Amin, K. G., & Mohanta, H. K. (2012, July). Data driven soft sensor of a cement mill using generalized regression neural network. In *Data Science & Engineering (ICDSE), 2012 International Conference on* (pp. 98-102). IEEE.
6. Pani, A. K., & Mohanta, H. K. (2013, February). A hybrid soft sensing approach of a cement mill using principal component analysis and artificial neural networks. In *Advance Computing Conference (IACC), 2013 IEEE 3rd International* (pp. 713-718). IEEE.

Biographies

Biography of the Candidate

Mr Ajaya Kumar Pani has done his Graduation in Chemical Engineering from National Institute of Technology, Rourkela and Post Graduation in Chemical Engineering from Institute of Technology, Banaras Hindu University, Varanasi. He worked as a Junior Project Fellow at Regional Research Laboratory (Presently Institute of Minerals and Materials Technology), Bhubaneswar for 1 year and as a Graduate Engineer Trainee at J K Paper Mills, Rayagada, Odisha for 1 year. Prior to joining BITS, he served as a Lecturer at Jagannath Institute of Technology & Management (Presently Centurion University), Parlakhemundi, Odisha for 4 years. He joined BITS Pilani in the year 2007 and has since been working as a Lecturer in the Department of Chemical Engineering at BITS-Pilani, Pilani Campus.

Biography of the Supervisor

Dr Hare Krishna Mohanta is an Assistant Professor in the Department of Chemical Engineering in BITS-Pilani, Pilani Campus, Rajasthan. He has over 16 years of teaching experience and 1 year industrial experience. He has obtained his B.E. (Chemical Engineering) degree in 1995 from NIT Rourkela, M.Tech. (Chemical Engineering) in 1998 from IIT Kanpur, and Ph.D. in Chemical Engineering (Specialization: Process Control) in 2006 from BITS Pilani. He is a Life Associate Member of Indian Institute of Chemical Engineers (IIChE) and a Member of Institution of Engineers (India). He has several publications in the international and national journals. His areas of research include wavelet-based process identification & control, design of soft sensors for process monitoring & control, reactive distillation design & control, nonlinear control & state estimation, catalytic pyrolysis of hydrocarbons besides modeling & simulation.