Addis Ababa University

Addis Ababa Institute of Technology

Department of Electrical and Computer Engineering

NEURAL NETWORK BASED DATA-DRIVEN CLINKER QUALITY PREDICTION: CASE STUDY ON MUGHER CEMENT FACTORY

A Thesis Submitted to Addis Ababa Institute of Technology, School of Graduate Studies, Addis Ababa University

In Partial Fulfillment of the Requirement for the Degree of MASTER OF SCIENCE IN ELECTRICAL ENGINEERING (INDUSTRIAL CONTROL ENGINEERING)

By

Mihreteab Negash

Advisor: Dr. Eng. Getachew Alemu

October, 2012

Addis Ababa University

Addis Ababa Institute of Technology

Department of Electrical and Computer Engineering

NEURAL NETWORK BASED DATA-DRIVEN CLINKER QUALITY PREDICTION: CASE STUDY ON MUGHER CEMENT FACTORY

By Mihreteab Negash

APPROVED BY BOARD OF EXAMINERS

Chairman, Department of Graduate Committee	Signature
Advisor	Signature
Internal Examiner	Signature
External Examiner	Signature

DECLARATION

I, the undersigned, declare that this thesis is my original work, has not been presented for a degree in this or other universities, all sources of materials used for this thesis work have been fully acknowledged.

Name: Mihreteab Negash Signature: _____

Place: Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa

Date of Submission: October, 2012

This thesis has been submitted for examination with my approval as a university advisor.

Advisor's Name: Dr. Eng. Getachew Alemu Signature: _____

Dedication

... to my supportive family

especially

to my courageous and caring mother

Genet Yemanebirhan

ACKNOWLEDGEMENTS

I am grateful for the support given by many peoples in accomplishing this thesis. Peoples from Addis Ababa University, Mugher cement enterprise, and friends have contributed much at different stages of the thesis work.

I would like to thank my advisor Getachew Alemu (Dr. Eng.) for his continuous supervision and encouragement. I am also thankful for the big effort made by Ato Lebsework Negash (M.Sc.) – *lecturer at the department of electrical and computer engineering (AAiT)* - to establish a link with the enterprise. Furthermore, I like to thank Fitsum Sebsibe – *graduate student of the school of pharmacy (AAU),* Samuel Sisay – *graduate student of computer science (AAU)* and Mohammed Muntaz- *graduate student in huazhong university of science and technology (China)* - for their good hand at various stages. Moreover, the discussions on the thesis matters that I have with Bemnet Wendimagegnehu -*a friend and classmate*- were helpful.

As the thesis is a case study, I would like to thank Mugher cement enterprise for the permission and collaboration offered. I have gratitude that Ato Mehari Yalew–*coal firing project manager*- has played a great role in going through official procedures and arranging accommodation and transportation. Furthermore, Ato Fasil Ayele – *senior process engineer,* Ato Yirgalem Addis - *electrical engineer*, Ato Gebeyew Werke –*mechanical engineer*, Ato Sisay Cheru- *cement production team leader*, Ato Yasin Juhar- *mechanical engineer* and Ato Kebede Egersa- *senior quality assurance expert*- have helped me to visit the plant, while collecting data and throughout the continuous interactions to discuss matters involved in the thesis work. Last but not least, I am thankful that Ato Biniam Aynalem-*driver*- has been giving me a ride to Mugher whenever needed.

Mihreteab Negash

ABBREVIATIONS

AAiT	Addis Ababa Institute of Technology
AAU	Addis Ababa University
AM	Alumina Modulus
C_2S	Belite
C_3A	Aluminate
C_3S	Allite
C_4AF	Ferrite
LOI	Loss On Ignition
LSF	Lime Saturation Factor
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
MSE	Mean Squared Error
SCG	Scaled Conjugate Gradient
SM	Silica Modulus
XRF	X-Ray Fluorescence

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

ABSTRACT

Soft sensors are key solutions in predicting importance process variables. In process industries, important parameters which are difficult or cost a lot to measure online can be predicted using soft sensors. In this thesis a data driven soft sensor is developed using neural network to predict important clinker quality parameters.

The developed predictor is significant and can be categorized to the class of neural network based soft sensors. The significance of the thesis is that it avoids measurement delay incurred while analyzing clinker samples. As a result, quick control actions can be taken and clinker quality can be further improved. This is one of the solutions provided by soft sensors. Many soft sensors have been developed in different application areas and cement factory is the one. Some papers report neural network based predictors that are developed on rotary cement kiln. These works are related to the thesis. However, the thesis has its own new contribution. The first new feature is that it has developed data synthesis strategy. Besides, multiple and advanced neural network architectures are used to get improved result. Moreover, it is of the first kind for the selected case, which is the third line of Mugher cement factory.

The thesis is developed stage wise and a desired result is obtained. First, cement production specific to the case is studied. Then, data of all the recorded variables in the factories database is collected. This data collection is accompanied by variable selection and data encoding. The data is processed prior to using it for training the neural networks. This data preprocessing treated missing and outlier values. Based on the cleaned data, new data is synthesized to have enough dataset to work on. Finally, neural network models are developed and trained on this dataset. As a result, neural network models are obtained that can predict LSF, SM, AM and C_3S values of clinker with mean square error values of 4.3482, 0.0027, 0.0011 and 10.8759 respectively.

In conclusion, in this thesis a neural network based data driven clinker quality predictor is developed. While developing the predictor, Mugher cement factory is used as a case study. The developed predictor estimates LSF, SM, AM and C_3S values

Key words: Soft sensor, neural network, clinker quality prediction.

CHAPTER 1

INTRODUCTION

1.1 Motivation and case selection

Soft sensors are key solutions in process industries. I get motivated on building intelligent soft sensors, to estimate variables in process industries using neural network, while going through a book called Neural Systems for Control by Omid M. Omidvar and David L. Elliott. Then after, I needed a case to work on. A doctorate thesis by Petr Kaldec on robust and adaptive soft sensor discussed many application areas of soft sensors [1]. Using this as a ground work, I have selected a factory to work on as a case among many application areas.

Initially pharmaceutical industries were chosen as an application area. In these industries penicillin quality can be predicted using soft sensors. Unfortunately, I have found after a survey that none of the pharmaceutical industries in Ethiopia has the fermentor, which is used to produce penicillin. Therefore, a shift in application area selection became a must.

Pulp & paper, sugar and cement factories are other application areas to choose from. Because of the current increase and expansion in cement factories in Ethiopia, I selected the cement sector. After this I searched for a cement factory which has rotary kiln, close to Addis Ababa and operational. Up on these criteria, Mugher cement factory is found to be the case to work on.

Finally, my motivation on building intelligent soft sensors and search for an application area settled on developing neural network based clinker quality predictor by taking Mugher cement factory as a case study.

1.2 Statement of the Problem

There are two major problems while measuring clinker quality parameters. The first one is measurement delay .The time delay of the laboratory analysis is around four hours and that of the X-Ray Fluorescence (XRF) technique is about fifteen minutes. These delays cause hindrance in communicating quality report, resulting in difficulties of making timely control adjustment whenever it is required. This in turn significantly affects producing good quality clinker. The second one is lack of backup for the XRF. In case of malfunction, quality measurement cannot be done using the XRF and there is no backup to supplement due to its high cost. Therefore, measurement delays and absence of backup indicates the need for clinker quality prediction.

1.3 Significance of the thesis

The thesis has two sided importance. On the first, the neural network based quality prediction avoids measurement delay and enables quick control actions to be taken. Therefore, the developed predictor can supplement clinker quality measurement and aids on effective kiln operation. As a result, clinker quality can be further improved. On the other side, as the first of its kind, it creates motivation on applying neural network based soft sensor to inland process industries.

1.4 Objective of the thesis

This thesis on the stated problem has general objective and specific objectives. The general objective of the thesis is to develop a neural network based data-driven clinker quality predictor by taking Mugher cement factory as a case study. The specific objectives are:

- To study cement production in Mugher cement factory.
- To collect historical process data.
- To pre-process the collected data.
- To develop different neural network based data-driven predictors.

- To select the better predictor after testing and comparing the different predictors.

1.5 Scope of the thesis

Neural network based data-driven clinker quality predictor will be developed taking Mugher cement factory as a case study. Not all clinker quality related parameters will be predicted, rather only the major ones. Though clinker quality indicates the quality of cement, the focus of this thesis is specifically clinker production not the overall cement production process.

1.6 About Mugher cement factory

Mugher cement factory is located 90 km north west of Addis Ababa at attitude of 2450 m in west Shoa zone of Oromia region. It has three production lines with a total capacity of 1.5 million tons of clinker per year. The thesis is done on the third line with production capacity of 3000 tons of clinker per day. The production line uses dry process and has precalciner [13].

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction to cement manufacturing process

Cement production is an energy intensive manufacturing process. As a result, adding a little improvement in the system means a lot of gain. Rotary kiln, in which clinker is produced, is the heart of the process. This kiln contains nonlinearity, lag and there is no precise mathematical model to represent it [10]. Thus, neural network models are developed to predict clinker quality parameters, whereby contributing to clinker quality improvement. In this section the cement manufacturing process and clinker parameters are discussed.

2.1.1 Cement manufacturing process

Portland cement is the most common type of cement in general use around the world. It is the type of cement produced by Mugher cement factory. There are four fundamental stages in the production of Portland cement: quarrying, raw material preparation, clinkering and cement milling [2,3,20,24].

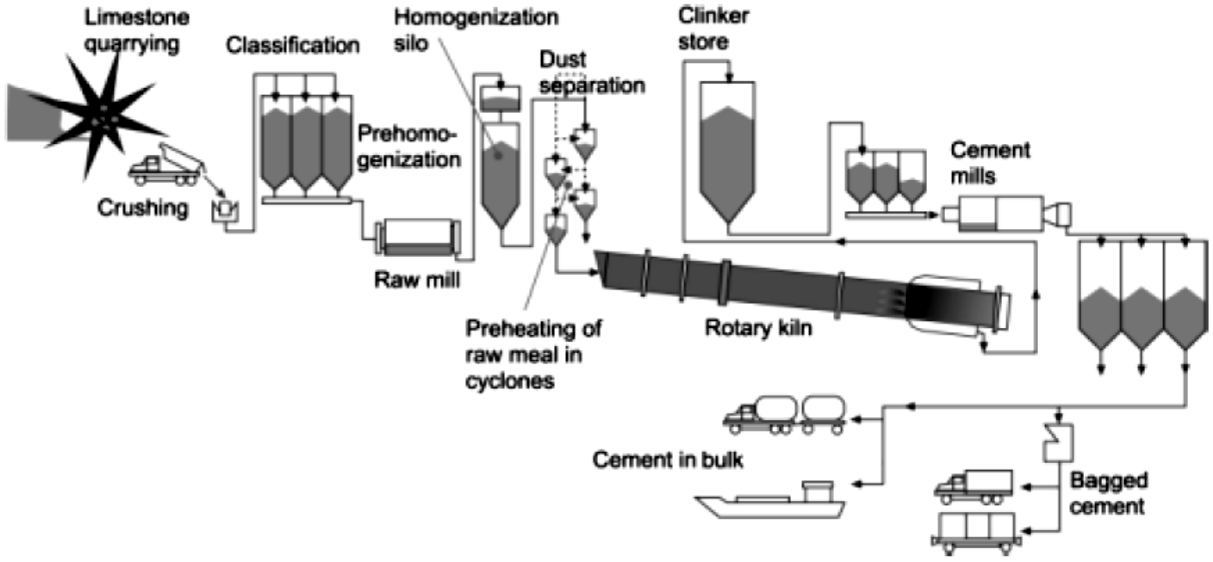


Figure 2.1.1: Cement manufacturing from limestone quarrying to cement bagging [25].

Step 1: Quarrying

The raw material for cement manufacture is a rock mixture of limestone (which is rich in $CaCO_3$) and clay (a source of silica, alumina and Fe_2O_3). These are quarried and stored separately. The lime and silica provide the main strength of the cement, while the iron reduces the reaction temperature and gives the cement its characteristic grey color.

Step 2: Raw material preparation

The steps involved here depend on the process used. There are two main cement manufacturing processes: the dry process and the wet process.

The dry process

The quarried clay and limestone are crushed separately. Then these raw materials are mixed in the required proportion, if necessary, corrective minerals are added. The mix is then fed into a mill.

The wet process

The clay is mixed to a paste in a washmill – a tank in which the clay is pulverized in the presence of water. Crushed lime is then added and the whole mixture further ground. Any material which is too coarse is extracted and reground. The slurry is then tested to ensure that it contains the correct balance of minerals, and any extra ingredients blended in as necessary.

Step 3: Clinkering

This is the step which is characteristic of Portland cement. The finely ground material is dried, heated (to enable the sintering reactions to take place) and then cooled down again. While it is being heated various chemical reactions take place to form the major mineral constituents of Portland cement. The powder from the dry process does not contain much moisture, so can be dried in a preheater tower. As it falls through the tower it is heated from 70 to 800^{o}C. The moisture evaporates, up to 20% of the decarbonation (loss of CO_2) occurs and some intermediate phases such as $CaO.Al_2O_3$ begin to appear. The mixture is then fed into the kiln. The slurry from the wet process contains too much moisture to be successfully dried in a preheater tower. Instead,

the slurry is fed directly into the kiln where it is formed into dry balls by the heat and rotation of the kiln. Because of this extra role of the kiln, wet process kilns are generally longer than dry process kilns. The kilns used in both processes are inclined on a shallow angle and lined with heat-resistant bricks.

Step 4: Cement milling

To produce the final product the clinker is mixed with gypsum, which is added as a set retarder, and ground in the cement mill.

2.1.2 Portland cement clinker and clinker formation

Portland cement clinker

Portland cement clinker is a dark grey nodular material made by heating ground limestone and clay at a temperature of about 1400^{o}C- 1500^{o}C. The nodules are ground up to a fine powder to produce cement, with a small amount of gypsum added to control the setting properties.

Clinker formation

Portland cement clinker mainly consists of CaO, SiO_2, Al_2O_3 and Fe_2O_3; these accounts for more than 95%. The minor components in total less than 3% are MgO, TiO_2, P_2O_5 and alkalis. In clinker, they are not present in individual oxide, but exist as compounds formed by two or more oxides. The mineral phases consist mainly of alite, belite, aluminate and ferrite.

- Allite (C_3S) is the most important constituent, 50-70% in normal Portland cement clinkers.
- Belite (C_2S) constitutes 15-30% of normal Portland cement clinkers.
- Aluminate (C_3A) constitutes 5-10% of most normal Portland cement clinkers.
- Ferrite (C_4AF) makes up 5-15% of normal Portland cement clinker

There is a set of reactions in the kiln to form clinker [2,3]. Up to a temperature of about 700^{o}C, activation of the silicates through the removal of water and changes in the crystal structures take

place. Within the temperature range 700^{o}C-900^{o}C, decarbonation of the calcium carbonate (calcite) occurs, together with the initial combination of the alumina, ferric oxide and of activated silica with lime. From 900^{o}C to 1200^{o}C, belite forms. Above 1250^{o}C and more particularly above 1300^{o}C, the liquid phase appears and this promotes the reaction between belite and free lime to form alite. The molten phase goes to a glass or, if cooling is slow, the C_3A crystallizes out and in extreme cases the alite dissolves back into the liquid phase and reappears as secondary belite. Alkali sulfates condense out as a separate phase during the cooling process. The principal reactions taking place are conveniently divided into three groups, as follows:

(1) Reactions below about 1300°C, of which the most important are (a) the decomposition of calcite (calcining), (b) the decomposition of clay minerals, and (c) reaction of calcite or lime formed from it with quartz and clay mineral decomposition products to give belite, aluminate and ferrite. Liquid is formed only to a minor extent at this stage, but may have an important effect in promoting the reactions. At the end of this stage, the major phases present are belite, lime, aluminate and ferrite. The last two may not be identical with the corresponding phases in the final product.

(2) Reactions at 1300-1450°C (clinkering): A melt is formed, mainly from the aluminate and ferrite, and by 1450°C some 20-30% of the mix is liquid. Much of the belite and nearly all the lime react in the presence of the melt to give alite. The material nodulizes, to form the clinker.

(3) Reactions during cooling: The liquid crystallizes, giving mainly aluminate and ferrite. Polymorphic transitions of the alite and belite occur.

The variation in typical contents of phases during the above reactions is shown in fig.2.1.2 below.

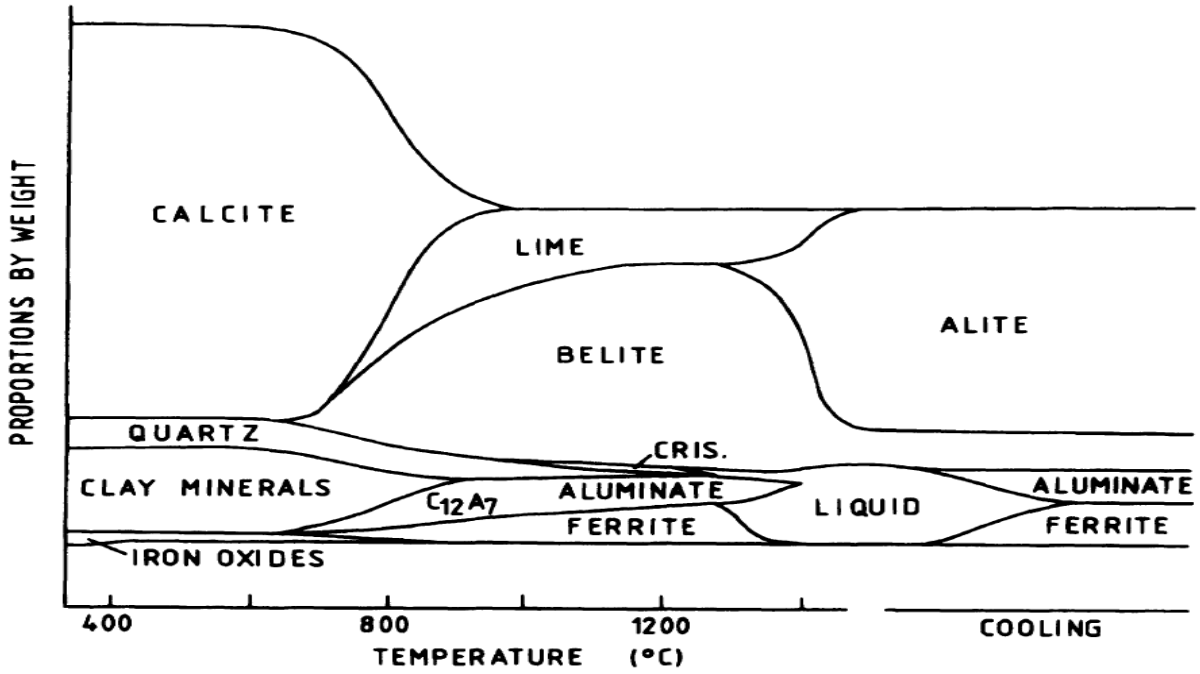


Figure 2.1.2: Variations in typical contents of phases during the formation of Portland cement clinker [2].

2.1.3 Clinker moduli and minerals of clinker

Parameters based on the oxide composition are very useful in describing clinker characteristics [5]. The following parameters are widely used (chemical formulae represent weight percentages).

2.1.3.1 Clinker moduli

The content of the major clinker oxides is controlled by using modulus calculations to maintain the required proportion of CaO, SiO_2, Al_2O_3 and Fe_2O_3. The moduli are calculated as follows [5]:

1. Lime saturation factor (LSF)

$$\text{Lime Saturation factor (LSF)} = \frac{100\ \text{CaO}}{2.8\ \text{SiO}_2 + 1.18\ \text{Al}_2\text{O}_3 + 0.65\text{Fe}_2\text{O}_3} \ldots\ldots\ldots \text{eq. 2.1.1}$$

- LSF is the amount of CaO which is enough to saturate SiO_2, Al_2O_3, Fe_2O_3 to form Portland cement clinker.
- LSF governs the ratio of alite to belite and also shows whether the clinker is likely to contain an unacceptable proportion of the free lime.

If the value of LSF is higher than the usual (90-98%):

- Fuel consumption increases.
- Burning zone temperature increase and heat loss by radiation increase.
- Brick work will be damaged.
- Reducing condition in the kiln is formed which greatly affects clinker quality and brick life.
- Large clinker nodules are formed which deteriorate clinker quality.
- Free CaO increases which affects the quality of clinker and production of sound cement.

When LSF is lower:

- Clinker burning is easy with low heat of consumption.
- Free lime content is low and the lower the early strength of the cement.

2. Silica modulus (SM)

$$\text{Silica Modulus (SM)} = \frac{SiO_2}{Al_2O_3 + Fe_2O_3} \ldots\ldots\ldots \text{eq. 2.1.2}$$

- It characterizes the ratio of solid to liquid in the clinker.
- Its major effect is on the quantity of liquid phase or flux potentially present out at clinkering temperature.

If the value of SM is higher than the usual (2.0 – 2.4):

- Clinker is hard to grind.
- The raw mix is difficult to burn due to lower content of Al_2O_3 and Fe_2O_3.
- Low cement strength

When SM is lower than the usual:

- The raw mix is easy to burn.
- Ring and ball formation is increased due to excess liquid.
- Slow hardening of cement.

3. Alumina modulus (AM)

$$\text{Alumina Moduls (AM)} = \frac{Al_2O_3}{Fe_2O_3} \ldots\ldots\ldots \text{eq. 2.1.3}$$

It governs the proportion of aluminate to ferrite Phases in the clinker, which has important effects on cement properties and also determines the quantity of liquid formed at relatively low temperature.

If the value of any higher than the usual (1.5-1.65):

- Hard burning
- High early cement strength with low sulfate resistance.

When AM is lower than the usual:

- Easily burnable
- Low early cement strength with high sulfate resistance and low heat of hydration.

2.1.3.2 Clinker minerals

The oxides of calcium, silicon, aluminum and iron are the four major components that react to form the main clinker minerals. These minerals are:

Alite (C_3S)

- It is the main strength constituent of cement.
- Used for rapid hydration, high initial and good final strength.

Belite (C_2S)

- Has low heat of hydration.
- Contribute to late strength.

Aluminate (C_3A)

- Used for rapid hydration that promotes early strength.
- Reacts with sulfate and undergoes volume expansion.
- Shrinks appreciably on hydration.

Ferrite (C_4AF)

- Contribute little or no for strength development.
- Gives normal cement its colors
- Has moderate heat of hydration.

The approximate proportion of the four main minerals in Portland cement clinker is calculated using a set of equations called Bogue equation [5]. These equations are listed as follows

$$C_3S = 4.07(CaO - Free\ CaO) - 7.6SiO_2 - 1.43Fe_2O_3 - 6.7Al_2O_3 - 2.85SO_3 \quad \ldots . eq.\ 2.1.4$$
$$C_2S = 2.87SiO_2 - 0.75C_3S \quad \ldots\ldots\ldots . eq.\ 2.1.5$$
$$C_3A = 2.65Al_2O_3 - 1.69Fe_2O_3 \quad \ldots\ldots\ldots . eq.\ 21..6$$
$$C_4AF = 3.04Fe_2O_3 \quad \ldots\ldots\ldots . eq.\ 2.1.7$$

2.2 Material composition measurement techniques in cement industry

Material composition measurement is important in cement manufacturing. The measurement is used to acquire target clinker composition. The composition of raw materials, kiln meal and clinker can be determined using wet chemical analysis and XRF technique.

Wet chemical analysis and XRF techniques are two different methods. In the case of wet chemical analysis, samples of raw material, kiln meal or clinker is analyzed in a chemical laboratory. However, the XRF technique utilizes electromagnetic radiation to analyze the specimen prepared.

2.3 Paper works on soft sensors in cement factory

Soft sensors have application in predicting important process parameters. Among many process industries cement factory is the one to get the benefit of soft sensors. There are many papers on the application of soft sensors [6, 7, 8, 9, 10,11].

Water quality parameters prediction is done using multilayer perceptron network [6], the quality of polypropylene is predicted using soft sensor [7], hierarchical neural network is used to predict product quality of industrial ethylene pyrolysis process [8] and cascade and feedforward backpropagation neural network models are used for prediction of sensory quality of an instant drink [9]. These papers show the importance of soft sensors for process parameter prediction though none of them involves cement manufacturing.

Neural network based temperature predictor for cement rotary kiln is reported in a paper [10]. In this predictor backpropagation and Elman neural network are used. This paper serves as a good illustration of the benefits of soft sensor in cement factory. However, it is not directly linked to clinker quality prediction, which is the focus of this thesis.

Neural network based clinker quality parameters prediction is also reported [11]. This predictor has used backpropagation neural network for simultaneous prediction of free lime, C_3S, C_2S and C_3A. Besides, the paper gives recommendation on using some other network architectures to further improve the result.

CHAPTER 3

MATERIAL AND METHODOLOGY

The basic steps in any data driven soft sensor design method are data collection, data preprocessing, model selection and training and finally model validation which are shown in the schematic diagram below [12].

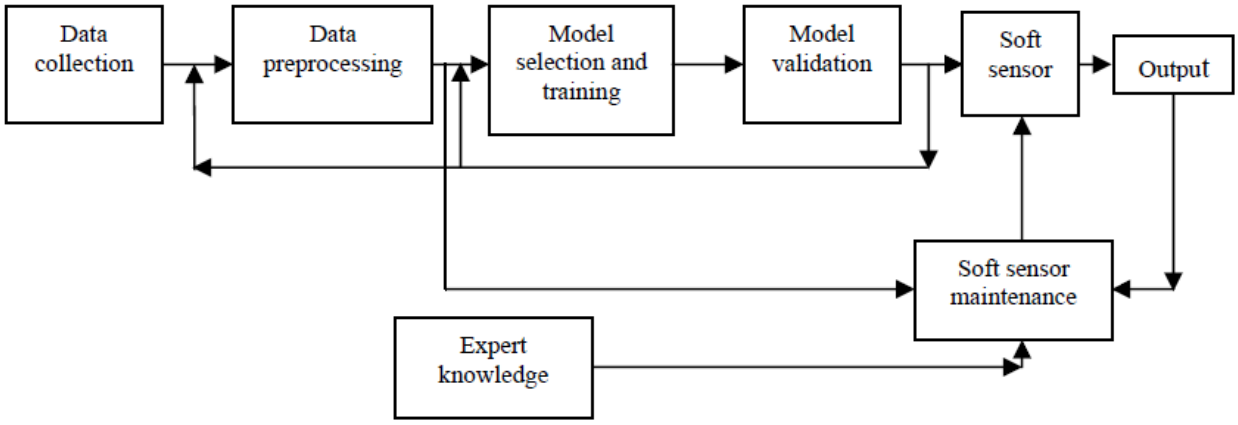


Figure 3.1: Soft sensor design steps

3.1 Data collection

An industrial database provides data of all the variables that are recorded. However, all the available variable data is not relevant to the process variables to be estimated. Thus, relevant variables are selected with the help of experts from Mugher cement factory. Then the corresponding data is collected and encoded.

3.1.1 Variable selection

Up on a five day visit to Mugher cement factory, in two phases, relevant variables regarding clinker quality are selected out. This will not be possible without the genuine assistance offered

by experts from Mugher cement factory. Accordingly, the model will have in total of twelve inputs and four outputs. These parameters are shown in table 3.1.1 below.

	Input		Output
List no.	Operating parameters	Kiln meal variables	Clinker variables
1	Secondary air temperature (^oC)	SiO_2	Lime saturation factor (LSF)
2	Calciner rising pipe temperature (^oC)	Al_2O_3	Silica modulus (SM)
3	Kiln gas flue chamber temperature (^oC)	Fe_2O_3	Alumina modulus (AM)
4	Quantity of oil before kiln burner (Kg/hr)	CaO	Alite (C_3S)
5	Quantity of oil before calciner burner (Kg//hr)		
6	Kiln speed (rpm)		
7	Kiln meal flow rate (ton/hr)		
8	Hot gas ID fan power (A)		

Table 3.1.1: List of input-output variables

The following free hand sketch, in blocks, of part of the third line shows points of actions of some of the relevant variables.

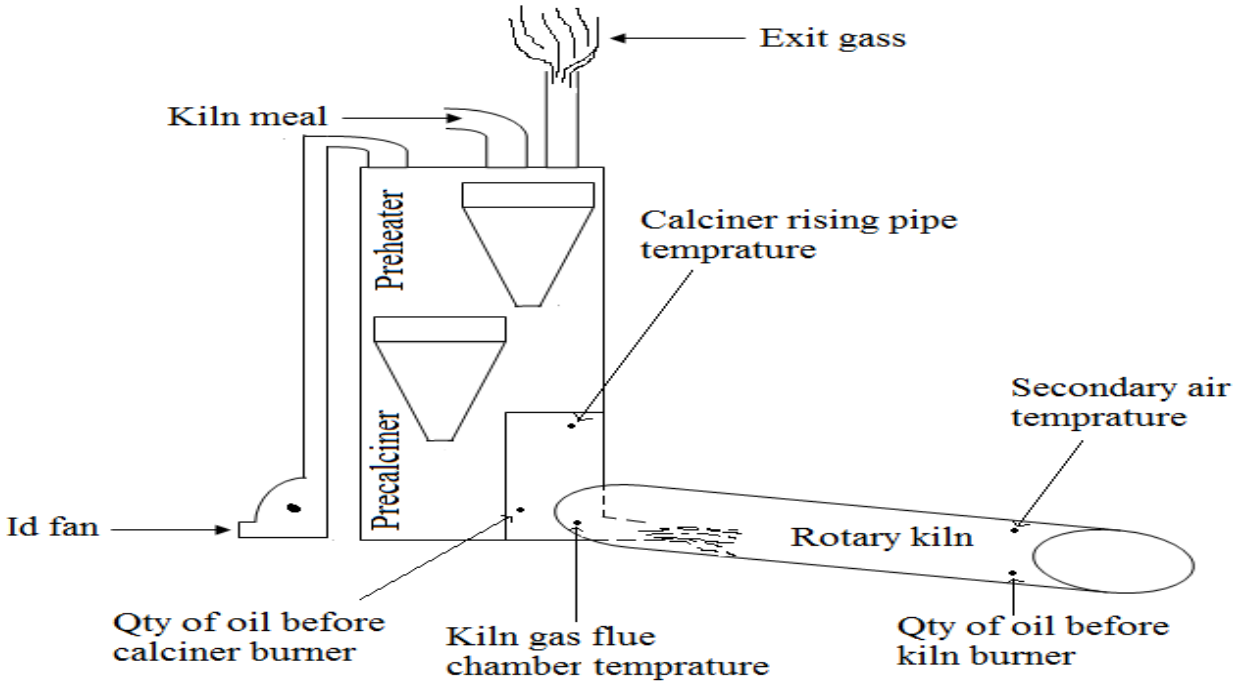


Figure 3.1.1: Variables points of actions

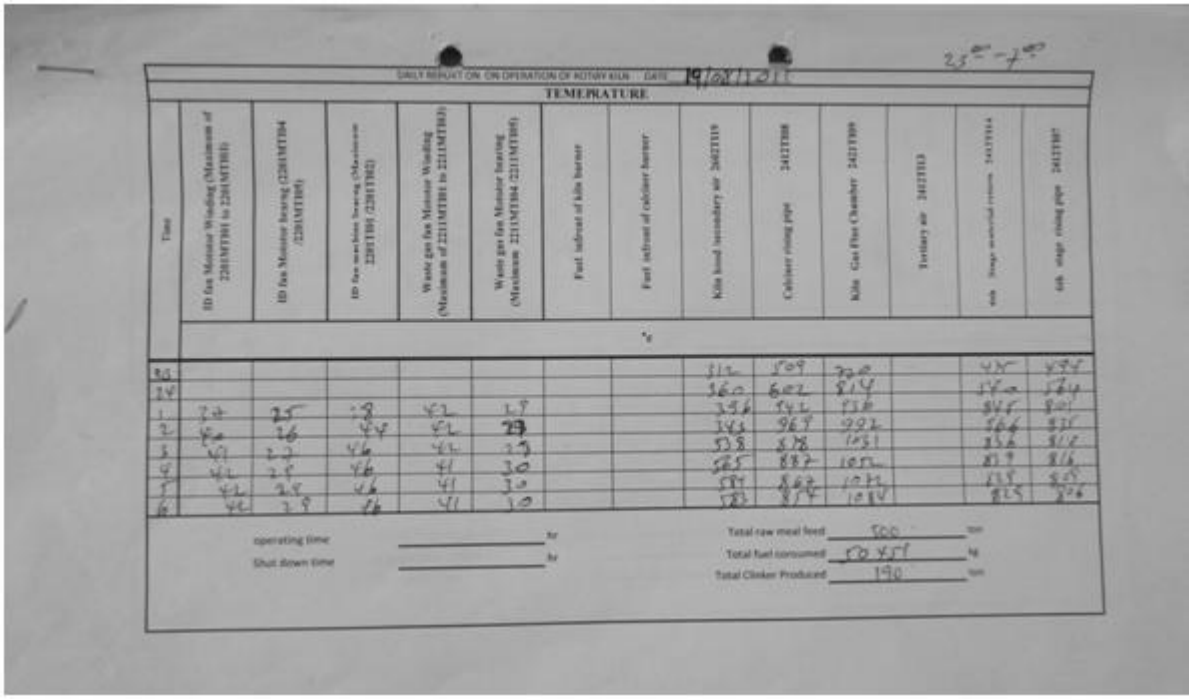
3.1.2 Collecting data for operating parameters, kiln meal and clinker variables

During the stay in Mugher cement factory historical process data is collected. This data corresponds to the operating parameters, kiln meal and clinker variables.

Collecting data for the operating parameters

Operating parameters are normally recorded every one hour on log sheets manually. Thus, I am forced to take pictures of more than one thousand (1338) pages using high resolution digital camera. This data corresponds to five month historical data. Then I have converted about one thousand (927) JPEG images, for ease of retrieving, in to PDF files using a software called Image to PDF Converter Free 5.0.

The figure below shows the picture of a typical log sheet (The picture of the log sheet is given in appendix VI for a better readability). On such sheets operating parameters are recorded manually.



DAILY REPORT ON OPERATION OF ROTARY KILN DATE 19/08/2011

TEMEPRATURE

Time	ID fan Motor Winding (Maximum of 2201MT301 to 2301MT301)	ID fan Motor bearing (2201MT304 /2301MT305)	ID fan motor bearing (Maximum 2201T301 /2301T302)	Waste gas fan Motor Winding (Maximum of 2211MT301 to 2311MT302)	Waste gas fan Motor bearing (Maximum 2211MT304 2311MT305)	Fuel inthroat of kiln burner	Fuel inthroat of calciner burner	Kiln hood secondary air 24127119	Calciner rising pipe 24127108	Kiln Gas Flow Chamber 24127109	Tertiary air 24127113	4th Stage secondary cyclone 24127114	4th stage rising pipe 24127107
22								512	509	720		435	494
24								360	602	814		564	564
1	24	25	38	42	29			396	542	836		845	801
2	40	26	44	42	29			343	969	992		566	835
3	41	27	46	41	29			538	878	1051		356	819
4	42	28	46	41	30			585	882	1052		859	816
5	42	28	46	41	30			584	868	1071		838	805
6	42	28	46	41	30			583	874	1084		845	806

operating time _____ hr

Shut down time _____ hr

Total raw meal feed 500 ton

Total fuel consumed 50 451 lit

Total Clinker Produced 190 ton

Figure 3.1.2: A picture of typical log sheet on which operating parameters are recorded

Collecting data for kiln meal and clinker variables

Collecting the kiln meal and clinker data is relatively easy for it is obtained in machine printed format. Kiln meal data is recorded every one hour and that of clinker is recorded every day.

3.1.2.1 The problem of dimensionality mismatch

Clinker data is measured every day and kiln meal and operating parameters are measured every one hour. Thus, for hundreds of clinker data samples there will be thousands of kiln meal and operating parameters data samples. This forced me to collect large historical data for few input – output sample pairs. Thus, the problem of dimensionality mismatch increased the data size.

3.1.3 Data encoding

The following has been done while encoding the hand written data of the operating parameters on the log sheets. The encoded data is given in the CD named 'auxiliary'.

- Non recorded values are marked by 'dash' symbol.
- Correction is made on data format of some log sheets.
- Some log sheets have two different dates. This problem is resolved by looking on the remark sheets.
- Difficult to read values are marked out to be replaced.
- Data sheets with completely unclear or not recorded dates are not encoded.

Since the kiln meal and clinker data are obtained in machine printed format, their conversion to soft copy is relatively easy compared to the operating parameters data. While converting the historical data in to soft copy Microsoft Office Excel 2007 is used as a spread sheet application.

3.2 Data preprocessing

The aim of this section is to produce cleaned historical data. The historical data is not ready to use for training due to the presence of missing values, outliers and others. Thus, the data collected from the industrial database is subjected to appropriate treatment.

3.2.1 Preliminary work

The historical data is not recorded in a manner suitable for further work. Besides, input-output dependence needs to be defined. Therefore, before going to missing value and outlier treatment preliminary work is done.

3.2.1.1 Correcting time format mismatch

Both the operating parameters and kiln meal and clinker variables are recorded on 24hr system. In the operational parameters' data sheet a new day begins in the morning at 7. However, in kiln meal and clinker data sheet a new day begins at 0 in the midnight. Thus, this mismatch is

corrected by converting the time format of the operating parameters to match that of the kiln meal and clinker. Figure 3.2.1 shows the conversion.

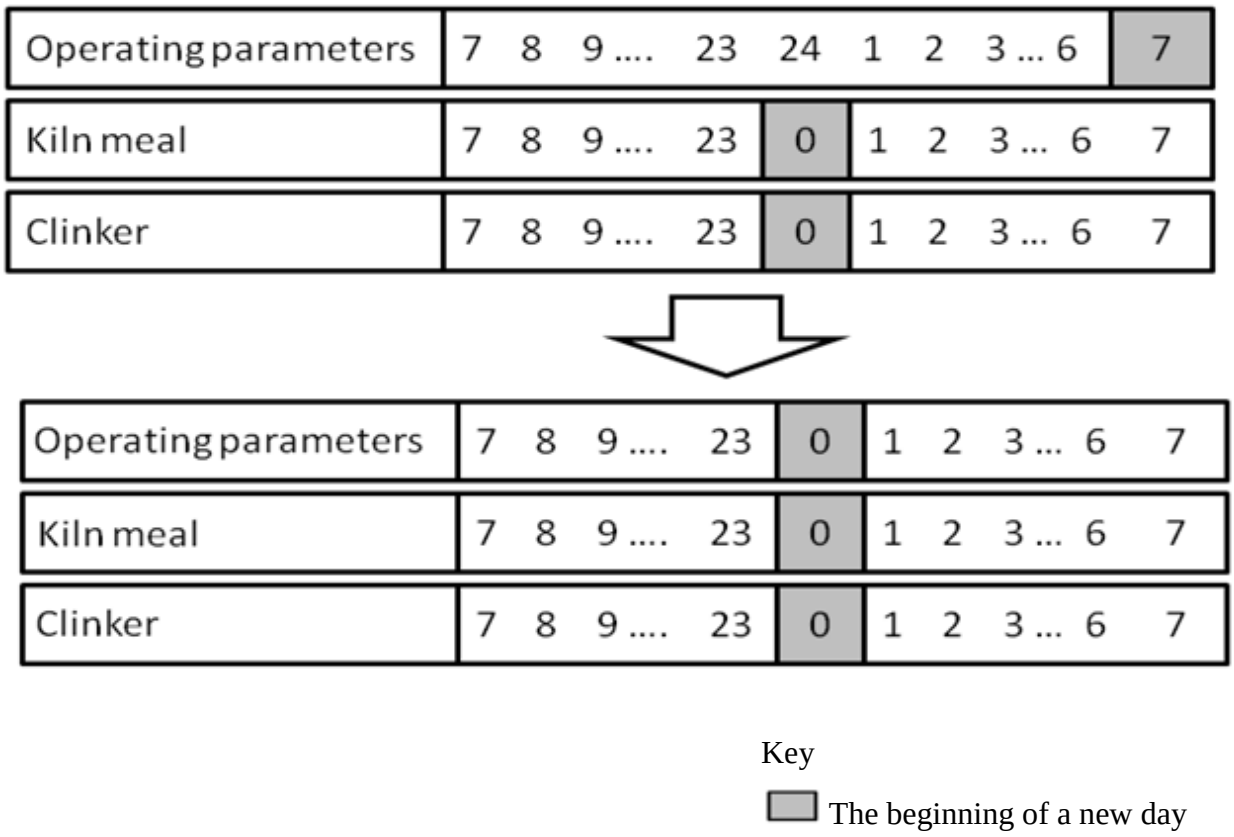


Figure 3.2.1: Time format conversion

3.2.1.2 Defining input-output time dependence

Clinker is analyzed once in 24hr, usually in the morning around 8. The analysis is done on a blended sample of many hours, usually of 24hr, where a sample is taken every 2hr. However, kiln meal and operating parameters are recorded every 1hr. Thus one set clinker analysis result depends on the average of many hours, usually 24hr, data of kiln meal and operating parameters. This time dependence is shown in figure 3.2.2 below.

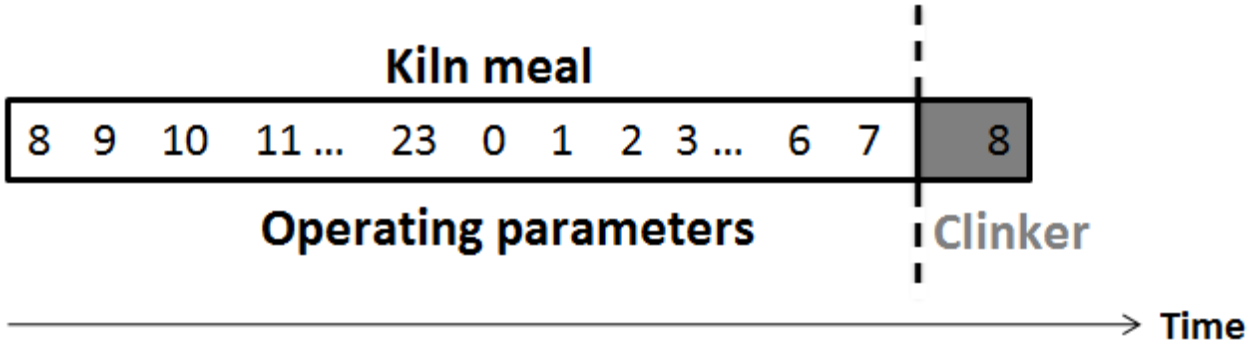


Figure 3.2.2: Input-output time dependence

3.2.2 Missing value imputation

Missing data corresponds to data values that should be present in a dataset but that, for various reasons, are absent [14]. It is treated using known mathematical expressions and linear interpolation.

3.2.2.1 Treating missing value using known mathematical expression

Values of some variables which are present but not recorded or difficult to read are replaced using known mathematical relationship. The variables are quantity of oil before kiln burner, quantity of oil before calciner burner and kiln meal. Thus, around 57 values are found to be missing and imputed for using eqs.3.2.1-3.2.3.

$$\text{Quantity of oil before kiln burner (Kg/hr)} = \frac{\Delta\text{fuel oil counter at kiln burner (Kg)}}{1hr} \quad \ldots\ldots\ldots \text{eq.3.2.1}$$

$$\text{Quantity of oil before calciner burner (Kg/hr)} = \frac{\Delta\text{fuel oil counter at calciner burner (Kg)}}{1hr} \quad \ldots \text{eq.3.2.2}$$

$$\text{Kiln meal flow rate (ton/hr)} = \frac{\Delta\text{kiln meal counter (ton)}}{1\text{hr}} \quad \ldots\ldots\ldots\ldots\ldots \text{eq.3.2.3}$$

3.2.2.2 Treating missing value using linear interpolation

While calculating daily averages of the input variables some day's values are absent. These values are imputed using linear interpolation. In this method the previous and next values are used to calculate the current value as shown by figure 3.2.3 and eq.3.2.4. As a result, around 14 daily average values are found to be missing and thus imputed for.

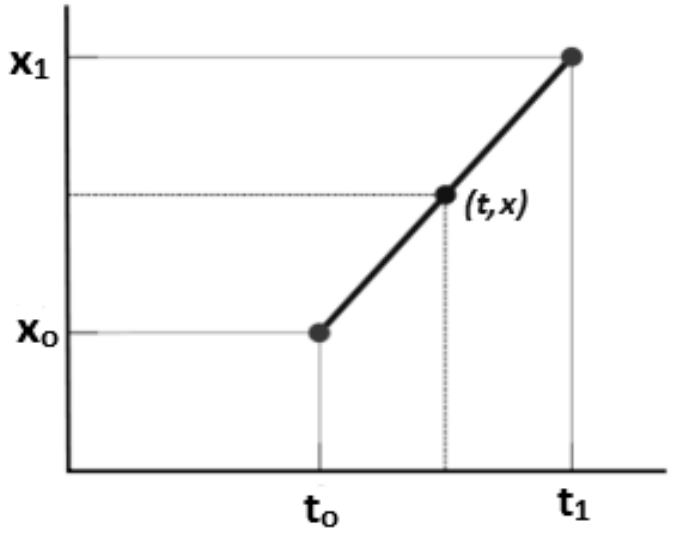


Figure 3.2.3: Linear interpolation

$$x = \frac{x_1 - x_o}{t_1 - t_o} t + x_o - \frac{x_1 - x_o}{t_1 - t_o} t_o \qquad \text{..... eq.3.2.4}$$

3.2.3 Outlier detection

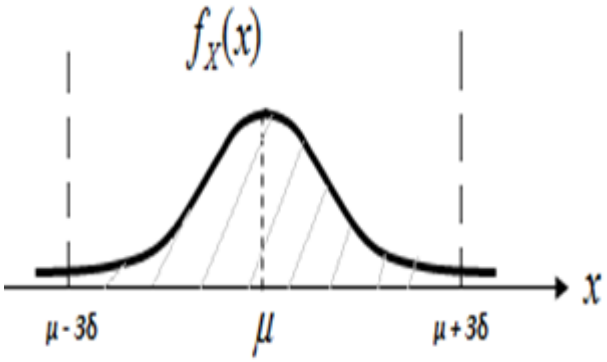
An outlier is an entry in a dataset that is anomalous with respect to the behavior seen in the majority of the other entries in the dataset [14]. Outliers can be distinguished into two types. Obvious outliers are those whose values do not satisfy the physical and technological limitations (e.g. a negative fuel rate). 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 [12].

Obvious outliers are detected using process knowledge. This results in detecting five obvious outliers. Once an obvious outlier is detected, it is removed and treated as a missing value. However, non obvious outliers are detected using the 3δ edit rule.

3.2.3.1 The 3δ edit rule

The 3δ edit rule is based on the hypothesis that the variables considered are normally distributed and it is generally applicable even though the normality condition is seldom satisfied [12, 15]. Accordingly, 99.73% of the data is within the 3δ region of the mean.



Thus, according to this rule all process values satisfying the condition :

$|x_i - \mu| > 3\delta$ are outliers

where :

μ : mean

δ : standard deviation

Figure 3.2.4: Normal distribution

Using this method, three justifiable outliers are detected. Then after they are removed and treated as a missing value. Finally, after all the preprocessing stages, forty-eight cleaned input-output dataset is prepared.

3.3 Data synthesizing

After preprocessing the data, forty-eight cleaned input-output dataset is obtained. However, the smaller the size of the data, the difficult will it be to train the neural network. Besides, small

dataset will not have enough variations to be representative. Therefore, a dataset is required to be synthesized systematically for there was no more data available to be collected.

To synthesis the data only one of the input attributes is varied while keeping all the rest fixed. This will avoid complications and gives large amount of data. The variables that are to be varied are kiln meal attributes, i.e. kiln meal oxides. To obtain process like data, the variation is limited within a sound possible range of variation. Based on the input data synthesized in this way, the output is generated by using mathematical expression. Furthermore, developed and known mathematical expressions are used along with MATLAB to accompany the synthesizing process.

3.3.1 Input data synthesizing strategy

Two methods are proposed to calculate the range in the kiln meal oxide variation. These methods are compared and the better one that comes out is used to synthesize the input data.

I. Conventional range

This method will generate input data set by using the constraint that the daily average kiln meal oxide composition should be within the maximum-minimum range of the hourly measured oxide composition of the kiln meal.

Advantage of the method:

- Easy mathematical manipulation.

Disadvantage of the method:

- The maximum-minimum range can be highly affected by incidental hourly oxide level drop or rise. Thus, making unrealistic large variation on the daily average oxide composition.

This method is described by giving an instance. As an example, input data generation on the basis of SiO_2 composition of the kiln meal is given as follows.

The SiO_2 content, just like the other oxides, can be used to generate an input dataset that is twice the size of the original input dataset. This is obtained by increasing and decreasing the SiO_2 level.

First, a one set of data can be synthesized by increasing SiO_2 level of the original input dataset. The value to be added is the difference between the maximum hourly oxide level and the maximum average daily oxide level. This makes the newly generated data not to violate the upper limit.

Second, another set of data can be synthesized by decreasing SiO_2 level of the original input dataset. The value to be deducted is the difference between the minimum average daily oxide level and the minimum hourly oxide level. This makes the newly generated data not to violate the lower limit.

Therefore a new dataset twice the original data size can be generated by adding and deducting values.

From 'kiln meal' and 'Input-Output' sheets of the excel file (saved in the CD named auxiliary),

Maximum SiO_2 (hourly basis) = 13.49 Maximum SiO_2 (daily average) =12.84

Minimum SiO_2 (hourly basis) = 10.41 Minimum SiO_2 (daily average) = 12.10

<u>Generating by adding a value</u>

The value to be added= Maximum SiO_2 (hourly basis) - Maximum SiO_2 (daily average)

= 13.49 – 12.84

= 0.65

By adding 0.65 to SiO_2 content of the kiln meal, given in the Input-Output sheet, a new dataset is generated. Table 3.3.1 shows the synthesized data.

Generating by deducting a value

The value to be deducted = Minimum SiO_2 (daily average) - Minimum SiO_2 (hourly basis)

$$= 12.10 - 10.41$$

$$= 1.69$$

By subtracting 1.69 to SiO_2 content of the kiln meal, given in the Input-Output sheet, a data set is generated. Table 3.3.1 shows the synthesized data.

kiln meal (daily average)			
seq. no.	SiO2 (original)	SiO2 (synthesized by adding)	SiO2 (synthesized by deducting)
1	12.53	13.18	10.84
2	12.54	13.19	10.85
3	12.54	13.19	10.85
4	12.44	13.09	10.75
5	12.52	13.17	10.83
6	12.62	13.27	10.93
7	12.69	13.34	11.00
8	12.84	13.49	11.15
9	12.57	13.22	10.88
10	12.38	13.03	10.69
11	12.80	13.45	11.11
12	12.61	13.26	10.92
13	12.64	13.29	10.95
14	12.41	13.06	10.72
15	12.48	13.13	10.79
16	12.34	12.99	10.65
17	12.53	13.18	10.84
18	12.55	13.20	10.86
19	12.59	13.24	10.90
20	12.60	13.25	10.91
21	12.69	13.34	11.00
22	12.63	13.28	10.94
23	12.69	13.34	11.00
24	12.69	13.34	11.00
25	12.78	13.43	11.09
26	12.56	13.21	10.87
27	12.47	13.12	10.78
28	12.59	13.24	10.90
29	12.47	13.12	10.78
30	12.51	13.16	10.82
31	12.66	13.31	10.97
32	12.57	13.22	10.88
33	12.40	13.05	10.71
34	12.66	13.31	10.97
35	12.73	13.38	11.04
36	12.72	13.37	11.03
37	12.44	13.09	10.75
38	12.41	13.06	10.72
39	12.44	13.09	10.75
40	12.56	13.21	10.87
41	12.31	12.96	10.62
42	12.29	12.94	10.60
43	12.40	13.05	10.71
44	12.10	12.75	10.41
45	12.35	13.00	10.66
46	12.40	13.05	10.71
47	12.65	13.30	10.96
48	12.44	13.09	10.75
average	12.54	13.19	10.85

Table 3.3.1: Original and synthesized data for SiO_2 composition of kiln meal using conventional range.

II. Range calculated from average values

This method will generate input data set by using the constraint that the daily average kiln meal oxide composition should be within the range (calculated from average values) of the hourly kiln meal oxide content.

Advantage of the method

- Avoids the 'range' from being highly affected by incidental hourly oxide level drop or rise. As a result, provides better realistic input dataset variation.

Disadvantage of the method

- Relative mathematical complexity.

The range is calculated by using average of higher values of the hourly basis oxide level as the upper limit and average of lower values of the hourly basis oxide level as the lower limit. The percentage of the dataset (hourly basis) that is used to calculate the limits is determined by considering the fact that the original daily kiln meal oxide composition is highly precise.

An example on input data generation on the basis of SiO_2 composition of the kiln meal is given below. In this example two percent averaging and five percent averaging are used for comparison.

For the two percent case, the top 2% of the hourly basis SiO_2 level and the lower 2% are used to calculate the upper and lower limits respectively. Since a total of 694 samples are available, see the 'kiln meal' sheet, two percent of it means fourteen samples. Therefore, the top fourteen and the lower fourteen samples are taken. Averages of these values give upper and lower limits. Thus,

Upper limit of SiO_2 = average of the top fourteen = 13.23

Lower limit of SiO_2 = average of the lower fourteen = 11.83

Maximum SiO_2 (daily average) =12.84

Minimum SiO_2 (daily average) = 12.10

Generating by adding a value

The value to be added= Upper limit of SiO_2 - Maximum SiO_2 (daily average)

= 13.23– 12.84

= 0.39

By adding 0.39 to SiO_2 content of the kiln meal, given in the 'Input-Output' sheet, a data set is generated. Table 3.3.2 shows the synthesized data.

Generating by deducting a value

The value to be deducted = Minimum SiO_2 (daily average) - Lower limit of SiO_2

= 12.10 – 11.83

= 0.27

By subtracting 0.27 to SiO_2 content of the kiln meal, given in the 'Input-Output' sheet, a data set is generated. Table 3.3.2 shows the synthesized data.

For the five percent case, the top 5% of the hourly basis SiO_2 level and the lower 5% are used to calculate the upper and lower limits respectively. Since a total of 694 samples are available, five percent of it means thirty five samples. Therefore, the top thirty five and the lower thirty five samples are taken. Averages of these values give upper and lower limits. Thus,

Upper limit of SiO_2 = average of the top thirty five = 13.11
Lower limit of SiO_2 = average of the lower thirty five = 12.00
Maximum SiO_2 (daily average) =12.84
Minimum SiO_2 (daily average) = 12.10

Like the two percent case, the value to be added and subtracted is calculated. The values are 0.27 and 0.1 respectively. Table 3.3.3 shows the synthesized data for five percent case.

kiln meal (daily average)			
seq. no.	SiO2	SiO2 (synthesized by adding)	SiO2 (synthesized by deducting)
1	12.53	12.92	12.26
2	12.54	12.93	12.27
3	12.54	12.93	12.27
4	12.44	12.83	12.17
5	12.52	12.91	12.25
6	12.62	13.01	12.35
7	12.69	13.08	12.42
8	12.84	13.23	12.57
9	12.57	12.96	12.30
10	12.38	12.77	12.11
11	12.80	13.19	12.53
12	12.61	13.00	12.34
13	12.64	13.03	12.37
14	12.41	12.80	12.14
15	12.48	12.87	12.21
16	12.34	12.73	12.07
17	12.53	12.92	12.26
18	12.55	12.94	12.28
19	12.59	12.98	12.32
20	12.60	12.99	12.33
21	12.69	13.08	12.42
22	12.63	13.02	12.36
23	12.69	13.08	12.42
24	12.69	13.08	12.42
25	12.78	13.17	12.51
26	12.56	12.95	12.29
27	12.47	12.86	12.20
28	12.59	12.98	12.32
29	12.47	12.86	12.20
30	12.51	12.90	12.24
31	12.66	13.05	12.39
32	12.57	12.96	12.30
33	12.40	12.79	12.13

kiln meal (daily average)			
seq. no.	SiO2	SiO2 (synthesized by adding)	SiO2 (synthesized by deducting)
34	12.66	13.05	12.39
35	12.73	13.12	12.46
36	12.72	13.11	12.45
37	12.44	12.83	12.17
38	12.41	12.80	12.14
39	12.44	12.83	12.17
40	12.56	12.95	12.29
41	12.31	12.70	12.04
42	12.29	12.68	12.02
43	12.40	12.79	12.13
44	12.10	12.49	11.83
45	12.35	12.74	12.08
46	12.40	12.79	12.13
47	12.65	13.04	12.38
48	12.44	12.83	12.17
average	12.54	12.93	12.27

Table 3.3.2: Original and synthesized data for SiO_2 composition of kiln meal (average of 2%)

kiln meal (daily average)			
seq. no.	SiO2	SiO2 (synthesized by adding)	SiO2 (synthesized by deducting)
1	12.53	12.80	12.43
2	12.54	12.81	12.44
3	12.54	12.81	12.44
4	12.44	12.71	12.34
5	12.52	12.79	12.42
6	12.62	12.89	12.52
7	12.69	12.96	12.59
8	12.84	13.11	12.74
9	12.57	12.84	12.47
10	12.38	12.65	12.28
11	12.80	13.07	12.70
12	12.61	12.88	12.51
13	12.64	12.91	12.54
14	12.41	12.68	12.31
15	12.48	12.75	12.38
16	12.34	12.61	12.24
17	12.53	12.80	12.43
18	12.55	12.82	12.45
19	12.59	12.86	12.49
20	12.60	12.87	12.50
21	12.69	12.96	12.59
22	12.63	12.90	12.53
23	12.69	12.96	12.59
24	12.69	12.96	12.59
25	12.78	13.05	12.68
26	12.56	12.83	12.46
27	12.47	12.74	12.37
28	12.59	12.86	12.49
29	12.47	12.74	12.37
30	12.51	12.78	12.41
31	12.66	12.93	12.56
32	12.57	12.84	12.47

kiln meal (daily average)			
seq. no.	SiO2	SiO2 (synthesized by adding)	SiO2 (synthesized by deducting)
33	12.40	12.67	12.30
34	12.66	12.93	12.56
35	12.73	13.00	12.63
36	12.72	12.99	12.62
37	12.44	12.71	12.34
38	12.41	12.68	12.31
39	12.44	12.71	12.34
40	12.56	12.83	12.46
41	12.31	12.58	12.21
42	12.29	12.56	12.19
43	12.40	12.67	12.30
44	12.10	12.37	12.00
45	12.35	12.62	12.25
46	12.40	12.67	12.30
47	12.65	12.92	12.55
48	12.44	12.71	12.34
Average	12.54	12.81	12.44

Table 3.3.3: Original and synthesized data for SiO_2 composition of kiln meal (average of 5%)

It can be observed from table 3.3.4 below that the average of the data generated by the conventional range method deviates by large value from the average of the original data when it is compared with method II (range calculated from average value). Even with in this method the five percent case gives a better variation with a smaller deviation than the two percent case.

Therefore, method II will be used to generate input data from oxides of the kiln meal by fixing the rest input attributes.

		(method I) Conventional range	Method II (range calculated from average value)	
			2%	5%
Deviation of mean value ($\lvert\bar{u}_o - \bar{u}_{sy}\rvert$)	Generating by adding a value	0.65	0.39	0.27
	Generating by deducting a value	1.69	0.27	0.1

Where

$\bar{u}_o$: Mean of the original data

$\bar{u}_{sy}$: Mean of the synthesized data

Table 3.3.4: Comparison of input data synthesizing methods

3.3.2 Output data synthesizing strategy

The outputs, i.e. clinker parameters, are calculated from measured oxides of the clinker. The outputs are calculated as follows.

$$\text{Lime Saturation factor (LSF)} = \frac{100\ \text{CaO}}{2.8\ \text{SiO}_2\ + 1.18\ \text{Al}_2\text{O}_3\ + 0.65\text{Fe}_2\text{O}_3} \quad \ldots\ldots\ldots \text{eq. 3.3.1}$$

$$\text{Silica Module (SM)} = \frac{\text{SiO}_2}{\text{Al}_2\text{O}_3\ +\ \text{Fe}_2\text{O}_3} \quad \ldots\ldots\ldots \text{eq. 3.3.2}$$

$$\text{Alumina Module (AM)} = \frac{\text{Al}_2\text{O}_3}{\text{Fe}_2\text{O}_3} \quad \ldots\ldots\ldots \text{eq. 3.3.3}$$

$$C3S = 4.07\left(\text{CaO} - \text{Free CaO}\right) - 7.6\ \text{SiO}_2 - 1.43\text{Fe}_2\text{O}_3 - 6.7\ \text{Al}_2\text{O}_3 - 2.85\ \text{SO}_3 \quad \ldots \text{eq. 3.3.4}$$

As can be seen from the above formulas the outputs are calculated from oxides. Thus, if the oxides values are obtained, the outputs can be calculated. Many of the oxides values can be obtained (approximately) from the fact that while the meal is burned some percentage of its mass is lost. Therefore, from the knowledge of this loss called loss on ignition (LOI) the clinker oxides, i.e. CaO, SiO_2, Fe_2O_3 and Al_2O_3, can be approximately calculated from kiln meal oxides by using eqs.3.3.5-3.3.8.

Up on an expert's recommendation from the factory, LOI have average value of 35%. This value can be used. However, LOI calculated from the original data using eqs.3.3.5-3.3.8 will give more process like LOI. Table 3.3.5 shows the average values of LOI, whose calculation is shown in the 'Input-Output (temp.)' sheet of the excel file (given in the CD named auxiliary).

	SiO_2	**Al_2O_3**	**Fe_2O_3**	**CaO**
LOI (average)	39.36	29.95	35.31	35.14

Table 3.3.5: Average loss on ignition

$$\%\text{CaO in Clinker} = \frac{(\%\ \text{CaO in kiln meal}\ )\ 100}{100 - \text{LOI}} \quad \ldots\ldots\ldots \text{eq. 3.3.5}$$

$$\%\ \text{SiO}_2\ \text{in Clinker} = \frac{(\ \%\ \text{SiO}_2\ \text{kiln meal}\ )\ 100}{100 - \text{LOI}} \quad \ldots\ldots\ldots\ldots\ eq.\ 3.3.6$$

$$\%\ \text{Al}_2\text{O}_3\ \text{in Clinker} = \frac{(\ \%\ \text{Al}_2\text{O}_3\ \text{kiln meal}\ )\ 100}{100 - \text{LOI}} \quad \ldots\ldots\ldots\ldots \text{eq. 3.3.7}$$

$$\%\ \text{Fe}_2\text{O}_3\ \text{in Clinker} = \frac{(\ \%\ \text{Fe}_2\text{O}_3\ \text{kiln meal}\ )\ 100}{100 - \text{LOI}} \quad \ldots\ldots\ldots\ldots \text{eq. 3.3.8}$$

As can be seen from eq.3.3.4, the calculation of C_3S is involved with free CaO and SO_3 values, which cannot be obtained from kiln meal composition by using LOI. Thus, multiple regression is used to fit a model on the available original data. The model is of the form:

$C_3S = a_0 + a_1CaO + a_2SiO_2 + a_3Al_2O_3 + a_4Fe_2O_3$ …….. eq.3.3.9

Multiple regression solves for unknown coefficients a_0, a_1, a_2, a_3 and a_4 from the original clinker data given in the 'Input-Output (temp.)' sheet. After importing the data to MATLAB workspace, the following code will give the coefficients.

```
>> % MATLAB code to do multiple regression
>> x=[ones(size(CaO)) CaO SiO2 Al2O3 Fe2O3]; % x is the design matrix
>> a=x\C3S % a is the vector of the coefficients

a =

  -38.2749
    4.3795
   -7.0331
   -8.0014
    1.0512
```

Thus, the model becomes

C_3S= -38.2749+ 4.3795CaO -7.0331SiO_2 -8.0014Al_2O_3 + 1.0512Fe_2O_3 ……. eq.3.3.10

The maximum of the absolute value of the deviation of the data from the model is used to validate the model. The following MATLAB code gives the maximum of the absolute value of the deviation.

```
>> % MATLAB code to calculate maximum of the absolute value of the deviation
>> y=x*a; % y is data from the model
>> MaxErr = max(abs(y - C3S))% MaxErr is maximum of the absolute value of the
                              % deviation

MaxErr =

    1.0829
```

This value is much smaller than any of the data values, indicating that this model accurately follows the data.

The maximum of the absolute value of the deviation, using average values of free CaO and SO_3 to be 1.5 and 1.53 - obtained from clinker data with guidance from an expert in the factory - respectively, is 6.0267. Thus, the multiple regression model is better than the Bogue calculation used with proposed values. Therefore, the multiple regression model will be used to find C_3S values from generated clinker oxide values.

3.3.3 Generating the input-output dataset

Using the data synthesis strategy discussed thus far, input-output dataset is generated. Thus, in this section the generated input and corresponding outputs are given. Furthermore, how specifically the data is synthesized is discussed.

I. Input data

Out of the total twelve input attributes, four of them will be used to synthesize new input dataset and method II (range calculated from average value) will be employed. If this method is used as in the case of the example given (given in section 3.3.1), each kiln meal oxide will generate dataset of twice the size of original dataset, which is forty eight. This could make the input-output dataset to be dominated by artificial features. Thus, only half randomly selected input dataset will be used as a basis while generating. Then, either by adding or deducting a value new dataset is generated. In this way, 144 samples of input-output dataset can be obtained.

Generating from kiln meal SiO_2

Half of the kiln meal SiO_2 data is randomly selected using the MATLAB built in function dividerand. The following MATLAB code does the random selection after the data is imported to MATLAB workspace.

```
>> % MATLAB code to randomly dividing the kiln meal SiO2
>> x=SiO2';
>> [selcV,valV,restV,selcInd,valInd,restInd]= dividerand(x,0.5,0,0.5);
>> selcInd % gives the indices of the random selection

selcInd =

  Columns 1 through 18

     1     3     5     7     9    10    12    17    18    19    22    25
26    27    30    32    37    39

  Columns 19 through 24

    40    42    43    45    47    48
```

Generating by adding a value (the 5% case) is used since majority of the original data is above average.

Upper limit of SiO_2 = average of the top thirty five = 13.09

Maximum SiO_2 (daily average) =12.84

The value to be added = Upper limit of SiO_2 - Maximum SiO_2 (daily average)

= 13.09-12.84

= 0.25.

Adding this value to the randomly selected dataset gives 24 samples of input data shown in the 'Input-Output (synthesized)' sheet of the excel file given in the CD named auxiliary.

Generating from kiln meal Al_2O_3

Similar to the SiO_2 case the kiln meal Al_2O_3data is selected by writing a MATLAB code. The code is the following:

```
>> % MATLAB code to randomly dividing the kiln meal Al2O3
>> x=Al2O3';
>> [selcV,valV,restV,selcInd,valInd,restInd]= dividerand(x,0.5,0,0.5);
>> selcInd % gives the indices of the random selection

selcInd =

  Columns 1 through 18

     3     4     5     7    12    17    19    21    22    23    26    27
28    29    30    31    35    36

  Columns 19 through 24
    37    41    42    43    46    48
```

Generating by deducting a value (the 5% case) is used since majority of the original data is less than or equal to the average.

Lower limit of Al_2O_3 = average of the lower thirty five = 4.09

Minimum Al_2O_3 (daily average) = 4.08

The value to be deducted = Minimum Al_2O_3 (daily average) - Lower limit of Al_2O_3

= 4.08 – 4.09

= -0.01

Since the value is negative, the lower limit is not actually a limit. Thus, 2% case will be used. For this case:

Lower limit of Al_2O_3 = average of the lower fourteen= 4.03

Minimum Al_2O_3 (daily average) = 4.08

The value to be deducted = Minimum Al_2O_3 (daily average) - Lower limit of Al_2O_3

= 4.08 – 4.03

= 0.05

Subtracting this value to the randomly selected dataset gives 24 samples of input data shown in 'Input-Output (synthesized)' sheet.

Generating from kiln meal Fe_2O_3

The MATLAB code for random selection of kiln meal Fe_2O_3 data is given below.

```
>> % MATLAB code to randomly dividing the kiln meal Fe2O3
>> x=Fe2O3';
>> [selcV,valV,restV,selcInd,valInd,restInd]= dividerand(x,0.5,0,0.5);
>> selcInd % gives the indices of the random selection

selcInd =

  Columns 1 through 18

     3     5     6     7     8    11    14    16    17    22    28    30
32    33    34    35    38    40

  Columns 19 through 24

    41    42    45    46    47    48
```

Generating by adding a value (the 5% case) is used since majority of the original data is above average.

Upper limit of Fe_2O_3 = average of the top thirty five = 3.07

Maximum Fe_2O_3 (daily average) = 2.98

The value to be added = Upper limit of Fe_2O_3 - Maximum Fe_2O_3 (daily average)

= 3.07 - 2.98

= 0.09

Adding this value to the randomly selected dataset gives 24 samples of input data shown in 'Input-Output (synthesized)' sheet.

Generating from kiln meal CaO

Similar to the SiO_2, Al_2O_3 and Fe_2O_3 cases, the kiln meal CaO data is also randomly selected. The code used for selection is given below.

```
>> % Matlab code to randomly dividing the kiln meal CaO
>> x=CaO';
>> [selcV,valV,restV,selcInd,valInd,restInd]= dividerand(x,0.5,0,0.5);
>> selcInd % gives the indices of the random selection

selcInd =

  Columns 1 through 18

     3     6     7     8    10    12    14    15    20    21    22    24
26    28    29    30    32    33
  Columns 19 through 24

    38    41    43    44    45    48
```

Generating by adding a value (the 5% case) is used since majority of the original data is above average.

Upper limit of CaO = average of the top thirty five = 43.12

Maximum CaO (daily average) = 42.9

The value to be added = Upper limit of CaO – Maximum CaO (daily average)

= 43.12 – 42.9

= 0.22

Adding this value to the randomly selected dataset gives 24 samples of the input data shown in Input-Output (synthesized) sheet. Therefore, input dataset given in the sheet has a total of 144 samples.

II. Output data

Whenever the kiln meal SiO_2 is used to generate input data, the clinker SiO_2 composition is obtained using eq.3.3.12. The remaining clinker oxides (i.e. Al_2O_3, Fe_2O_3 and CaO) take their original value since only one parameter (SiO_2) is varied while the others being constant. When the other kiln meal oxides (i.e. Al_2O_3, Fe_2O_3 and CaO) are used to generate input data, procedure similar to the SiO_2 case is followed.

After this the four clinker oxides, corresponding to the input data, are completely obtained. The output data is generated using eqs.3.3.1-3.3.3 and eq.3.3.10. The 'Input-Output (synthesized)' sheet gives the input-output dataset prepared for training with intermediate values.

To obtain clinker oxides from kiln meal oxides use the following equations.

$$\%\text{CaO in Clinker} = \frac{(\%\ \text{Cao in kiln meal})\ 100}{100 - 35.14} \quad \ldots\ldots\ldots \text{eq. 3.3.11}$$

$$\%\ SiO_2 \text{ in Clinker} = \frac{(\%\ SiO_2 \text{ kiln meal})\ 100}{100 - 39.36} \quad \ldots\ldots\ldots \text{eq. 3.3.12}$$

$$\%\ Al_2O_3 \text{ in Clinker} = \frac{(\%\ Al_2O_3 \text{ kiln meal})\ 100}{100 - 29.95} \quad \ldots\ldots\ldots \text{eq. 3.3.13}$$

$$\%\ Fe_2O_3 \text{ in Clinker} = \frac{(\%\ Fe_2O_3 \text{ kiln meal})\ 100}{100 - 35.31} \quad \ldots\ldots\ldots \text{eq. 3.3.14}$$

Thus, finally 144 input-output dataset are prepared using the data synthesis strategy. This dataset will be used to train neural networks. The next section discusses neural network model selection and training.

3.4 Neural network model selection and training

Biological neural systems are the original inspiration for artificial neural networks, which have been developed as generalization of the mathematical models of human cognition or neural biology [21,22].

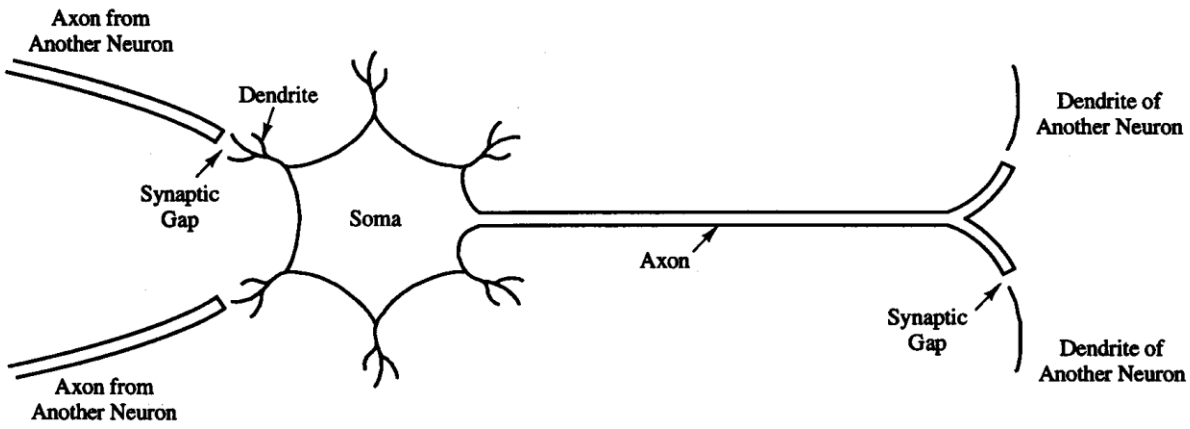


Figure 3.4.1: Biological neuron

A neuron with a single scalar input and bias appears on the figure 3.4.2 below. The scalar input *p* is transmitted through a connection that multiplies its strength by the scalar weight *w* to form the product *wp*, again a scalar. The scalar bias, *b* is simply added to the product *wp* to form the argument of the transfer function *f*, which produce the scalar output a. Two or more of such neurons shown can be combined in a layer, and a particular network could contain one or more such layers.

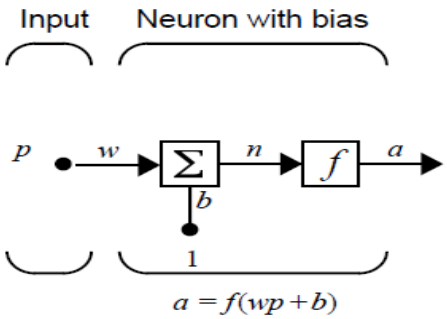


Figure 3.4.2: A neuron with scalar input and bias

A single layer neural network and its abbreviated notation are shown in figure 3.4.3 below. Without this abbreviated notation there will be so much detail that the main thoughts tend to be lost. It is needed to make a distinction between weight matrices that are connected to inputs and weight matrices that are connected between layers to describe networks having multiple layers. Weight matrices connected to inputs are called input weights (**IW**) and weight matrices coming from layer outputs are called layer weights (**LW**). Furthermore, superscripts are used to identify the source (second index) and the destination (first index) for the various weights and other elements of the network. Thus, this abbreviated notation is used to depict the neural networks that are used to model the input-output relationship.

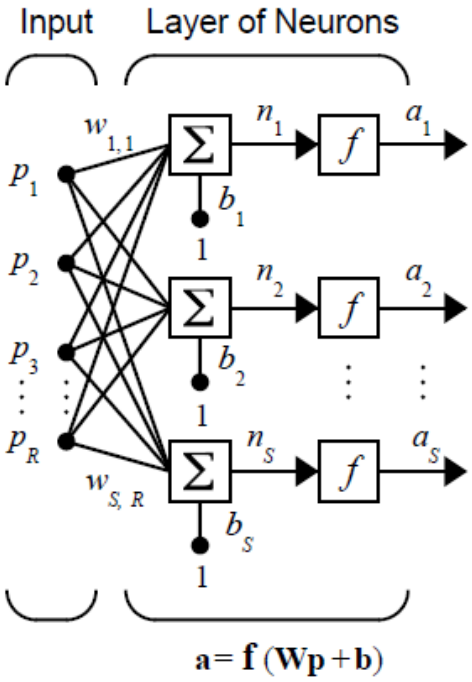


Figure 3.4.3 (a)

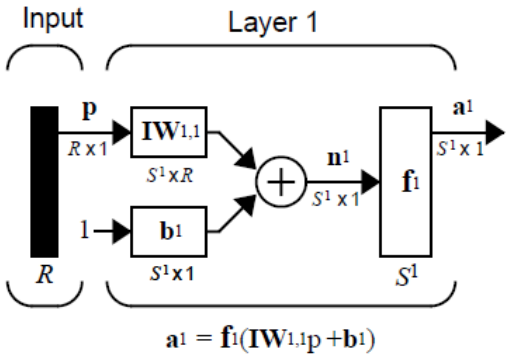


Figure 3.4.3 (b)

Where R is the number of elements in the input vector and S/S^1 is the number of neurons in the layer.

Figure 3.4.3(a): A single layer network and (b): A single layer network with abbreviated network notation

3.4.1 Network model, architectures and features

Neural network models are selected to be compared with a proposed benchmark. The benchmark network is multiple input multiple output (MIMO) feedforward neural network. Using this benchmark two models are selected. The first one is multiple input single output (MISO) feedforward neural network. For ease of predicting single output per network than multiples, it is expected that the model will give better result. The second one is modified Elman network. This model is chosen because it is derived from one of the most known neural network, its dynamic nature could fit the delay feature of the kiln system and its strength for complex system modeling is indicated in some papers [10,18,19]. The model architectures in abbreviated notation are shown in figures 3.4.4-3.4.6 below.

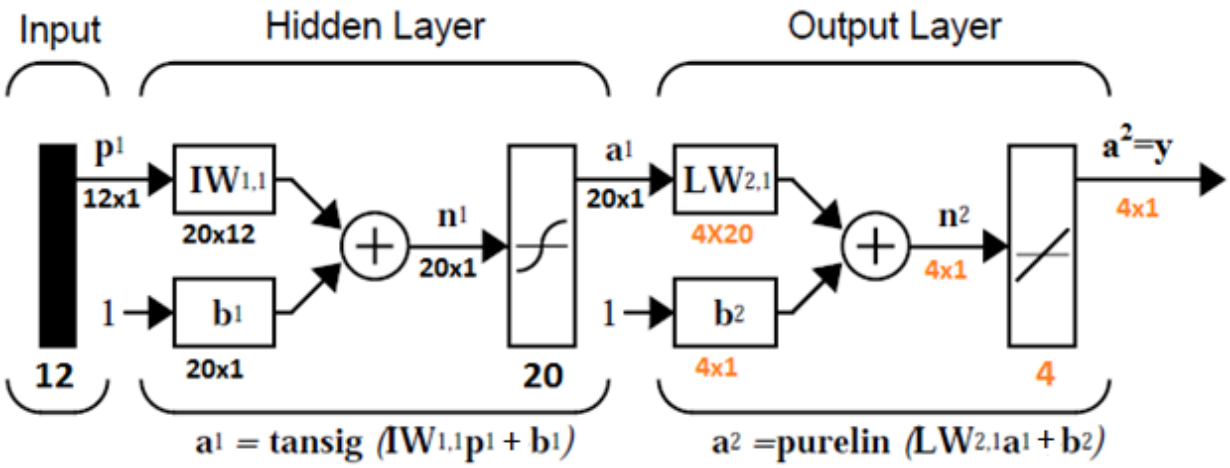


Figure 3.4.4: The benchmark network (multiple input multiple output feedforward network: 12-20-4)

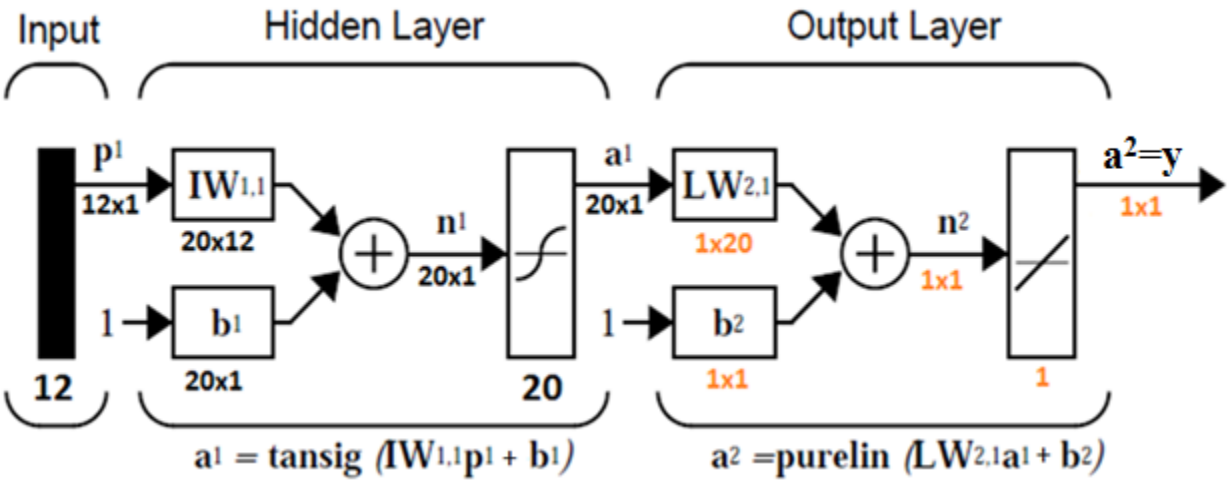


Figure 3.4.5: Multiple input single output feedforward network: 12-20-1

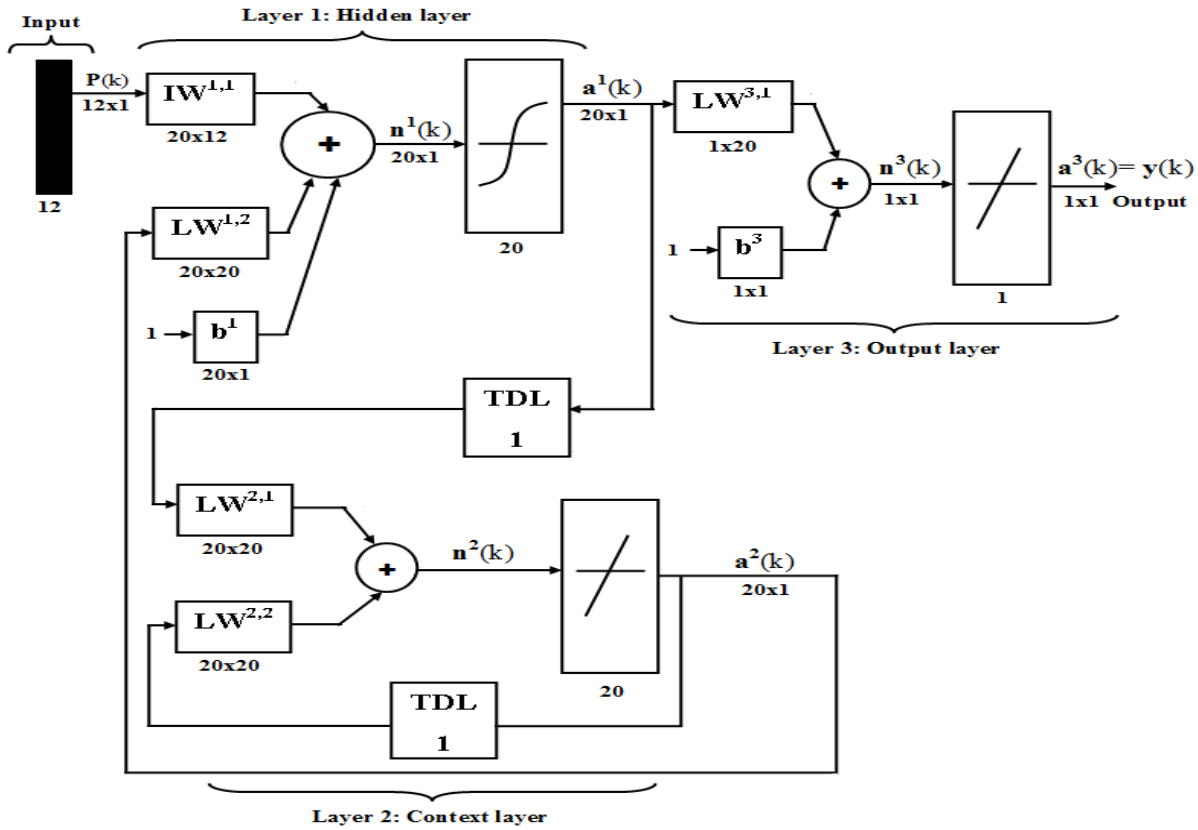


Figure 3.4.6: The modified Elman network [10]

All of the network models share common features on normalization, number of hidden layer neurons, performance function, training algorithm, generalization and training stopping criteria.

Normalization:

It is important for the data to be normalized. Otherwise important process variables having small magnitudes will be overshadowed by less important variables having larger magnitudes. Because of this the MATLAB normalization function 'mapminmax' is used to scale, as shown in eq.3.4.1 below, inputs and targets so that they fall in the range [0,1] before a network uses them. It is then applied in reverse to layer output values before being returned as network output values. Thus, the normalization effectively becomes a part of the network, just like the network weights and biases. As a result, the error values in this report are not normalized rather actual. All this makes the training more efficient.

$$y = \frac{(y\max - y\min) * (x - x\min)}{x\max - x\min} + y\min \quad \text{.....eq.3.4.1}$$

Where
x: the original data
y: the normalized data

Number of hidden layer neurons:

There is no a defined rule to follow while specifying the number of hidden layer neurons. It is determined after a trial and error procedure to refine a rough selection made by using a rule of thumb. The employed rule of thumb says the hidden layer neuron should not be greater than twice the number of inputs. Thus, it is settled that the hidden layer to have twenty neurons. In the case of modified Elman network the number of neurons in the context layer is twenty because the hidden and context layers have the same size.

Performance function:

All of the network models use mean square error (mse) as a performance function.

$$mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \quad \text{.....eq.3.4.2}$$

Where

t_i: target output

a_i: network output

N: training data sample size

Generalization:

One of the problems that occur during neural network training is called overfitting. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations. Early stopping is the technique that is used to avoid this problem.

In this technique of early stopping, the available data is randomly divided in to three subsets. The first subset is the training set, which is used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error normally decreases during the initial phase of training, as does the training set error. However, when the network begins to overfit the data, the error on the validation set typically begins to rise. When the validation error increases for more than a specified number of iterations (the default value is used which is five iterations), the training is stopped, and the weights and biases at the minimum of the validation error are returned. The third subset is the test set. This subset is not used during training, but it is used to compare different models. It is also useful to plot the test set error during the training process.

Therefore, early stopping improves generalization by randomly dividing the data for different purposes. This technique stops neural network training for network not to further learn training examples avoiding new situations.

Training function:

The Matlab training function 'trainscg' is used to train all of the network models. It is recommended to be used with networks that have limited data and use early stopping to avoid overfitting. This function, 'trainscg' is the scaled conjugate gradient training algorithm (SCG) [23].

Training stopping criteria:

The networks are trained according to the following values of training parameters.

Maximum number of epochs to train = 1000
Performance goal (mse) = 0
Maximum time to train in seconds = ∞
Minimum performance gradient = 1e-6
Maximum validation failures = 5

Training stops when any of the following conditions occur:
1) The maximum number of epochs (repetitions) is reached.
2) The maximum amount of time has been exceeded.
3) Performance has been minimized to the goal.
4) The performance gradient falls below the minimum.
5) Validation performance has increased more than the maximum validation failures.

CHAPTER 4

RESULT AND DISCUSSION

4.1 The neural network models result

The MATLAB Neural Network Toolbox is used to create, train and test the neural network models. This is possible because in the toolbox networks have an object-oriented representation. The representation allows defining various architectures even complicated ones.

As a means of observing model performance, network output (predicted) versus target output is plotted for the training and testing sets. This plot gives visual evaluation of how good the prediction is. Besides, change of network errors from iteration to iteration and the iteration of best network result is given by a plot labeled as validation performance. Furthermore, training parameters and mean square errors (actual not normalized) are reported in a tabular form.

In the target versus predicted plot, target values are line connected and predicted values are shown as point values. The better the prediction, the better the predicted value will follow that of the target values. This can be easily inspected by visual means. The validation performance plot shows how the network model performs to avoid overfitting while being trained. After the network begins to train, the error on the training as well as the validation set will decrease. However; when the network begins to overfit, the error on the validation set begin to increase. If the increment is for more than five iterations, the network at the start of rising error value is returned. This network is the network of best validation performance having weight and bias value at the iteration the network is returned.

The MATLAB code for the benchmark, MISO and modified Elman network is given in the appendix. As the codes for each sub models of the MISO are the same except the data used, a representative code is given. The same is done to the modified Elman. Thus, the codes for lime saturation factor sub models of the MISO and modified Elman model are given in the appendix.

4.1.1 The benchmark model

The benchmark model simultaneously predicts all of the output quality parameters. The MATLAB code for this model is given in appendix II.

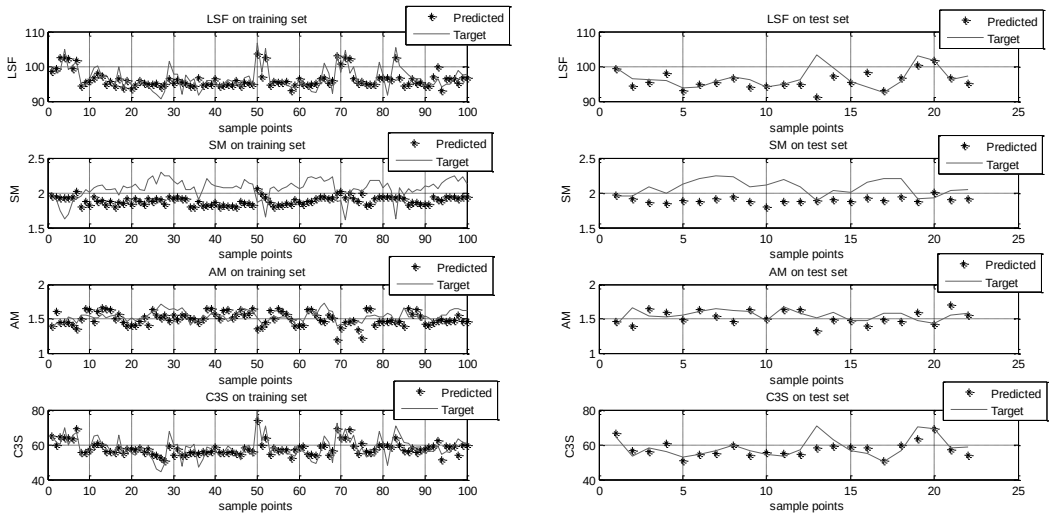


Figure 4.1.1: Target versus predicted value of the benchmark network

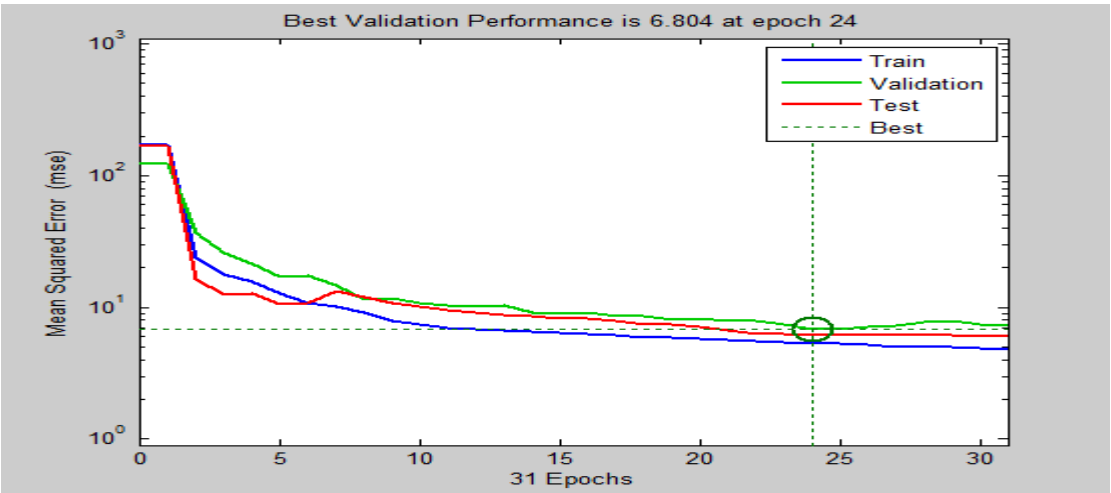


Figure 4.1.2: Validation performance of the benchmark network

Training parameters	
Epoch	31 Iterations
Time (sec.)	0:00:01
Performance	5.29
Gradient	1.44
Validation checks	6

Table 4.1.1: Training parameters of the benchmark network

	Mean square error			
	LSF	SM	AM	C3S
Training set	4.4730	0.0560	0.0103	16.6111
Test set	9.0888	0.0459	0.0112	15.6418

Table 4.1.2: Mean square errors of the benchmark network

4.1.2 The multiple input single output (MISO) network model

Lime saturation factor (LSF) sub model:

The MATLAB code for this sub model is given in appendix III.

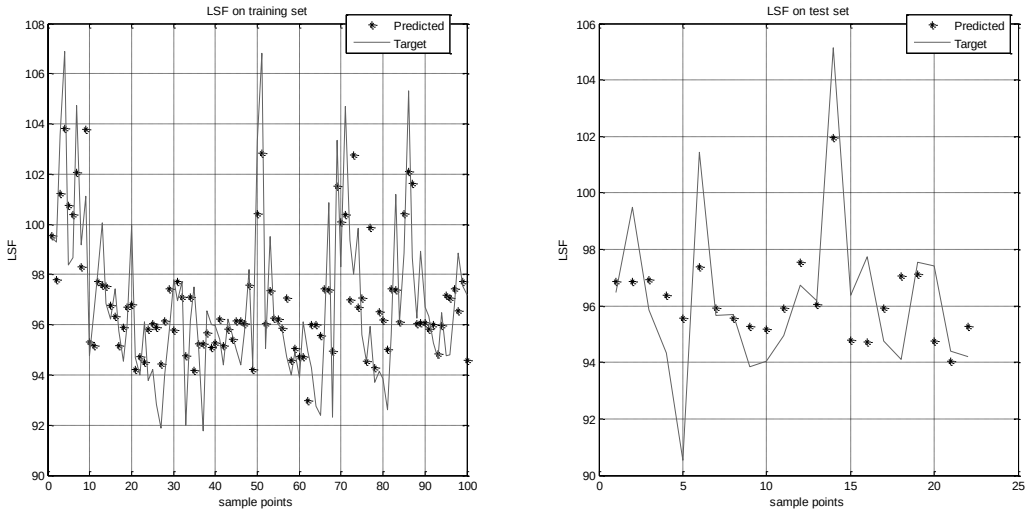


Figure 4.1.3: Target versus predicted value of the MISO network. [LSF sub model]

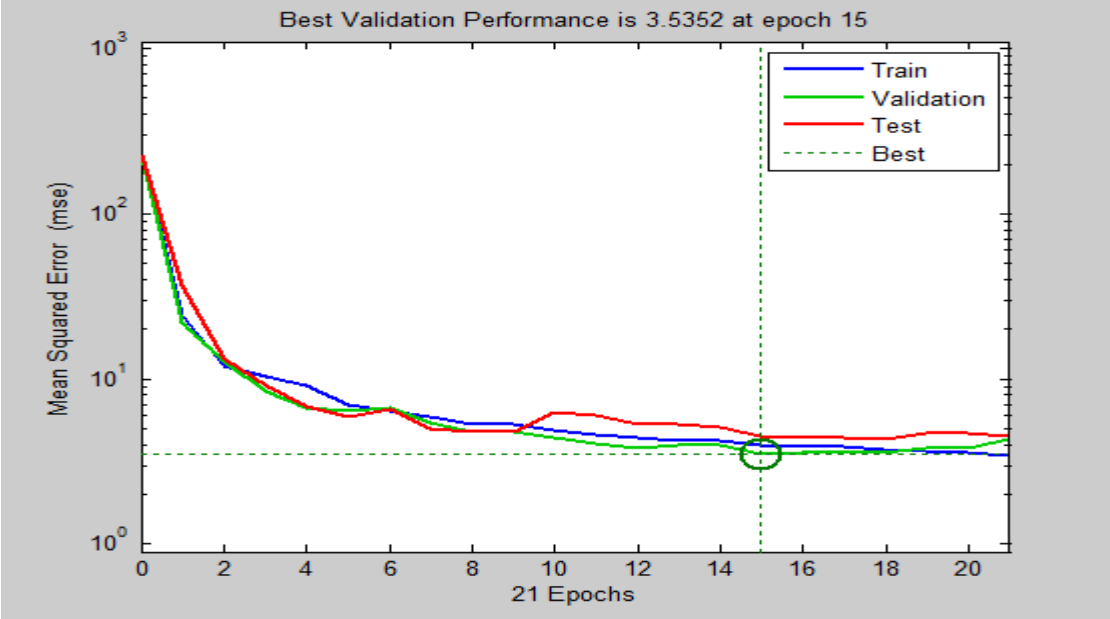


Figure 4.1.4: Validation performance of the MISO network.

[LSF sub model]

Training parameters	
Epoch	21 Iterations
Time (sec.)	0:00:00
Performance	4.01
Gradient	6.08
Validation checks	6

Table 4.1.3: Training parameters of the MISO network.

[LSF sub model]

	Mean square error (LSF)
Training set	4.0068
Test set	4.5229

Table 4.1.4: Mean square errors of the MISO network.

[LSF sub model]

Silica modulus (SM) sub model:

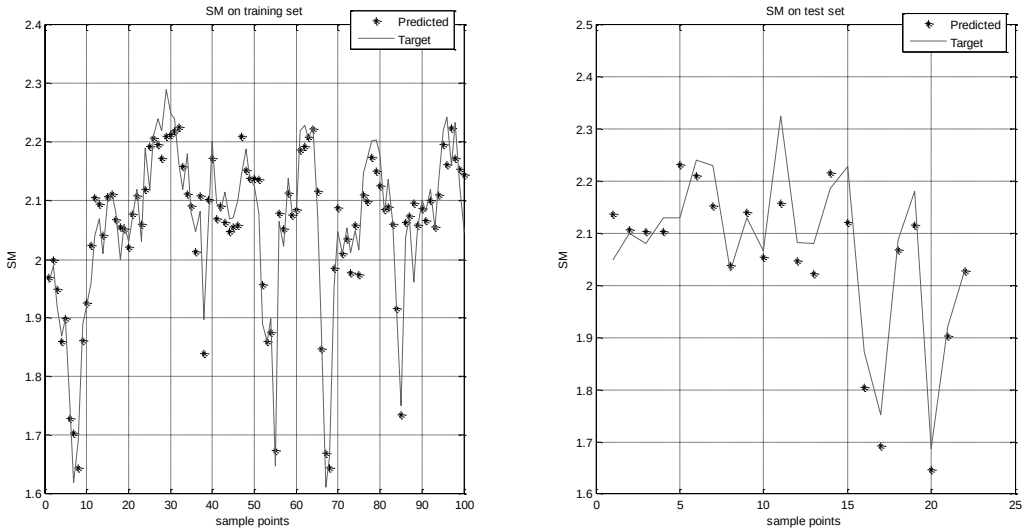


Figure 4.1.5: Target versus predicted value of the MISO network.

[SM sub model]

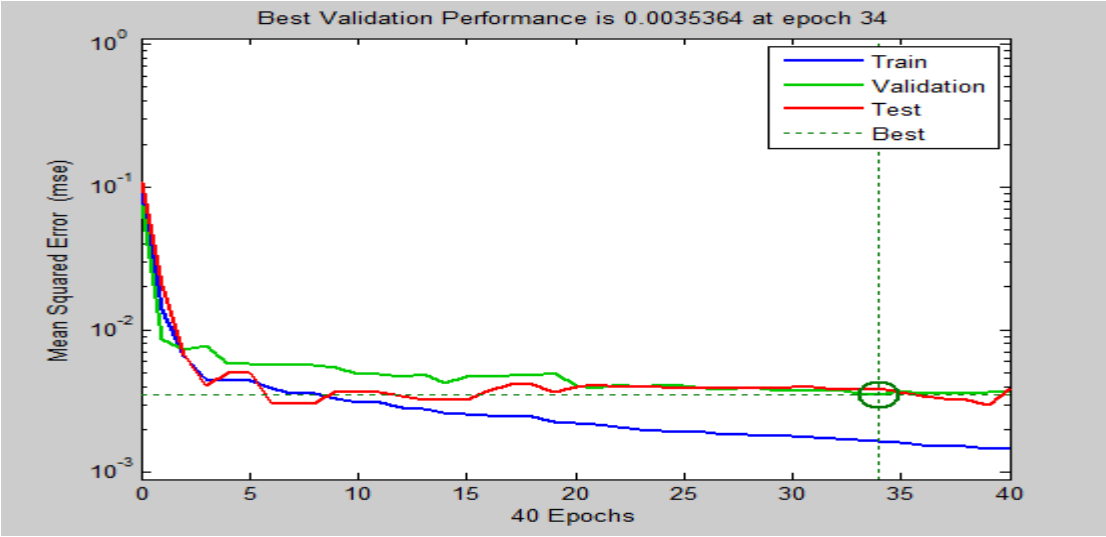


Figure 4.1.6: Validation performance of the MISO network.

[SM sub model]

Training parameters	
Epoch	40 Iterations
Time (sec.)	0:00:01
Performance	0.00166
Gradient	0.000831
Validation checks	6

Table 4.1.5: Training parameters of the MISO network.

[SM sub model]

	Mean square error (SM)
Training set	0.0017
Test set	0.0039

Table 4.1.6: Mean square errors of the MISO network.

[SM sub model]

Alumina modulus (AM) sub model:

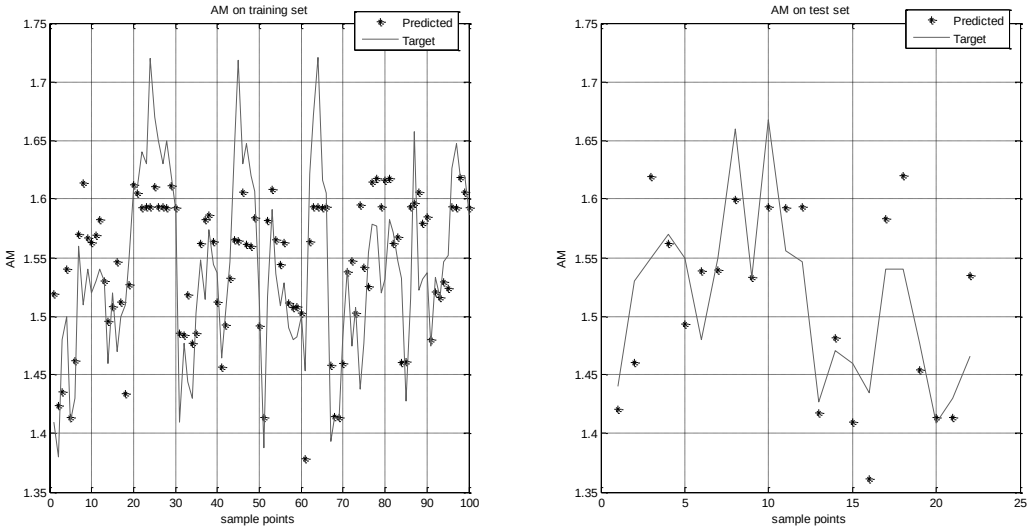


Figure 4.1.7: Target versus predicted value of the MISO network.

[AM sub model]

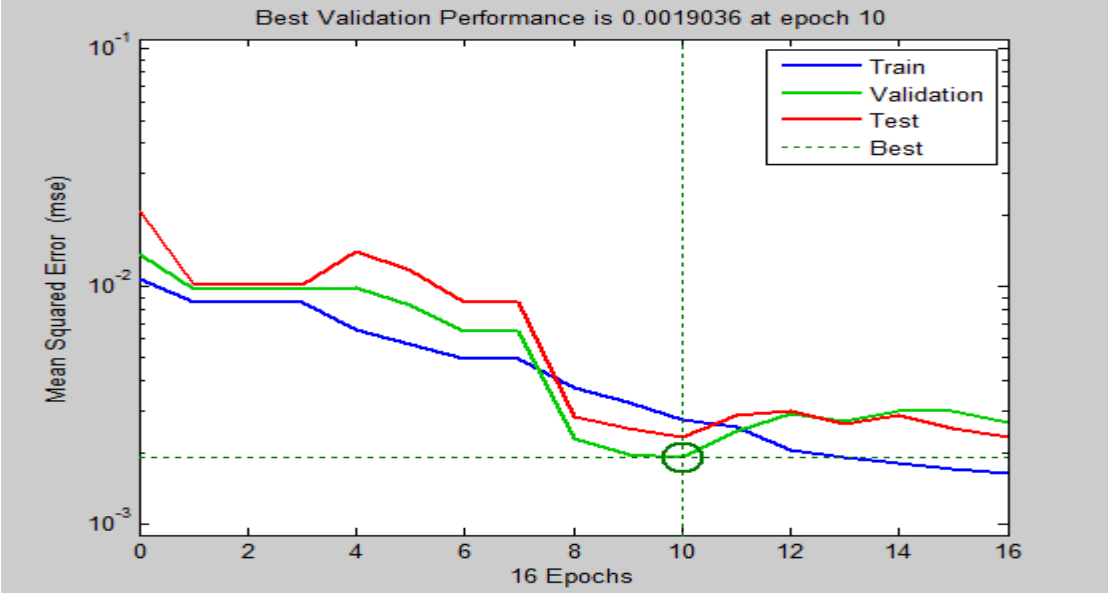


Fig.4.1.8: Validation performance of the MISO network.

[AM sub model]

Training parameters	
Epoch	16 Iterations
Time (sec.)	0:00:00
Performance	0.00276
Gradient	0.00171
Validation checks	6

Table 4.1.7: Training parameters of the MISO network.

[AM sub model]

	Mean square error (AM)
Training set	0.0028
Test set	0.0023

Table 4.1.8: Mean square errors of the MISO network.

[AM sub model]

Alite (C_3S) sub model:

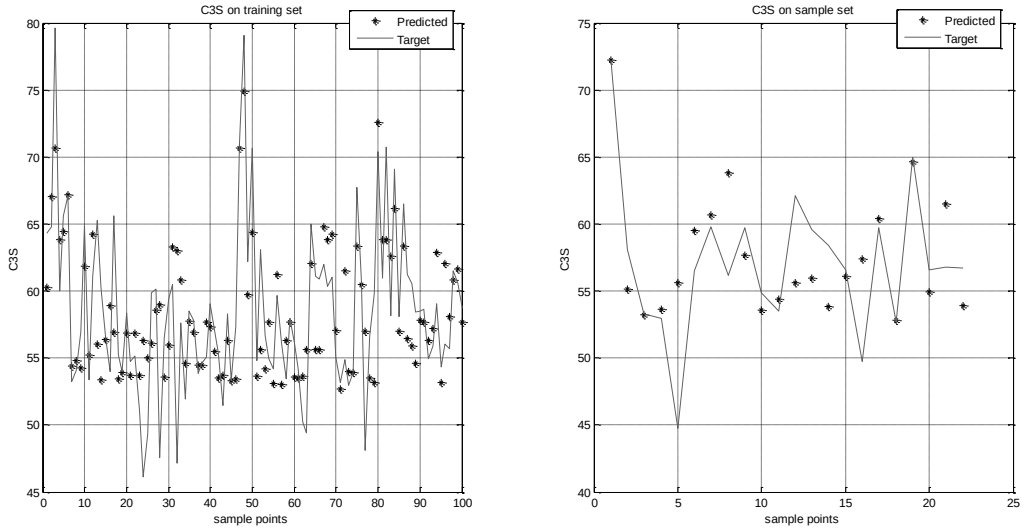


Figure 4.1.9: Target versus predicted value of the MISO network.

[Alite sub model]

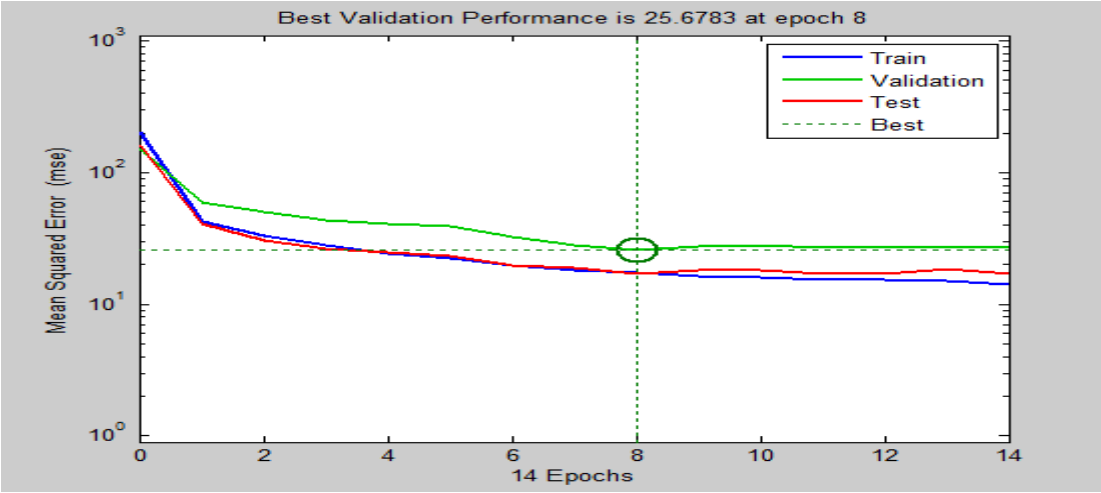


Figure 4.1.10: Validation performance of the MISO network.

[Alite sub model]

Training parameters	
Epoch	14 Iterations
Time (sec.)	0:00:00
Performance	17.5
Gradient	11.3
Validation checks	6

Table 4.1.9: Training parameters of the MISO network.

[Alite sub model]

	Mean square error (C_3S)
Training set	17.5088
Test set	16.8973

Table 4.1.10: Mean square errors of the MISO network.

[Alite sub model]

4.1.3 The modified Elman network model

Lime saturation factor (LSF) sub model:

The MATLAB code for this sub model is given in appendix IV.

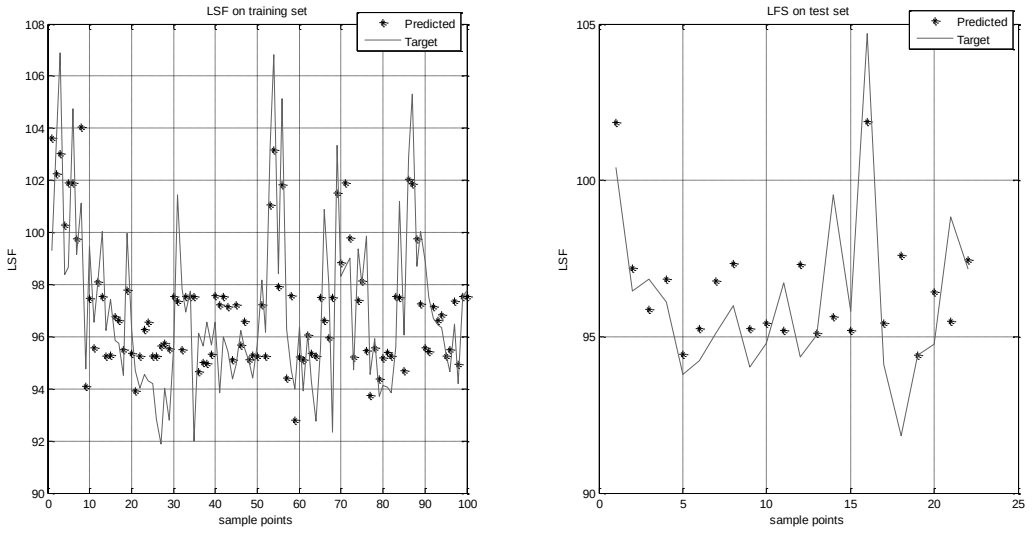


Figure 4.1.11: Target versus predicted value of the modified Elman network.

[LSF sub model]

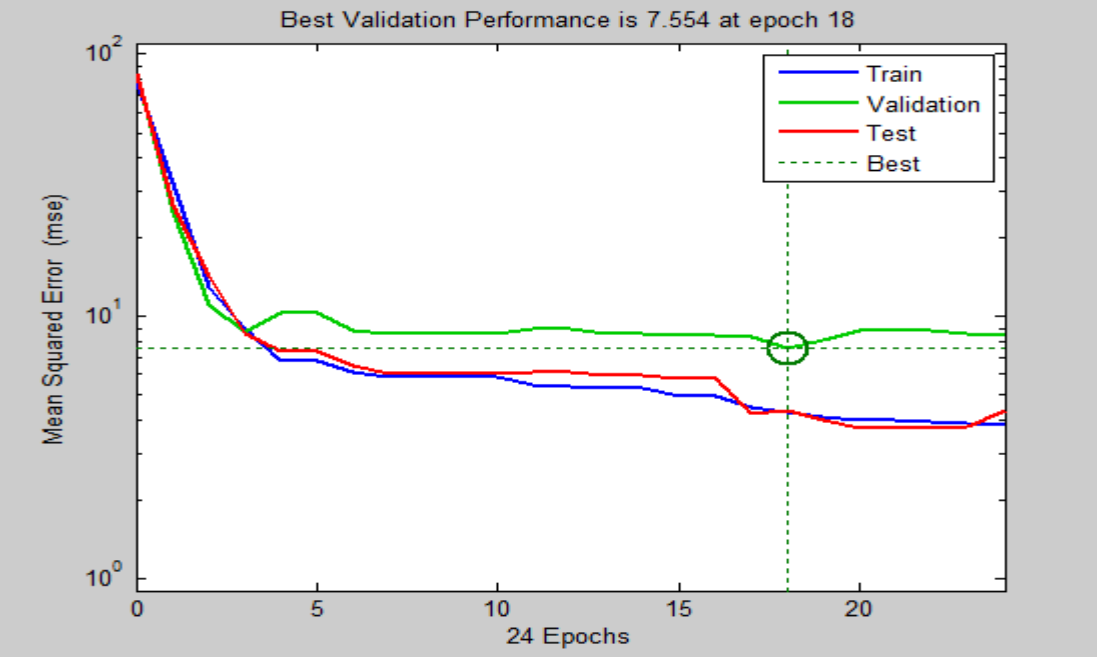


Figure 4.1.12: Validation performance of the modified Elman network.

[LSF sub model]

Training parameters	
Epoch	24 Iterations
Time (sec.)	0:00:12
Performance	4.31
Gradient	2.86
Validation checks	6

Table 4.1.11: Training parameters of the modified Elman network.

[LSF sub model]

	Mean square error (LSF)
Training set	4.3064
Test set	4.3482

Table 4.1.12: Mean square errors of the modified Elman network.

[LSF sub model]

Silica modulus (SM) sub model:

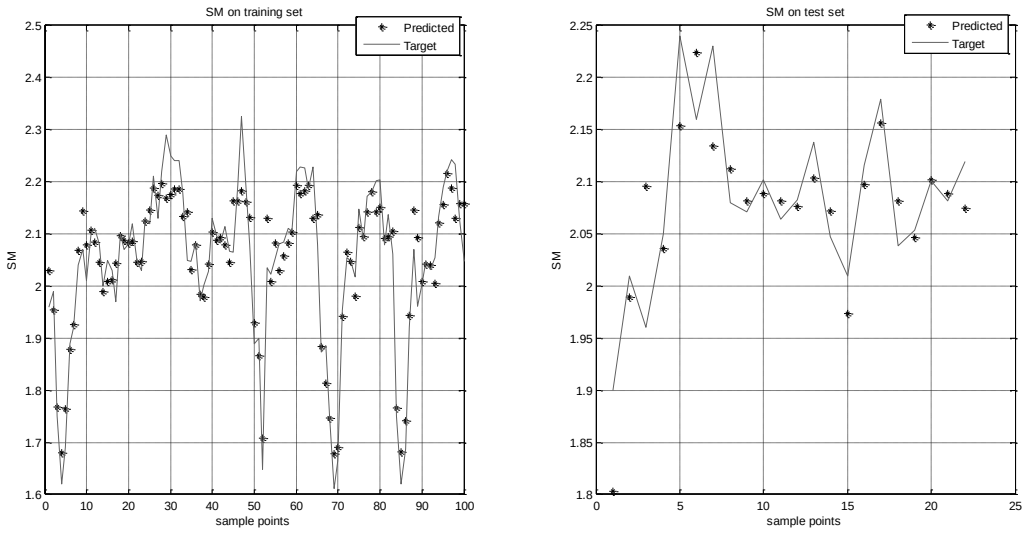


Figure 4.1.13: Target versus predicted value of the modified Elman network. [SM sub model]

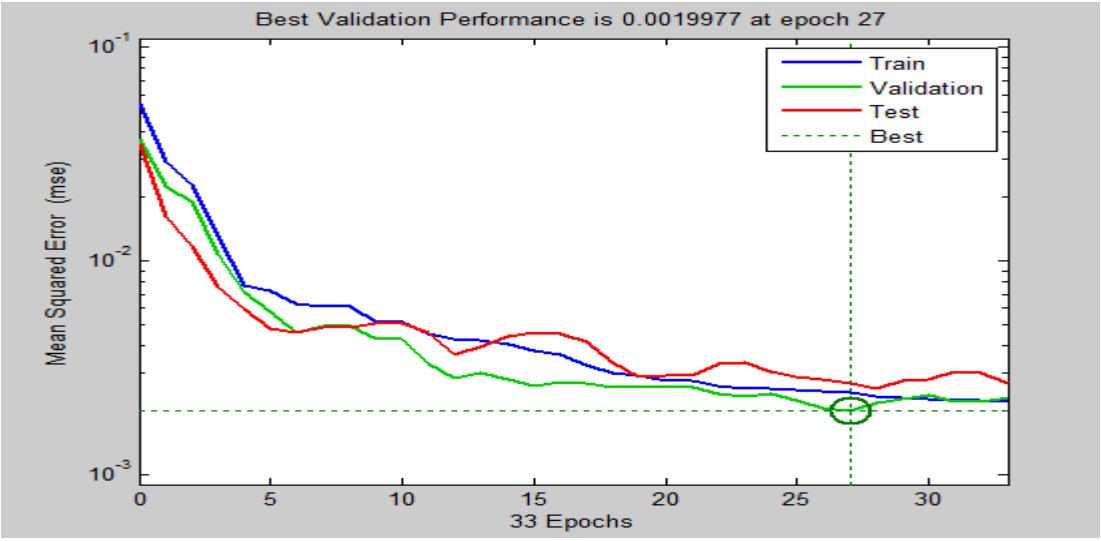


Figure 4.1.14: Validation performance of the modified Elman network. [SM sub model]

Training parameters	
Epoch	33 Iterations
Time (sec.)	0:00:18
Performance	0.00242
Gradient	0.00297
Validation checks	6

Table 4.1.13: Training parameters of the modified Elman network.

[SM sub model]

	Mean square error (SM)
Training set	0.0024
Test set	0.0027

Table 4.1.14: Mean square errors of the modified Elman network.

[SM sub model]

Alumina modulus (AM) sub model:

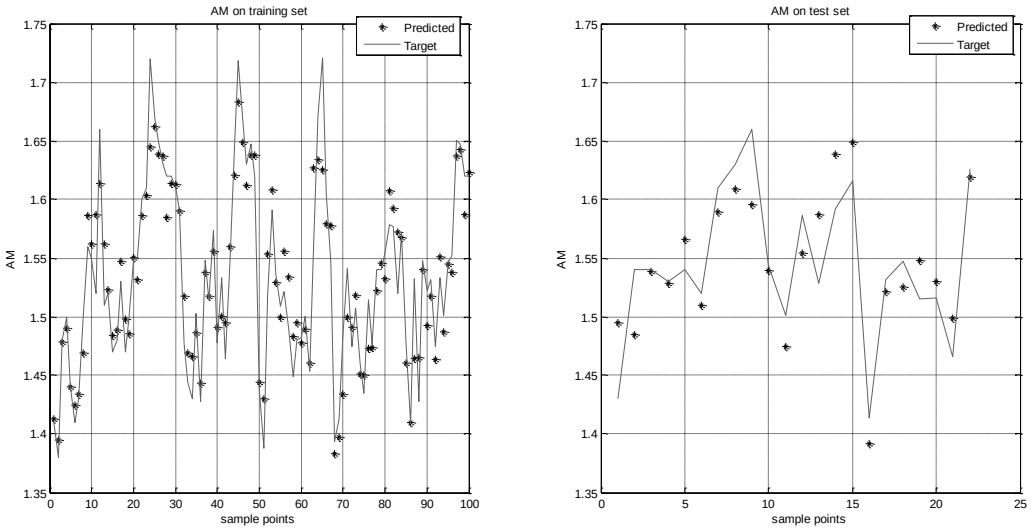


Figure 4.1.15: Target versus predicted value of the modified Elman network.

[AM sub model]

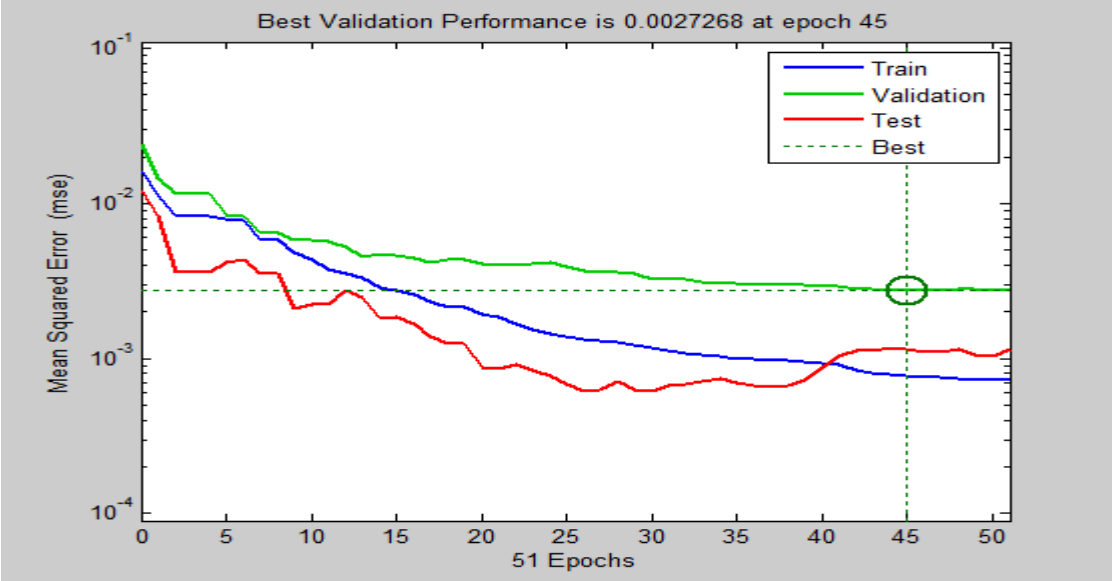


Figure 4.1.16: Validation performance of the modified Elman network.

[AM sub model]

Training parameters	
Epoch	51 Iterations
Time (sec.)	0:00:29
Performance	0.000773
Gradient	0.00143
Validation checks	6

Table 4.1.15: Training parameters of the modified Elman network.

[AM sub model]

	Mean square error (AM)
Training set	7.7289e-004
Test set	0.0011

Table 4.1.16: Mean square errors of the modified Elman network.

[AM sub model]

Alite (C_3S) sub model:

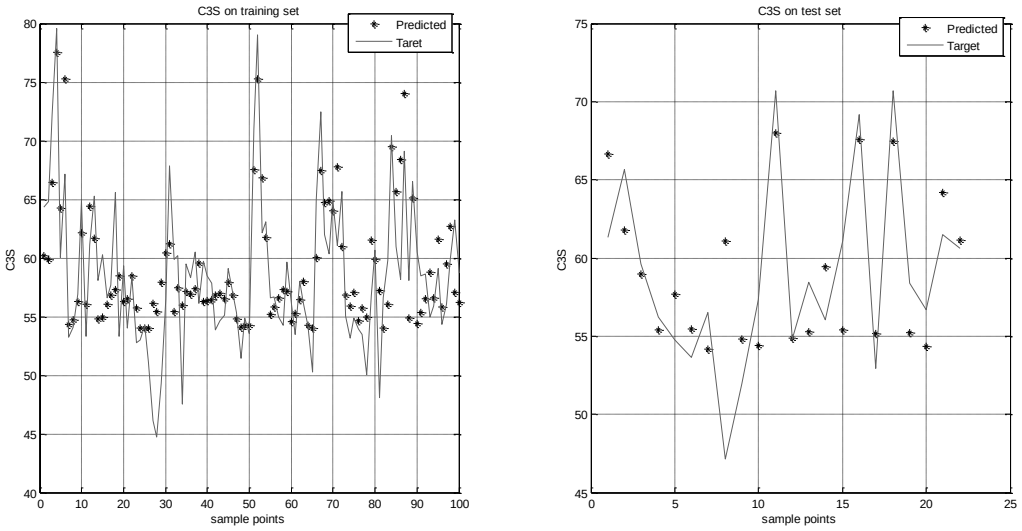


Figure 4.1.17: Target versus predicted value of the modified Elman network. [Alite sub model]

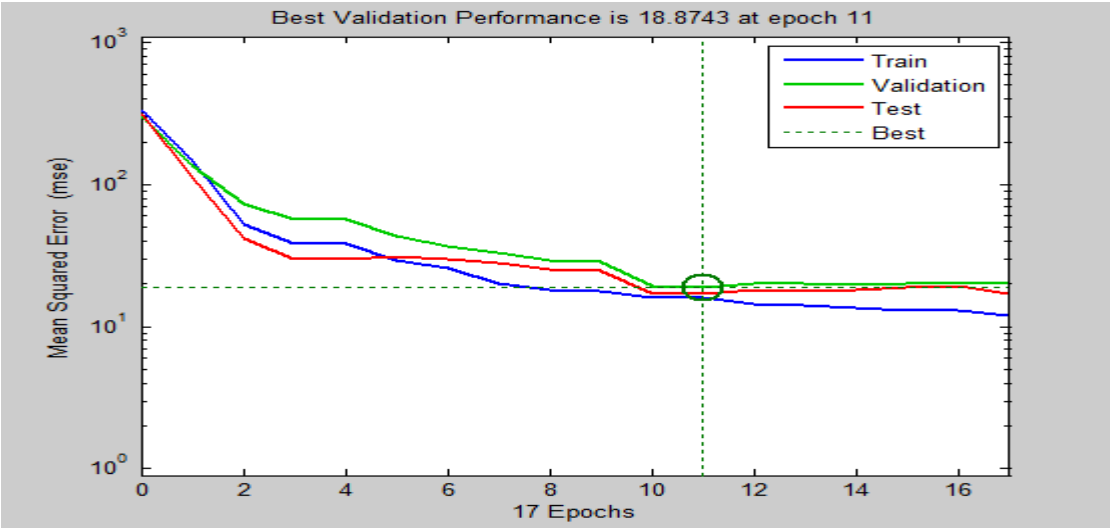


Figure 4.1.18: Validation performance of the modified Elman network. [Alite sub model]

Training parameters	
Epoch	17 Iterations
Time (sec.)	0:00:10
Performance	16.1
Gradient	28.0
Validation checks	6

Table 4.1.17: Training parameters of the modified Elman network.

[Alite sub model]

	Mean square error (C_3S)
Training set	16.0976
Test set	17.0127

Table 4.1.18: Mean square errors of the modified Elman network.

[Alite sub model]

In all models and sub models, the predicted value follows that of the target one. Visually inspecting these plots, it look that some models have better predicting performance than others. Besides, in the validation performance plot, the minimum error on the test set occurs around the iteration of best validation performance. This point of iteration is marked by a circle. Thus, the network being retuned at this point will better recognize new instances.

Visual inspection can be satisfactory for first instance information on how good the prediction is. However, it is difficult to compare multiple models using visual inspection. Thus, the mean square error (computed on actual values rather than normalized) is used as a means of comparing one model with the other to get the better one. Section 4.2 below compares and gives comprehensive summary and discussion on the results of each model and sub model given in this section.

4.2 Comparison of the models' result

The three models, including the benchmark, are compared with each other. The comparison is based on the mean square error on the test set for the set is not used while training. As a result, the mean square errors show the predicting performance of the models on new samples that the models are not trained on before. This performance comparison is presented in graphic and tabular forms.

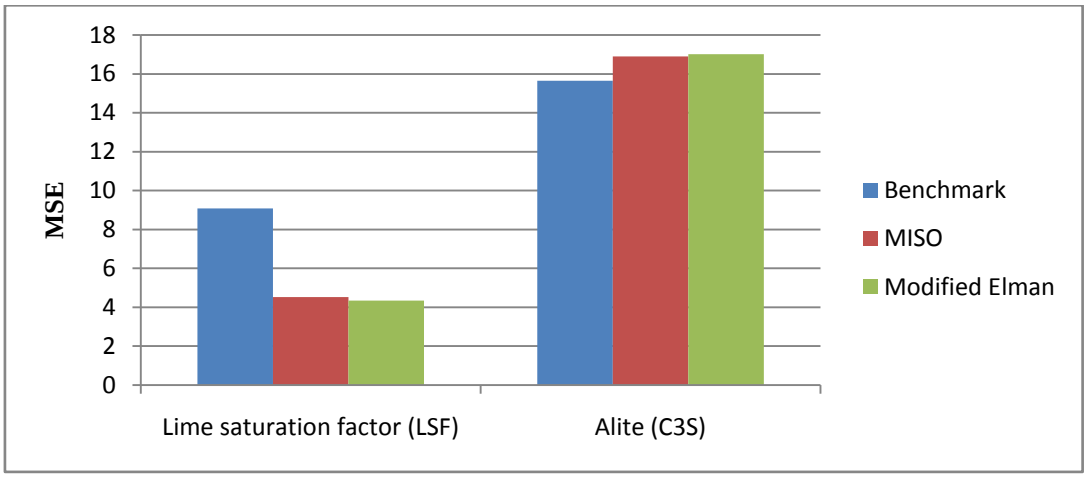


Figure 4.2.1: Mean square error plot of LSF and C_3S

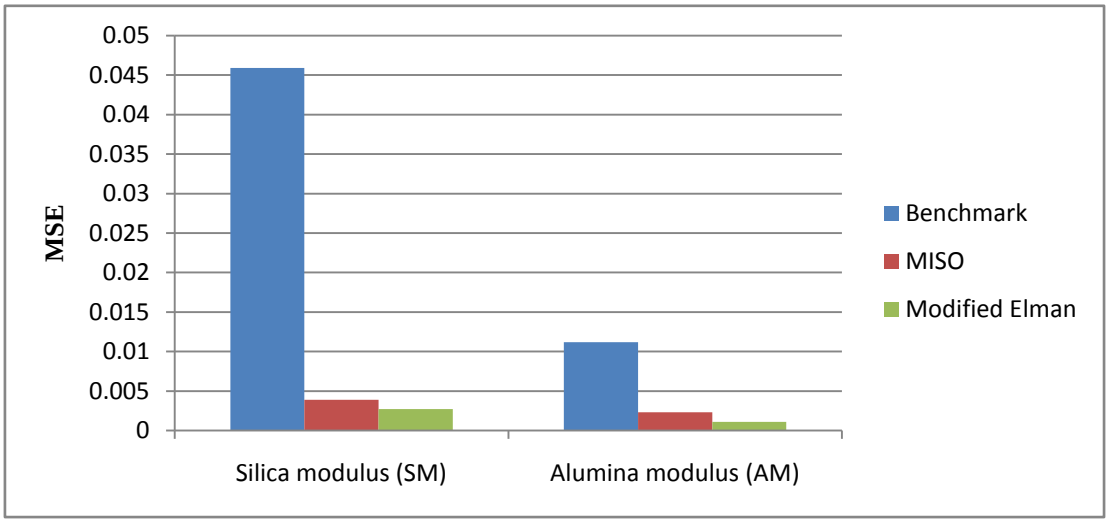


Figure 4.2.2: Mean square error plot of SM and AM

Lime saturation factor (LSF)				
Models	MSE	LSF value		$mse(\%) = \frac{mse}{\max} * 100$
		Minimum (min)	Maximum (max)	
Benchmark	9.0888	90.53	106.88	8.50
Multiple input single output (MISO)	4.5229	90.53	106.88	4.23
Modified Elman	4.3482	90.53	106.88	4.07

Table 4.2.1: Mean square error on lime saturation factor.

Silica modulus (SM)				
Models	MSE	SM value		$mse(\%) = \frac{mse}{\max} * 100$
		Minimum (min)	Maximum (max)	
Benchmark	0.0459	1.61	2.33	1.97
Multiple input single output (MISO)	0.0039	1.61	2.33	0.17
Modified Elman	0.0027	1.61	2.33	0.12

Table 4.2.2: Mean square error on silica modulus.

Alumina modulus (AM)				
Models	MSE	AM value		$mse(\%) = \frac{mse}{\max} * 100$
		Minimum (min)	Maximum (max)	
Benchmark	0.0112	1.38	1.72	0.65
Multiple input single output (MISO)	0.0023	1.38	1.72	0.13
Modified Elman	0.0011	1.38	1.72	0.06

Table 4.2.3: Mean square error on alumina modulus.

Alite (C_3S)				
Models	MSE	C_3S value		$mse(\%) = \frac{mse}{\max} * 100$
		Minimum (min)	Maximum (max)	
Benchmark	15.6418	44.72	79.60	19.65
Multiple input single output (MISO)	16.8973	44.72	79.60	21.23
Modified Elman	17.0127	44.72	79.60	21.37

Table 4.2.4: Mean square error on alite.

As can be seen from the above figures, figures 4.2.1 and figures 4.2.2, except for alite prediction, both multiple input single output (MISO) and modified Elman models have smaller mean square errors compared to the benchmark. Furthermore, the modified Elman model is generally superior in performance compared to the multiple input single output (MISO) model.

The above tables of comparison, table 4.2.2-4.2.4, show that the mean square errors of all the models on alite prediction are large. These errors are above nineteen percentage of the maximum alite value in the data. However, the mean square error while predicting the remaining three parameters, i.e. LSF, SM and AM, is comparatively smaller. This shows all of the three models are not good at predicting the alite value. As a result, an improvement on alite prediction is required.

4.3 The improvement on alite prediction

The improvement on alite prediction is achieved by predicting clinker oxides first and then calculating alite value using the multiple regression model developed, which is given by eq.3.3.10. This improvement is a simple rearrangement that rather than predicting the alite value directly, clinker oxides are predicted. These predicted values are used by the multiple regression model, which is now part of the improved alite predicting model, to calculate the alite value. As shown in the figure 4.3.4 below, the improved model's mean square error is smaller compared with the others. As can be seen form figure 4.3.3, the improvement reduces the error from above fifteen to around ten. The MATLAB code for this sub model is given in appendix V.

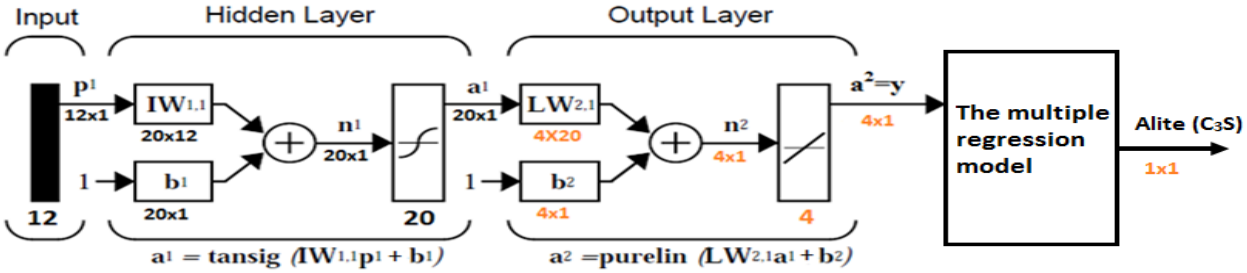


Figure 4.3.1: The improved alite predicting model

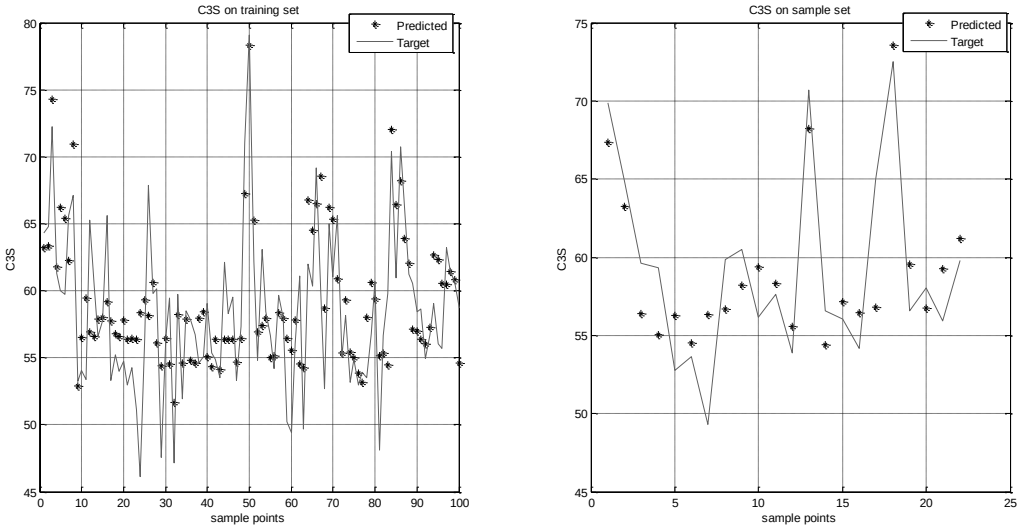


Figure 4.3.2: Target versus predicted value of the improved alite predicting model

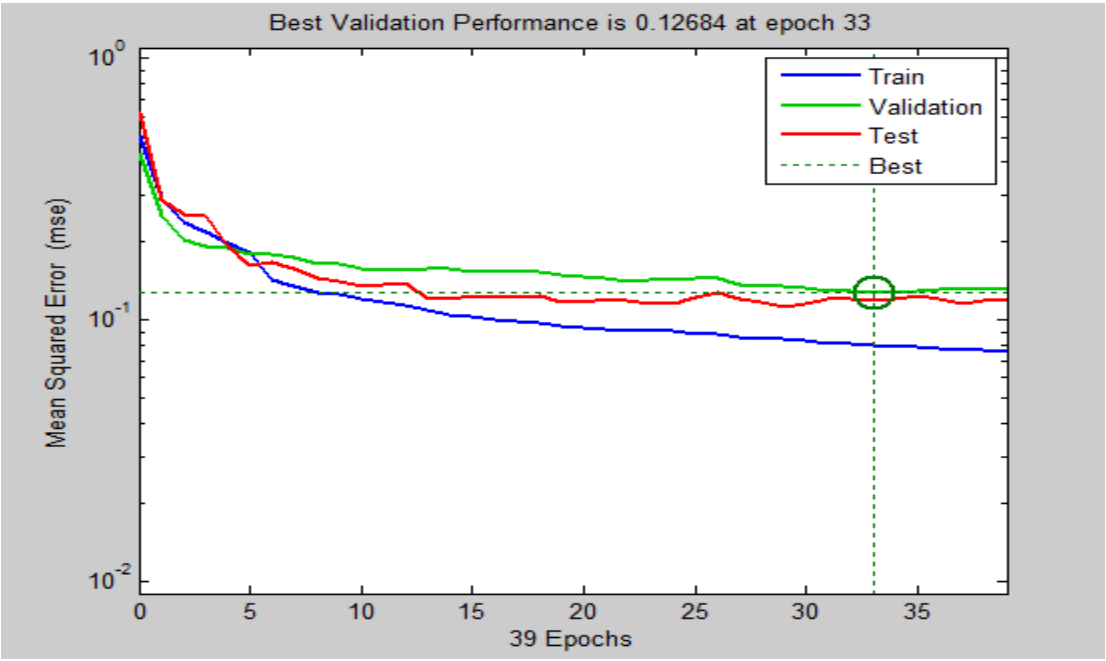


Figure 4.3.3: Validation performance of the improved alite predicting model

Training parameters	
Epoch	39 Iterations
Time (sec.)	0:00:00
Performance	0.0803
Gradient	0.0279
Validation checks	6

Table 4.3.1: Training parameters of the improved alite predicting model.

	Mean square error (C_3S)
Training set	15.3756
Test set	10.8759

Table 4.3.2: Mean square errors of the improved alite predicting model.

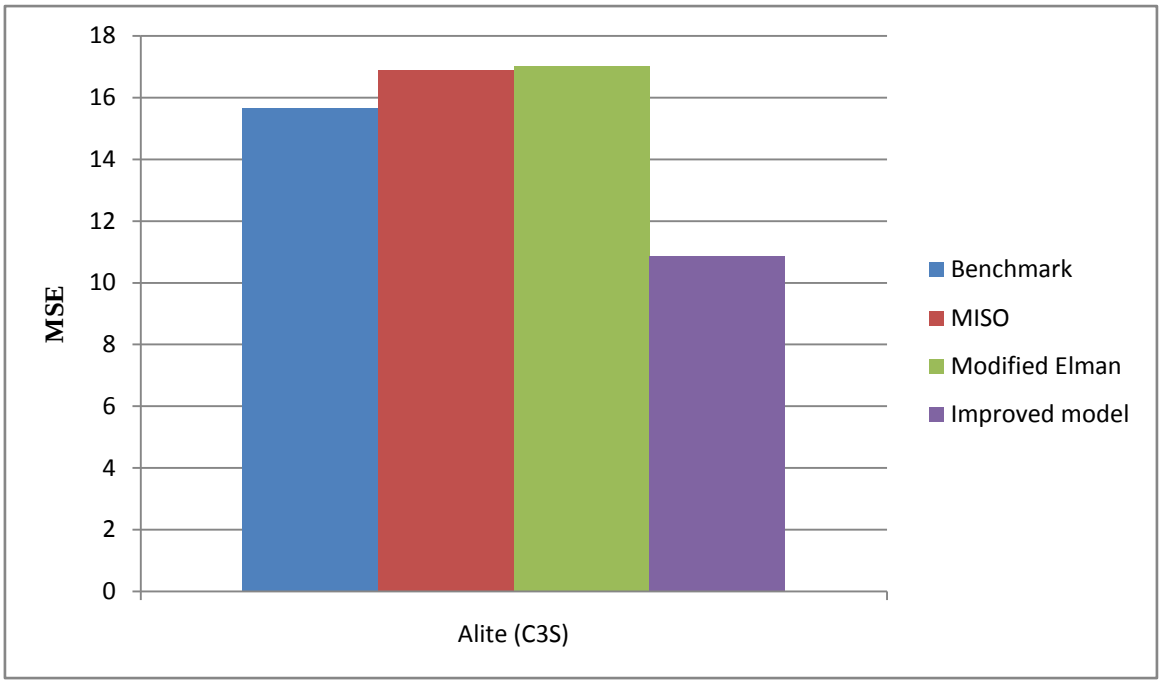


Figure 4.3.4: Mean square error plot of alite

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

In this thesis a neural network based data driven clinker quality predictor is developed taking Mugher cement factory as a case study. This predictor is developed through the steps of studying cement production in the selected case, collecting and preprocessing historical data, synthesizing artificial data, neural network model selection and finally training the network models.

Three neural network based models are used and compared and a fourth one is developed as an improvement. These models are the benchmark network, the multiple input single output (MISO) network, the modified Elman network and the improved alite predicting model. The clinker quality parameter predicting neural network models, developed in this thesis, are given in the CD named ‘auxiliary’.

The modified Elman network is best compared to the benchmark and multiple input single output (MISO) model in predicting LSF, SM and AM. However, the above mentioned models are involved with large mean square error while predicting alite. Therefore, an improved alite predicting model is developed. As a result, LSF (MSE=4.3482), SM (MSE=0.0027) and AM (MSE=0.0011) are better predicted by the modified Elman network and better prediction of alite (MSE=10.8759) is produced by the improved alite predicting model.

In conclusion, the predictor developed by this thesis can become an asset to the factory for estimating LSF, SM, AM and C_3S values of clinker. This will avoid measurement delays in the laboratory analysis or X-Ray Fluorescence technique. As a result, quick control action can be taken which in turn contributes to quality improvement.

5.2 Recommendation

It is recommended that the developed predictor can be further improved if the following upgrading can be achieved. First it is recommended to use larger historical process data. Besides, analyzing the actual economic gain of the quality improvement due to the predictor will make this thesis work astonishing. Furthermore, more complicated neural network structures can be employed as required. However, it will be difficult for all these enhancements to take effect unless Mugher cement factory digitalized all of the historical data, including operating parameters. Then after, the synthesized data used in this thesis can be removed and replaced by many actual input-output data pairs. This will make easy to train the network and the model can learn many features. I also recommend that, with these upgrading the thesis to be enriched as a bilateral work in association with Mugher cement factory.

REFERENCES

1. Petr Kadlec, On Robust and Adaptive Soft Sensors: a Thesis Submitted in Partial Fulfillment for the Degree of Doctor of Philosophy, Bournemouth University, December 17, 2009
2. H.F.W. Taylor, Cement Chemistry, 2nd edition,1997
3. Peter C. Hewlett, Lea's Chemistry of Cement and Concrete, Fourth Edition, 1998
4. Omid M. Omidvar and David L. Elliott, editors, Neural Systems for Control, February, 1997
5. Laekmariam Gedlu & Leuel Achamo, Basic Concepts of Cement Production Technology (In house training material for front line operational personnel), Mugher,2010
6. Ali Najah, Ahmed Elshafie, Othman A. Karim and Othman Jaffar, Prediction of Johor River Water Quality Parameters Using Artificial Neural Networks, European Journal of Scientific Research, ISSN 1450-216X Vol.28 No.3 (2009), pp.422-435.
7. Jian Shi and Xing-Gao Liu, Product Quality Prediction by a Neural Soft-sensor Based on MSA and PCA, International Journal of Automation and Computing 1 (2006) 17-22.
8. J. Wang et al. (Eds.), Hierarchical Neural Network Based Product Quality Prediction of Industrial Ethylene Pyrolysis Process,: ISNN 2006, LNCS 3973, pp. 1132 – 1137, 2006.
9. Sumit Goyal and Gyandera Kumar Goyal, Cascade and Feedforward Backpropagation Artificial Neural Network Models for Prediction of Sensory Quality of Instant Coffee Flavoured Sterilized Drink, Canadian Journal on Artificial Intelligence, Machine Learning and Pattern Recognition, Vol. 2, No. 6, August 2011.
10. Jiezhi Xin, Lianqing Zhu and Zhongyu Wang, editors, Temperature Prediction and Analysis Based on BP and Elman Neural Network for Cement Rotary Kiln, Fourth International Seminar on Modern Cutting and Measurement Engineering, Proc. of SPIE Vol. 7997,79972I, May 2011.
11. Ajaya Kumar Pani, Vamsi Vadlamudi, R J Bhargavi and Hare Krishna Mohanta, Neural Network Soft Sensor Application in Cement Industry: Prediction of Clinker Quality Parameters, Department of Chemical Engineering, Birla Institute of Technology and Science, Pilani, India.

12. Pani, Ajaya Kumar and Mohanta, Hare Krishna (2011), A Survey of Data Treatment Techniques for Soft Sensor Design, Chemical Product and Process Modeling: Vol. 6: Iss. 1, Article 2.
13. Mugher Cement Enterprise profile, 2011.
14. Ronald K. Pearson, Mining Imperfect Data, Dealing with Contamination and Incomplete Records, 2005.
15. Luigi Fortuna, Salvatore Graziani, Alessandro Rizzo and Maria G. Xibilia, Soft Sensors for Monitoring and Control of Industrial Process, 2007.
16. B. Lin et al., A Systematic Approach for Soft Sensor Development, Computers and Chemical Engineering 31 (2007) 419–425.
17. http://en.wikipedia.org/wiki/Cement_kiln#The_rotary_kiln (visited on 5/31/2012)
18. Javier Otamendi, Andrzej Bargiela, José Luis Montes and Luis Miguel Doncel Pedrera (Editors), Elman Neural Networks in Model Predictive Control, Proceedings 23rd European Conference on Modelling and Simulation.
19. Training of Elman Networks and Dynamic system Modelling, International Journal of Systems Science, D. T. Psent and X. Llut ,1996, volume 27, number 2, pages 221-226.
20. Javed I. Bhatty: editor in Chief, F. MacGregor Miller and Steven H. Kosmatka (editors), Innovations in Portland Cement Manufacturing, Portland Cement Association, 5420 Old Orchard Road,Skokie, Illinois 60077-1083.
21. Simon Hykin, Neural Network: A Comprehensive Foundation, 2nd ed., 1999.
22. Laurene Fausett, Fundamentals of Neural Networks: Architectures, Algorithms and Applications, 1994.
23. Moller, M.F., A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning, Neural Networks, Vol. 6, 1993, pp. 525-533.
24. Otto Labahn: Originated, Transl. by C. van Amerongen from the 6. German ed., Cement Engineers' Handbook/ - 4.Engl. ed. / by B. Kohlhass … - Wiesbaden; Berlin: Bauverlag, 1983.
25. Ursula Kääntee, Ron Zevenhoven, Rainer Backman and Mikko Hupa,Modelling a Cement Manufacturing Process to Study Possible Impacts of Alternative Fuels, Finland.

GLOSSARY

Cement: Cement is a material made by heating a mixture of limestone and clay in a kiln at about 1450^{o}C, then grinding to fine powder with a small addition of gypsum.

Clinker: Clinker is a nodular material that is ground up with gypsum to make cement. The nodules can be anything from 1mm to 25mm in diameter.

Free lime/ free CaO: It is unreacted CaO present in the clinker.

Induced draft (ID) fan: It is the fan that is installed in the cement production process to drive the gas stream movements.

Kiln meal: Kiln meal is the proportioned and homogenized raw material to be burned. When it is used as a rate variable, it is the rate of flow of the raw material.

Kiln meal variables: Kiln meal variable are variables that give the percentage by weight of the oxides in the kiln meal.

Kiln speed: Kiln speed is the angular speed of the rotary kiln.

Operating parameters: Operating parameters are those parameters which are manipulated, directly or indirectly, while producing clinker.

Precalciner: A precalciner is a furnace chamber introduced into the preheater into which 50-65% of the total amount of fuel is introduced.

Preheater: A preheater is a heat exchanger, usually of a type called a suspension preheater in which the moving powder of raw material is dispersed in a stream of hot gas coming from the kiln. The raw material leaves the preheater section at a temperature of about 800°C.

Quantity of oil before calciner burner: It is the rate of flow of oil in to the calciner burner.

Quantity of oil before kiln burner: It is the rate of flow of oil in to the kiln burner.

Rotary kiln: Rotary kiln is a tube made from steel plate, and lined with firebrick. The tube slopes slightly (1–4°) and slowly rotates on its axis at between 30 and 250 revolutions per hour

Soft sensors: sensors are predictive models. The term soft sensor is a combination of the words software, because the models are computer programs, and sensors, because the models deliver information similar to their hardware counterparts

X-ray fluorescence technique: This technique use electromagnetic radiation to analyze material composition in cement factory.

APPENDICES

APPENDIX I

Cement chemistry notation

Cement chemists use a form of notation which at first sight may seem a little odd. Oxides are referred to by their first letter: 'C' represents CaO, 'M' is MgO and so on; for all the oxides likely to be encountered in cementitious systems are shown in Appendix table 1 below.

List no.	Notation	Oxides
1	C	CaO
2	S	SiO_2
3	A	Al_2O_3
4	F	Fe_2O_3
5	K	K_2O
6	N	Na_2O
7	M	MgO
8	T	TiO_2
9	H	H_2O
10	$\bar{S}$	SO_3
11	$\bar{\bar{C}}$	CO_2

Appendix table 1: Cement chemistry notation

'Normal chemists' unfamiliar with this notation may find it strange to use 'C' to represent calcium oxide, rather than carbon, but there is a point to all this. It shortens what are otherwise very long, for example:

Tetracalcium aluminoferrite: $2(Ca_2AlFeO_5)$ is 4 $CaO.Al_2O_3.Fe_2O_3$; the compound becomes C_4AF when it get shortened.

Appendix II

MATLAB code for the Benchmark Model

```
% THE BENCHMARK NETWORK (MULTIPLE INPUT MULTIPLE OUTPUT FEEDFORWARD
% NETWORK:12-20-4)
% Loading the data
load C:\Users\mhr\Documents\MATLAB\thesis\networks\benchmark\data144ALL
% Write the actual path of the file used for training (data144ALL)
% using some of the inputs as an example
eg_myInput144=myInput144(:,1:3);
% using some of the outputs as an example
eg_myOutput144= myOutput144(:,1:3);
% Network definition
net=network;
% Architecture properties
net.numInputs=1;
net.numLayers=2;
net.biasConnect=[1;1];
net.inputConnect=[1;0];
net.layerConnect=[0 0 ;1 0];
net.outputConnect=[0 1];
% Subobject properties
net.inputs{1}.exampleInput=eg_myInput144;
net.inputs{1}.processFcns={'mapminmax'};
net.inputs{1}.processParams{1}.ymin=0;
net.inputs{1}.processParams{1}.ymax=1;
net.layers{1}.size=20;
net.layers{1}.transferFcn='tansig';
net.layers{1}.initFcn='initnw';
net.layers{2}.initFcn='initnw';
net.outputs{2}.exampleOutput=eg_myOutput144;
net.outputs{2}.processFcns={'mapminmax'};
net.outputs{1}.processParams{1}.ymin=0;
net.outputs{1}.processParams{1}.ymax=1;
% Network fnctions
net.initFcn = 'initlay';
net.performFcn = 'mse';
net.trainFcn='trainscg';
net.divideFcn = 'dividerand';
net.divideParam.trainRatio=0.7;
net.divideParam.valRatio=0.15;
net.divideParam.testRatio=0.15;
net.plotFcns = {'plotperform'};
```

```
% Network behavior
% initialization
net=init(net);
% trainning
[net,tr]=train(net,myInput144,myOutput144);
% simulation on the training set
y= sim(net,myInput144(:,tr.trainInd));
y=y';
myOutput144Tr=myOutput144(:,tr.trainInd)'; % training set target values
% LSF plot on training set
subplot(4,2,1);
plot(y(:,1),'*r')
title('LSF on training set')
xlabel('sample points')
ylabel('LSF')
grid on
hold on
plot(myOutput144Tr(:,1))
% SM plot on training set
subplot(4,2,3);
plot(y(:,2),'*r')
title('SM on training set')
xlabel('sample points')
ylabel('SM')
grid on
hold on
plot(myOutput144Tr(:,2))
% AM plot on training set
subplot(4,2,5);
plot(y(:,3),'*r')
title('AM on training set')
xlabel('sample points')
ylabel('AM')
grid on
hold on
plot(myOutput144Tr(:,3))
% C3S plot on the training set
subplot(4,2,7);
plot(y(:,4),'*r')
title('C3S on training set')
xlabel('sample points')
ylabel('C3S')
grid on
hold on
plot(myOutput144Tr(:,4))
% simulation on test set
```

```
test=sim(net,myInput144(:,tr.testInd));
test=test';
myOutputTest=myOutput144(:,tr.testInd)'; % test set target values
% LSF plot (on test)
subplot(4,2,2);
plot(test(:,1),'*r')
title('LSF on test set')
xlabel('sample points')
ylabel('LSF')
grid on
hold on
plot(myOutputTest(:,1))
% SM plot (on test)
subplot(4,2,4);
plot(test(:,2),'*r')
title('SM on test set')
xlabel('sample points')
ylabel('SM')
grid on
hold on
plot(myOutputTest(:,2))
% AM plot (on test)
subplot(4,2,6);
plot(test(:,3),'*r')
title('AM on test set')
xlabel('sample points')
ylabel('AM')
grid on
hold on
plot(myOutputTest(:,3))
% C3S plot (on test)
subplot(4,2,8);
plot(test(:,4),'*r')
title('C3S on test set')
xlabel('sample points')
ylabel('C3S')
grid on
hold on
plot(myOutputTest(:,4))
% mse calculation (on trainig data set)
eLSF=(y(:,1)-myOutput144Tr(:,1));
mseLSF=mse(eLSF);
eSM=(y(:,2)-myOutput144Tr(:,2));
mseSM=mse(eSM);
eAM=(y(:,3)-myOutput144Tr(:,3));
mseAM=mse(eAM);
```

```
eC3S=(y(:,4)-myOutput144Tr(:,4));
mseC3S=mse(eC3S);
% mse calculation (on test data set)
Test_eLSF=(test(:,1)-myOutputTest(:,1));
Test_mseLSF=mse(Test_eLSF);
Test_eSM=(test(:,2)-myOutputTest(:,2));
Test_mseSM=mse(Test_eSM);
Test_eAM=(test(:,3)-myOutputTest(:,3));
Test_mseAM=mse(Test_eAM);
Test_eC3S=(test(:,4)-myOutputTest(:,4));
Test_mseC3S=mse(Test_eC3S);
varbl=['LSF(tr/ts)' 'SM(tr/ts)' 'AM(tr/ts)' 'C3S(tr/ts)']
mse=[mseLSF mseSM mseAM mseC3S;Test_mseLSF Test_mseSM Test_mseAM Test_mseC3S]
```

Appendix III

MATLAB code for the lime saturation factor sub model of the MISO case

```
% MULTIPLE INPUT SINGLE OUTPUT (MISO) FEEDFORWARD NETWORK
% Lime Saturation Factor (LSF) sub model
% Loading the data
load C:\Users\mhr\Documents\MATLAB\thesis\networks\MISO\LSF\data144LSF
% Write the actual path of the file used for training (data144LSF)
% auxiliary
% using some of the inputs as an example
eg_myInput144=myInput144(:,1:3);
% using some of the outputs as an example
eg_LSF144= LSF144(:,1:3);
% Network definition
net=network;
% Architecture properties
net.numInputs=1;
net.numLayers=2;
net.biasConnect=[1;1];
net.inputConnect=[1;0];
net.layerConnect=[0 0 ;1 0];
net.outputConnect=[0 1];
% Subobject properties
net.inputs{1}.exampleInput=eg_myInput144;
net.inputs{1}.processFcns={'mapminmax'};
net.inputs{1}.processParams{1}.ymin=0;
net.inputs{1}.processParams{1}.ymax=1;
net.layers{1}.size=20;
net.layers{1}.transferFcn='tansig';
net.layers{1}.initFcn='initnw';
net.layers{2}.initFcn='initnw';
net.outputs{2}.exampleOutput=eg_LSF144;
net.outputs{2}.processFcns={'mapminmax'};
net.outputs{1}.processParams{1}.ymin=0;
net.outputs{1}.processParams{1}.ymax=1;
% Network fnctions
net.initFcn = 'initlay';
net.performFcn = 'mse';
net.trainFcn='trainscg';
% generalization
```

```
net.divideFcn = 'dividerand';
net.divideParam.trainRatio=0.7;
net.divideParam.valRatio=0.15;
net.divideParam.testRatio=0.15;
net.plotFcns = {'plotperform'};
% Network behavior
% initialization
net=init(net);
% trainning
[net,tr]=train(net,myInput144,LSF144);
% simulation on the training set
y= sim(net,myInput144(:,tr.trainInd));
subplot(1,2,1);
plot(y,'*r')
title('LSF on training set')
xlabel('sample points')
ylabel('LSF')
hold on
grid on
plot(LSF144(tr.trainInd))
% simulation on test set
test=sim(net,myInput144(:,tr.testInd));
subplot(1,2,2);
plot(test,'*r')
title('LSF on test set')
xlabel('sample points')
ylabel('LSF')
grid on
hold on
plot(LSF144(tr.testInd))
% mse calculation
e1=(y-LSF144(tr.trainInd))';
e2=(test-LSF144(tr.testInd))';
mse_train=mse(e1)
mse_test=mse(e2)
```

Appendix IV

MATLAB code for the lime saturation factor sub model of the modified Elman case

```
% MODIFIED ELMAN NEURAL NETWORK
% Lime Saturation Factor (LSF) sub model
% Loading the data
load C:\Users\mhr\Documents\MATLAB\thesis\networks\mElman\LSF\data144LSF
% Write the actual path of the file used for training (data144LSF)
% auxiliary
% using some of the inputs as an example
eg_myInput144=myInput144(:,1:3);
% using some of the outputs as an example
eg_LSF144= LSF144(:,1:3);
% Network definition
net=network;
% Architecture properties
net.numInputs=1;
net.numLayers=3;
net.biasConnect=[1;0;1];
net.inputConnect=[1;0;0];
net.layerConnect=[0 1 0;1 1 0;1 0 0];
net.outputConnect=[0 0 1];
% Subobject properties
net.inputs{1}.exampleInput=eg_myInput144;
net.outputs{1}.processParams{1}.ymin=0;
net.outputs{1}.processParams{1}.ymax=1;
net.inputs{1}.processFcns={'mapminmax'};
net.layers{1}.size=20;
net.layers{1}.transferFcn='tansig';
net.layers{1}.initFcn='initnw';
net.layers{2}.size=20;
net.layers{2}.initFcn='initnw';
net.layers{3}.initFcn='initnw';
net.outputs{3}.exampleOutput=eg_LSF144;
net.outputs{3}.processFcns={'mapminmax'};
net.outputs{1}.processParams{1}.ymin=0;
net.outputs{1}.processParams{1}.ymax=1;
net.layerWeights{2,1}.delays=1;
net.layerWeights{2,1}.learn=0;
net.layerWeights{2,2}.delays=1;
```

```
net.layerWeights{2,2}.learn=0;
% Network fnctions
net.initFcn = 'initlay';
net.performFcn = 'mse';
net.trainFcn='trainscg';
% generalization
net.divideFcn = 'dividerand';
net.divideParam.trainRatio=0.7;
net.divideParam.valRatio=0.15;
net.divideParam.testRatio=0.15;
net.plotFcns = {'plotperform'};
% Network behavior
% initialization
net=init(net);
% trainning
[net,tr]=train(net,myInput144,LSF144);
% simulation on the training set
y= sim(net,myInput144(:,tr.trainInd));
subplot(1,2,1);
%plot(y{1,1})
plot(y,'*r')
title('LSF on training set')
xlabel('sample points')
ylabel('LSF')
grid on
hold on
plot(LSF144(tr.trainInd))
% simulation on test set
test=sim(net,myInput144(:,tr.testInd));
subplot(1,2,2);
plot(test,'*r')
title('LFS on test set')
xlabel('sample points')
ylabel('LSF')
grid on
hold on
plot(LSF144(tr.testInd))
% mse calculation
e1=(y-LSF144(tr.trainInd))';
e2=(test-LSF144(tr.testInd))';
mse_train=mse(e1)
mse_test=mse(e2)
```

Appendix V

MATLAB code for the improvement alite (C_3S) prediction sub model

```
%IMPROVED ALITE PREDICTING MODEL
%THE NETWORK DIRECTLY MODELS INPUT-CLINKER OXIDE R/N NOT INPUT-ALITE R/N
% Loading the data
load C:\Users\mhr\Documents\MATLAB\thesis\networks\imprvdAlite\data144Oxide
% Write the actual path of the file used for training (data144Oxide)
% auxiliary
% using some of the inputs as an example
eg_myInput144=myInput144(:,1:3);
% using some of the outputs as an example
eg_myOutput144Oxide= myOutput144Oxide(:,1:3);
% Network definition
net=network;
% Architecture properties
net.numInputs=1;
net.numLayers=2;
net.biasConnect=[1;1];
net.inputConnect=[1;0];
net.layerConnect=[0 0 ;1 0];
net.outputConnect=[0 1];
% Subobject properties
net.inputs{1}.exampleInput=eg_myInput144;
net.inputs{1}.processFcns={'mapminmax'};
net.inputs{1}.processParams{1}.ymin=0;
net.inputs{1}.processParams{1}.ymax=1;
net.layers{1}.size=20;
net.layers{1}.transferFcn='tansig';
net.layers{1}.initFcn='initnw';
net.layers{2}.initFcn='initnw';
net.outputs{2}.exampleOutput=eg_myOutput144Oxide;
net.outputs{2}.processFcns={'mapminmax'};
net.outputs{1}.processParams{1}.ymin=0;
net.outputs{1}.processParams{1}.ymax=1;
% Network fnctions
net.initFcn = 'initlay';
net.performFcn = 'mse';
net.trainFcn='trainscg';
%generalization
```

```
net.divideFcn = 'dividerand';
net.divideParam.trainRatio=0.7;
net.divideParam.valRatio=0.15;
net.divideParam.testRatio=0.15;
net.plotFcns = {'plotperform'};
% Network behavior
% initialization
net=init(net);
% trainning
[net,tr]=train(net,myInput144,myOutput144Oxide);
% simulation on the training set
y= sim(net,myInput144(:,tr.trainInd));
y=-38.2749+4.3795*y(4,:)-7.0331*y(1,:)-8.0014*y(2,:)+1.0512*y(3,:);
subplot(1,2,1);
plot(y,'*r')
title('C3S on training set')
xlabel('sample points')
ylabel('C3S')
grid on
hold on
plot(actual_C3S(tr.trainInd))
% simulation on test set
test=sim(net,myInput144(:,tr.testInd));
test=-38.2749+4.3795*test(4,:)-7.0331*test(1,:)-
8.0014*test(2,:)+1.0512*test(3,:);
subplot(1,2,2);
plot(test,'*r')
title('C3S on sample set')
xlabel('sample points')
ylabel('C3S')
grid on
hold on
plot(actual_C3S(tr.testInd))
% mse calculation
e1=(y-actual_C3S(tr.trainInd))';
e2=(test-actual_C3S(tr.testInd))';
mse_train=mse(e1)
mse_test=mse(e2)
```

Appendix VI

Picture of a typical log sheet