DEVELOPMENT OF NEURAL NETWORK PREDICTION MODEL FOR GAS CONSUMPTION

by

MOHAMMAD SHOLEH BIN ABDULLAH

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)

(Electrical & Electronics Engineering)

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

DEVELOPMENT OF NEURAL NETWORK PREDICTION MODEL FOR GAS CONSUMPTION

by

Mohammad Sholeh Bin Abdullah

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

____

(Dr. Rosdiazli Ibrahim)
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

December 2009

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

____

Mohammad Sholeh Bin Abdullah

ACKNOWLEDGEMENTS

Assalamualaikum w.b.t.

The author wishes to take the opportunity to express his utmost gratitude to the individual that have taken the time and effort to assist the author in completing the project either directly or indirectly. Without the cooperation of these individuals, no doubt the author would have faced some minor complications throughout the course.

First and foremost the author's utmost gratitude goes to the author's supervisor, Dr. Rosdiazli Ibrahim. Thanks for all the never-ending support and trust given to the author in order to carry out this project. Without his guidance and patience, the author would not be succeeded to complete the project.

To the technician in Electrical & Electronics Engineering, especially Mrs. Siti Hawa, thank you for assisting the author in completing her project.

Last but not least, to entire PHD and Master's student especially Miss Maryam Jamela and others who helped the author by giving an advice and some information about the project, the author thank you all for the wonderful hands.

ABSTRACT

This report basically discusses the preliminary research done and basic understanding of the chosen topic, which is **Development of Neural Network Prediction Model for Gas Consumption**. The objective of this project is to create a system to predict gas consumption using neural network ideology. Prediction in this project gives a broad path of ideas especially when it is related to gas consumption. A study on the gas behavior, parameters involves and input-output analysis shall be conducted theoretically and experimentally. Neural network give the process application to draw an input-output mapping. An extensive process of learning will be given by feeding task examples that consist of unique input and corresponding desired response. Neural network also implement adaptability behavior which is important in order to response for changes especially environmental changes. At the end of project, the system should be able to improve gas consumption process by including the extensive features into the predicting model.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

This chapter will give an introduction and explanation of the project entitled "**Development of Neural Network Prediction Model for Gas Consumption**". A background about this project is given followed by statement of the problems to be addressed and lastly the objectives and scope of the work are pointed out.

1.1. Background of Study

The prediction of gas consumption has been crucial nowadays especially for gas distribution and transportation companies as well as for the government agencies that associated to this sector. A prediction might be divided into various types depending on the target or output desired. Neural network is one of the powerful tools that can be used to achieve the objective. Neural networks are a wide class of flexible nonlinear regression and discriminant models, data reduction models, and nonlinear dynamical systems. The idea basically mimics the brain process where the neurons are the computing elements to diverse the outcome. In this case, they consist of an often large number of "neurons" interconnected in often complex ways and often organized into layers. Artificial neural networks are used in three main ways:

- Models of biological nervous systems and "intelligence"
- Real-time adaptive signal processors or controllers
- Data analytic methods

From the purposes, a process application, which is gas consumption process line, will be connected with neural network system to create "intelligence" system as the result of self-organization or learning.

1.2. Problem Statement

Gas or more accurately natural gas has been broadly use and distribute from the sources to the users such as commercial industries and residential. The transportation of gas, in Malaysia, is using the pipeline from one destination to one destination. For the supplier which is PETRONAS Gas Berhad, all the transportation is using pipeline system to all customers, domestic or international. The problem occurred when the gas volume send is not equal to the received value. As an example, given here in Malaysia, the gas flow meter shows 100% of volume and when the gas reaching Thailand, the receiving value is about 98%-95%. Therefore the project is to create a system using neural network basis to improve the problem occur on gas consumption.

1.3. Objectives and Scope of Study

The study to be conducted is not limited to one area of field only as the project involving parameters from various fields.

The objectives of the project are:

1. To study on artificial neural network system and its implementation to the chosen application or process.

   The scope of study is to review each component inside the system such as background, overview, history and terminology, network applications, multilayer feed-forward neural networks, neural networks error calculations and as well as any other related agenda.

2. To develop a prediction model of gas consumption using artificial neural network system.

   The scope of study is to design and simulate prediction model using MATLAB software. The system must applied neural network forecasting value which is relationship between the variables being forecasted and variables used to produce the forecast, as well as the distribution of forecast errors.

CHAPTER 2
LITERATURE REVIEW

This chapter explains the concepts and theories involve in this project. It also justifies some of the decision that has been made in executing this project.

2.1 Gas Consumption in Real Life

Gas companies often face many challenges in the business of supplying gas to their customers. The challenges are forecasting the gas consumption for short, medium and long term (known as the sendout). The pattern of gas consumption is non-linear and varies according to several factors. Those factors may include nature changes, human demand and even geography factor. In order to adapt to the pattern or forecast the outcome, an extensive research need to be done to determine the inputs that largely affected the consumption and analysis the output behaviour.

2.2 Artificial Neural Network and Gas Consumption

Artificial neural network and gas consumption application is not a new thing among the researchers. The experts use this method for about a decade to forecast the gas consumption on various period of time. Adjustment and modification has been done since then to improve the result in term of reducing the error of the forecast and real value. Throughout the process, the author will explore and explain the use of artificial neural network while showing the significant to application process.

2.3 Artificial Neural Network

Neural network is one of the most popular methods used in the gas consumption. Neural network is defined as an information processing system that has been developed as a generalization of the mathematical model of human cognition. Neural network is chosen because of its capabilities to mimic human like performance on pattern recognition and classification. Neural network are highly efficient in classifying condition patterns of a system as normal or faulty, for gas consumption. It is highly efficient in detecting patterns and regularities in the input data. The application of neural network requires less restrictive assumptions as the structure input data.

In an article [12], it stated that a neural network system used in foreign exchange to select trading strategies, earned an average annual profit of 18% on a US $1 million position while a conventional system using moving averages earned only 12.3%. In Japan, Fujitsu Ltd., Kawasaki and Nippon Steel Corp. have developed a neural network based system that monitors a steel production process. The system has returned far better results from its predecessor and reduced cost by several million dollars. This shows that neural network can be used in any industry especially gas industry to handle processes and forecasting due to its adaptive and predictive characteristic.

2.4 Input Data Selection

Data selection has been very crucial in forecasting gas consumption behaviour. Among previous research, there are numbers of input that highly affected gas consumption. The selected inputs are:

2.4.1 Daily Temperature

Temperature is the most significant factor that effecting gas consumption because most gas is used for residential, commercial and industrial heating. The daily temperature inversely proportional with gas consumption. Figure 1 below shows normalized daily gas consumption and the average daily temperature of Milwaukee, WI [16].

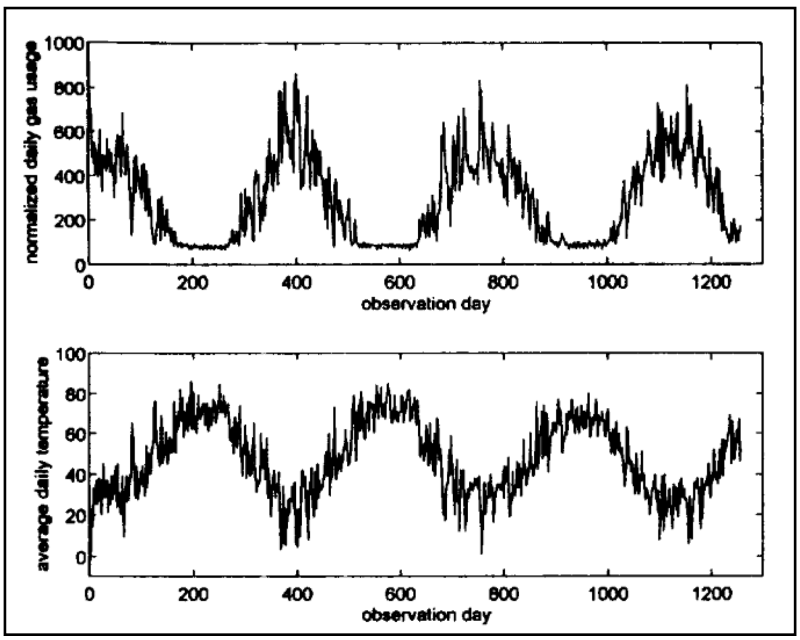

Figure 1: Comparison of Average Daily Temperature and Gas Usage in Milwaukee

From the graph, it is shown that temperature clearly affects the amount of gas usage and should be among the major factors of gas consumption. The temperature does not necessarily should be the exact temperature of day forecast but also can be the temperature of the next days (forecast weather) and previous days temperature. The amount of temperature selected depends on process and sensitivity of the system.

2.4.2 Wind Speed

From article [14, 15, 16], wind speed is one of the factor affecting gas consumption. A building loses more heat on a windy day than on a calm day. Although wind contributes to the process, wind speed is extremely hard to predict and including it in the model would only compromise the accuracy of the results.

2.4.3 Day of the Week

Day of the week symbolize essential factor to the process system. During weekdays, the industrial companies may use significant amount of gas for commercialize purposes. At the end of the week (weekend), some companies may not operating thus the gas consumes will drop to significant amount [14, 15, 16]. Other factors regarding days that matter are festival celebration and holidays.

2.4.4 Previous Day(s) Sendout

Sendout or gas usage itself plays an important role for input of gas consumption prediction modelling. In order to forecast the future, details of history need to be look out as the references. For short-term gas consumption prediction, the amount of changes during weekdays is not too much but in other related problems the changes is significant [15, 18].

2.4.5 Number of Customers

Although number of customers might be the least significant among other factors, it is still contributing to gas consumption. By knowing number of customers and their average usage, the supplier should aware of total consumption of gas. Number of customer might decrease on weekend due to company policy whether to operate or close the process [18].

2.4.6 Pressure

Boyle's law states that, *at a constant temperature, the volume of a given mass of gas varies inversely with pressure.* For two states of pressure (P_1, P_2) and two corresponding volumes (V_1, V_2), this is stated mathematically:

$$P_1 \cdot V_1 = P_2 \cdot V_2$$

From the equation, we could positively claim that a pressure does giving a variation in gas volume. Thus, pressure should be a strong selection for input data for gas consumption.

2.4.7 Calorific Value

Calorific value is amount of heat generated by a given mass of fuel when it is completely burned. It is measured in joules per kilogram. Calorific values are measured experimentally with a bomb calorimeter. For gas, the calorific value is used to calculate the transported amounts of energy, as this is what the consumers pay for, and not the delivered volume of gas. The calorific values for settlement purposes are checked and approved to ensure that the basic data are correct.

2.5 Data Preprocessing

Preprocessing means modification of the data before it is applied to neural network. By doing this, it transforms the data to make it suitable for neural network [18]. Some method use scaled numerical values and transform text values into numerical values. The ranging for input columns used is [-1, 1] and scaled as given equation:

For numerical values;

$$SF = (SRmax - SRmin)/ (Xmax - Xmin)$$

$$Xp = SRmin + (X-Xmin)*SF$$

Where X is the numerical data taken from the column, Xmin and Xmax is maximum values of data from the column, SRmin and SRmax are upper and lower scaling range limits, SF scaling factor and Xp is processed value.

For categorical values;

Other preprocessing also been done for categorical values. Encoding of categorical columns is encoded as binary encoding method. For example day factor written in Monday, Tuesday, Wednesday and etc. Monday is represented as [1,0,0,0,0,0,0] and Tuesday as [0,1,0,0,0,0,0]. Target column scaling range depends on the activation function of output layer.

2.5.1 Temperatures Forecast Preprocessing

There has been a discussion of temperature preprocessing as temperature play the crucial part of forecasting. There might have been various method of temperature preprocessing but the approach that will be discussed is robust and suitable for practical solutions in the real-world applications [17].

(A) Corpus of Historical Hourly Temperatures

This method is based on large corpus of historical hourly temperatures for each day around the year. Daily profile need to be validated before inserting to the corpus. No missing values or outliers could be let in. After gathering the data we might see a different between daytime and night time. There are also days when it is quite flat through 24 hours, or days even colder than night. The method is to find the value that have the most similar historical day in the corpus. A more stable approach is to find more than one most similar day and compose results as a (weighted) average.

(B) Similarity

The inputs still base on the large corpus but this time the values that been looking for are the most similar average, minimum and maximum temperature. The similarity function is defined as below:

$$f_{similar}(Avg_1,Min_1,Max_1,Avg_2,Min_2,Max_2) =$$
$$\{3*(Avg_1 - Avg_2)^2 + 2*(Min_1 - Min_2)^2 + 2*(Max_1 - Max_2)^2\} / (3+2+2)$$

Other than that, when searching for similar days in the corpus, we need to use at least 12 previous hourly values for the immediately preceding day and all the available hourly values from the current day. By doing this, we got more precise similarity and previous similarity data function filtered into:

$$f_{similar}(Avg_1,Min_1,Max_1,Tprev_1[],Avg_2,Min_2,Max_2,Tprev_2[]) =$$
$$\{3*(Avg_1 - Avg_2)^2 + 2*(Min_1 - Min_2)^2 + 2*(Max_1 - Max_2)^2 + \sum (Tprev_1[i]-Tprev_2[i])^2\} /$$
$$(3+2+2+12)$$

2.5.2 Error Management

The quality of prediction is strongly dependent on the quality of inputs. A great effort therefore focused on cleaning the temperature data.

The common spotted error cases are:

a) Single missing value
b) Single outlier
c) Period of missing values
d) Repeated values (jammed sensor)
e) Deviation from the mean (sensor needs to be calibrate)

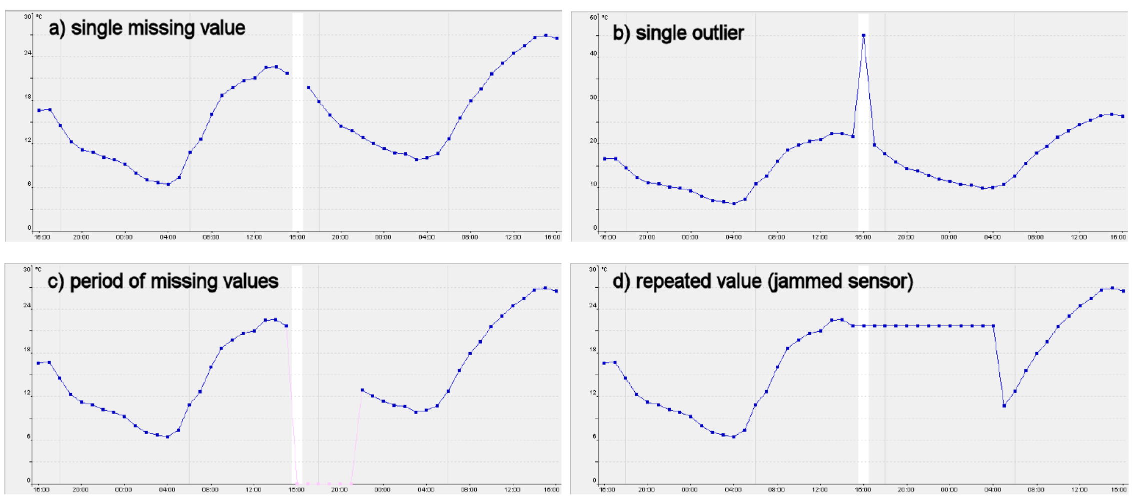

Figure 2: Various Kinds of Invalid Data

After detecting invalid data, we can fill the gap with the computed ones using the same approach as the forecast temperature. As the avg/min/max can no longer be rely on because of missing values, we can use the second similarity function to fill in the gap.

2.6 Method of Implementation

2.6.1 Feed-forward Neural Network

There are various methods in using neural network and the most popular method is feed-forward neural network with sigmoid nodes and trained with backpropagation learning rule. Feed-forward usually use three-layer without feeding back the output. The first layer consists of training inputs that determine by the user. The second layer is the hidden layer or neuron layer where the prediction interpretation take place and the third or last layer is the output layer sum up by the neuron of previous layer.

The goal of partition of the first hidden layer neurons is to aggregate the input data of each type independently from the rest of input data. This information is the aggregated in the next layers, where no restrictions are imposed on connections to the previous layers [16].

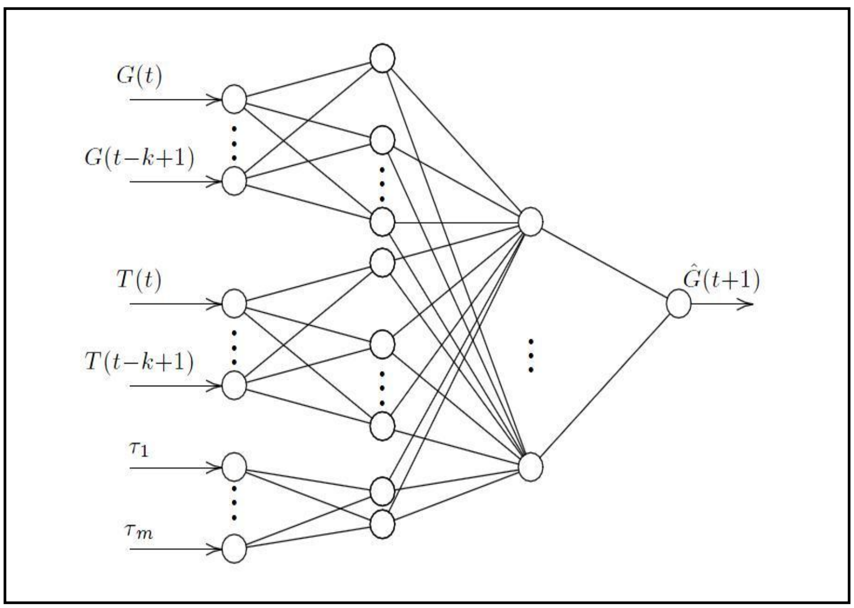

Figure 3: Feed-forward network architecture used in prediction

2.6.2 Recurrent Neural Network

Recurrent neural network are fundamentally different from feed-forward architectures in the sense that they operate, in addition to an input space, on an internal state space representing what already has been processed by the network [22]. It has almost similar structure as feed-forward in addition it has feedback loop of output to the input state use as training set. This method it is able to learn and predict time series that exhibit complex non-stationary behaviour.

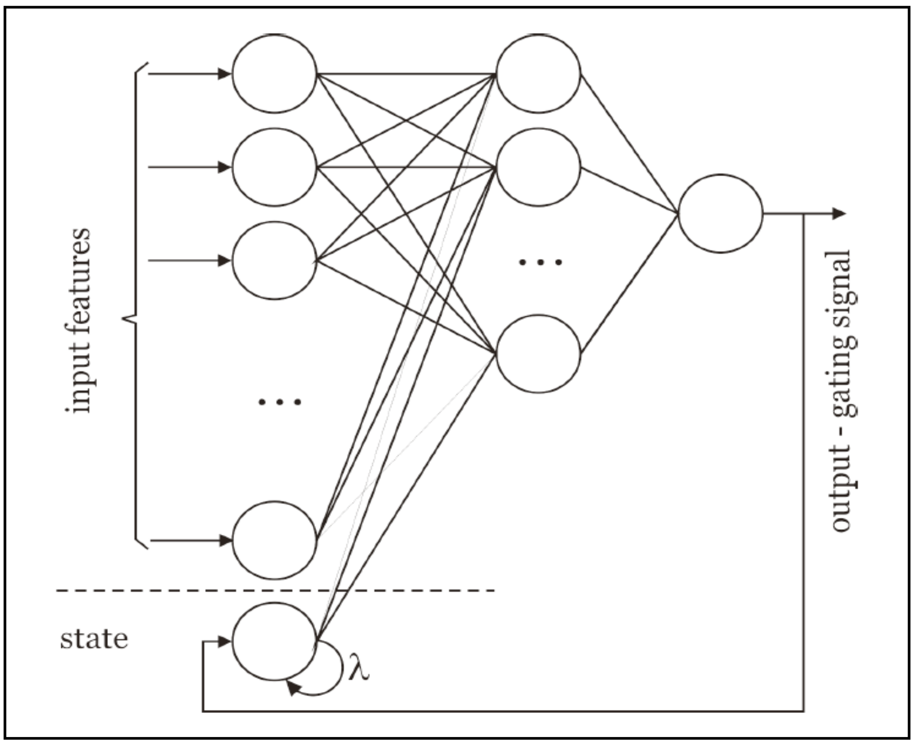

Figure 4: Recurrent network architecture used in prediction

CHAPTER 3
METHODOLOGY

This chapter will discuss in detail the project flow and its requirement to get the project started. It will show the sequence from start until the end on how to obtain the final result

3.1 Procedural Identification

Basically this project starts as soon as data collection has been received from the host company. Data received is confidential and not all personnel can access the data without authorization. The data received must be determined in term of input variable and its validity. This is crucial because there is no point continues to the next steps if the data is not valid.

The next step is to examine the data and select useful portions of the original data. The data received in bundle and there might be invalid data (missing or extreme values) included. A good or bad data division may influence the next steps. After selecting the best data, we have to select and define a model structure. This is the part where various techniques could be applied as the executioner. This is continued to next step where for each technique chosen must be structured to produce a reasonable and good result. The model's properties then examined and the result evaluated.

If the result is not optimum, the steps must be loop back to either data selection or the technique chosen. The process is been summarize to the flow chart in Figure 5:

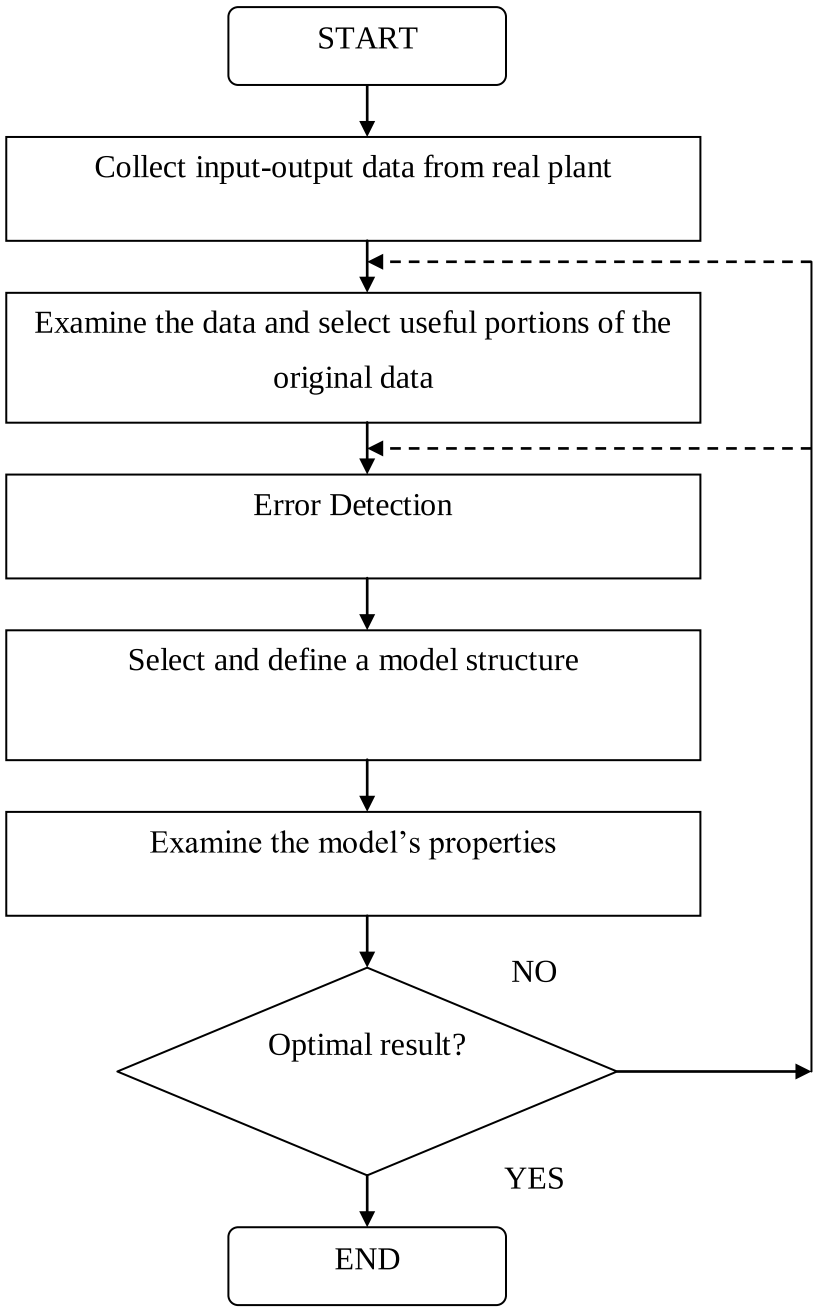

Figure 5: Procedure to determine a neural network model

3.2 Tool and Equipment Required

For this project, the tool required is software called MATLAB. Inside MATLAB, various tools will be used such as Neural Network Toolbox and Graphical User Interface (GUI) Toolbox to develop complete prediction software for gas consumption. Simulations are conducted to predict gas consumption while subsequently repeat the procedure using different method or inputs. Result will be analyzed and discussed for its characteristic.

3.3 Research

A throughout research were done through internet, public books and journals to collect all available information regarding the neural network system and gas consumption application in terms of its process and working principle. Further studies were done to identify the best inputs and its preprocessing. These are vital information as it will determine the prediction of gas consumption.

3.4 Project Activity

The first meeting was held on 2nd of March 2009 to discuss on this respective project. The person involves including the author supervisor, Dr. Rosdiazli and two engineers from PETRONAS Gas Berhad (PGB). The meeting was conducted on 9.00 am until 1.00 pm. They were several major things discussed during that meeting including the basic model diagram of the system and agreement on project milestone. On basic model diagram, we discussed on how the flow of an input and output mapping should be. This included all the parameters or variables correlated with artificial neural network system to produce the output desired.

On project milestone, several activities have been agreed on time of accomplishment. This includes budget presentation, beta version presentation, final model presentation and commissioning. Although this is only the first draft of the milestone, we expected the project should be finished in one year time. This meeting also symbolizes the “kick-off” of the project.

The author also involves in a talk regarding the artificial neural networks by one of the post graduate student who heavily involves in this line of project. The talk basically discussed on mathematical aspects of neural network including its implementation using MATLAB software. The author tried to familiarize with the coding on how to connect the data with the computational coding and will try to apply it to the project.

On 10th of March 2009, the author start to use the software tool, MATLAB, to run the system coding using the data received. The data received from PGB on daily basis and per hour recorded data. On 6th of April 2009, a simulation using neural network toolbox also has been done using the input data. The result will be discussed more on results and discussion chapter.

On August until October 2009, the author was trying to develop an error detection system to detect for an error in given data set. The model should be working in Simulink by using block diagram to implement it. All the result will be discussed more on results and discussion chapter.

CHAPTER 4
RESULTS AND DISCUSSION

This chapter will discuss about the results that been obtained by following the sequence of procedural in methodology. Each result will be discussed thoroughly and decision taken is justified.

4.1 Data Examination

Data collected will be pooled under one section and divide depending on the neural network trainers. The data collected are in hourly basis in a range of current 3 years. Although there are a lot of data pooled, not all the data suitable for the training as invalid data might be located in between the useful portion of data. The obvious part of data that was corrupted will instantaneously be left out.

Data normalization is next after the first examination. Normalization is the process of removing statistical error in repeated measured data. A statistical error is the amount by which an observation differs from its expected value; the latter being based on the whole data from which the statistical unit was chosen randomly. Outliers will also be left out from training and validation data. Although some set of data can be deleted with naked eyes, there are still various errors inside the large group of data that need to be considered. So a system of error detection will examine the data to detect the bad data before it will be used in the training and validation data.

4.2 Error Detection Model

In this model, there will be three types of error detection which includes check range system & data filtration, freeze data and set of zeros data.

4.2.1 Check Range System

Simulink has been a crucial tool in MATLAB. Simulink used in this project to create a check-range system to fit into the system. Depending on the customer demand, the check-range system should be versatile to react to any process values exchange. In every process line, there will be a process value operating range.

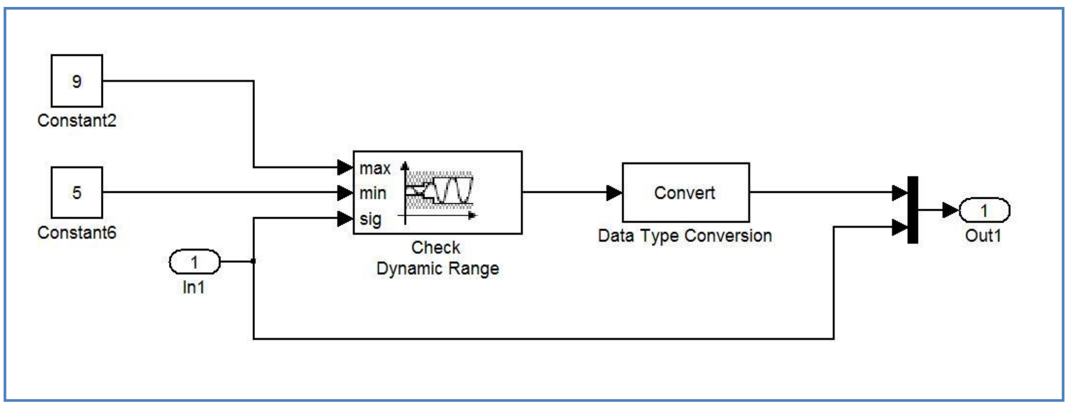

Figure 6: Simulink Block Diagram for Check Range System

From the diagram, we could see a flow of process input feed into a subsystem and produce a result. From left to right, there are basically three types of block been used for this subsystem.

Constant Block

- A parameter value can be defined in this block to be executed as reference while process running.

Data Type Conversion Block

- Convert the input of data type and scaling of the output. It has two possible goals. One goal is to have Real World Values of the input data and output to be equal. The other goal is to have the Stored Integer Values of the input and output to be equal. In this case, this block always comes after Check Dynamic range block diagram.

Check Dynamic Range Block

- This block will assert that one signal always lies between two other signals. The first input is the upper-bound signal while the second input is the lower-bound signal. The third input is the test signal where the signal input that feed into the system.

4.2.1 .1 Check Range System Output Graph

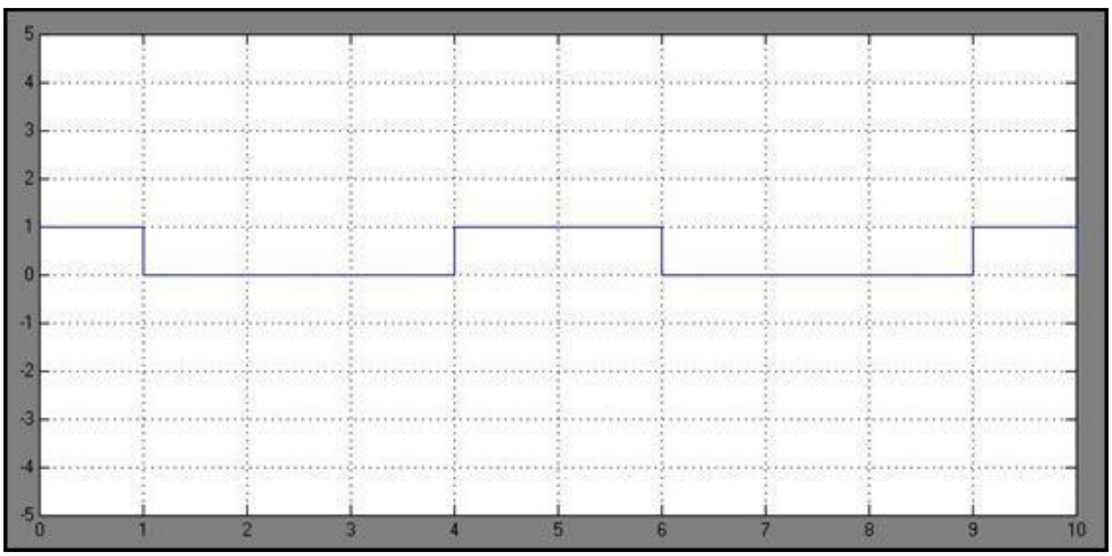

Figure 7: 10 units check-range system output

From the graph plotted, we could see there are 10 units of input given in a series of time and each values been evaluated whether in or out of range. For this case, the triggering part using “0” and “1” as reference which is widely used in digital system. From the graph, the low-value, “0”, represents the value which is out of the range while the high-value, “1”, is for the value that within the range.

However, the significant of this subsystem is to detect out of range value of the inputs. So it is necessary to convert the value of output in reversal manner so that the high-value always representing an alarm or caution action. In order to do the reversal, the author use logical operator which is NAND gate (NOT gate and AND gate) to accomplish it. There are two inputs of NAND gate which are the system input and constant “1”. The constant acts as parameter to validate the reversal process. The result can be seen from Figure 8:

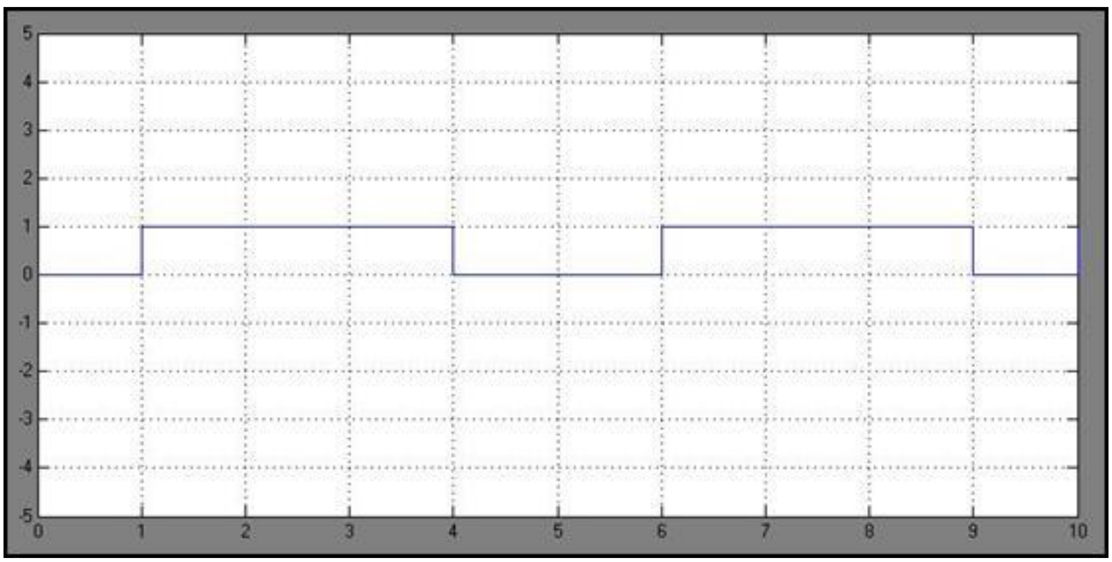

Figure 8: 10 units check-range system output (with NAND gate)

In order to implement the system for an alarm-triggering system, the author want to simplify the output graph by detecting the increase of value only rather than all the triggered values. This will hopefully work as the switch by taking the initial value only to activate the alarm. The picture of rising value of the same set of input can be seen from Figure 9:

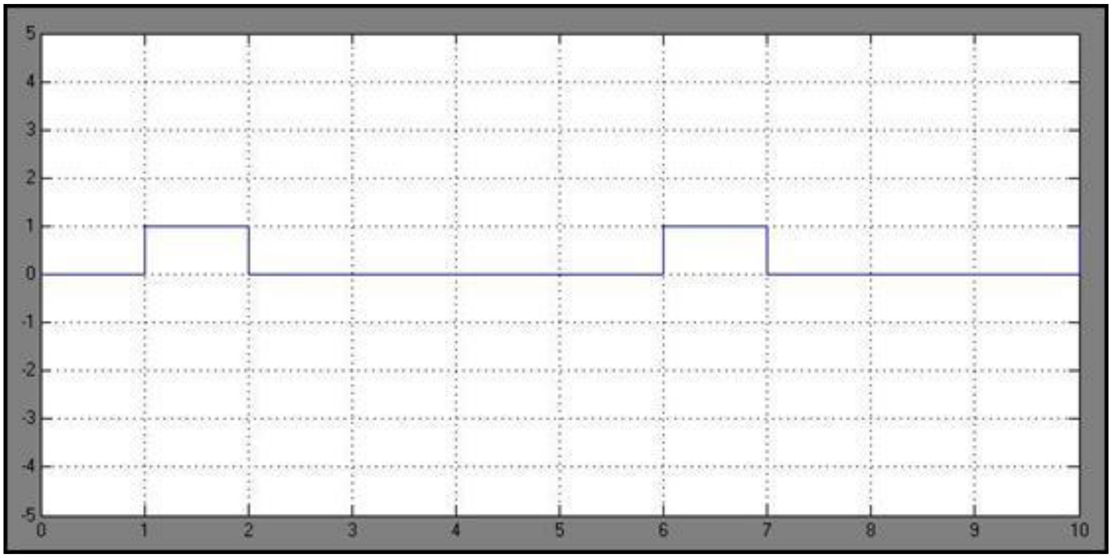

Figure 9: 10 units check-range system output (Rise Detector)

4.2.2 Data Filtration

Selection of data is very important especially to this project. The data selected should fulfill the criteria in order to produce accurate result. For this case, the author wants to emphasize on the parameters value that stay outside the range of operating condition. Some values from the field may have an error because of faulty of instruments, power supply problem, data freezing and other problems that may occurred. The easiest way to eliminate or extract the false value from the data sheet is by replacing it with the nearest valid value. As example, a set of data was collected from real-time value and the value of the data strictly monitored. If the data start to reach non-logical state, the data from non-logical state will be replaced by the last valid data of real-time. This applied to all data that not stable or out of range until the data come back to normal state.

To apply that, a block diagram using Simulink library has been used to create the subsystem. The diagram below shows the connection of the blocks (Figure 10).

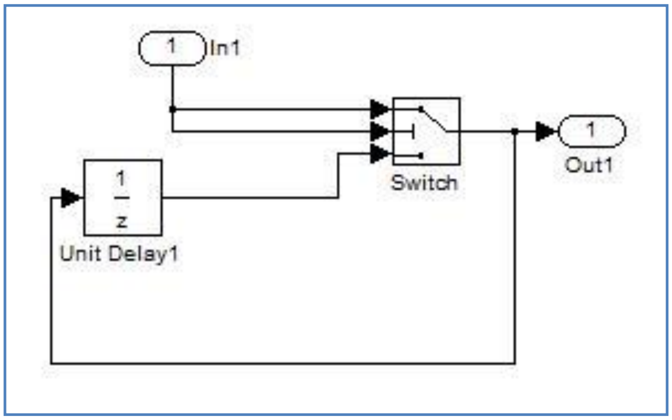

Figure 10: Upper-Bound Filtration

The diagram above is an Upper-Bound filtration Simulink diagram. The connections consist of two major blocks which are the switch and unit delay. The connections in the diagram work as a filter for low value data and replace it with nearest valid data.

Unit Delay

- Sample and hold with one sample period delay. It takes the sample of switch output and delays it by a period of time for feedback switch input. Details will be discussed more under Switch section.

Switch

- The switch consists of three inputs and operating under conditional manner. The inputs numbered 1 to 3 from top to bottom. The first and third input is data ports while the second input is the control port. To ensure the data taken not below the threshold value, the inputs connected parallel to each other for the first and second input while criteria for passing first input changed to "second input (control port) always greater than threshold value". The threshold value can be change at the setting options.
- The third input basically work as "back-up data" in case the input data did not fulfill the criteria to pass through. Third input data always the last valid data that pass through the switch.

- If a valid data feed into the system, the data will go to first port and second port simultaneously. To let the data from first port pass through, the data at second port (same data as first port) must meet the criterion which is "greater than threshold value". As the data is true, the data from first port will pass through as true data.
- If an invalid data feed into the system, the data still follow the same process as valid data except at the second port where the criterion not been fulfilled. The first input could not pass through and replace by the third input which is last valid input from first port. The initial condition value set to zero.

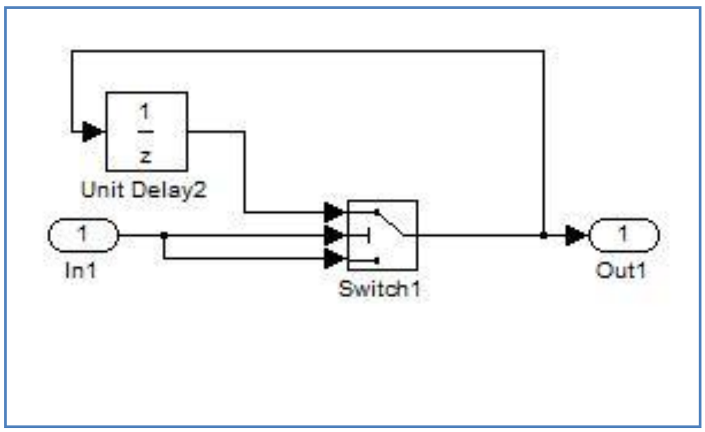

Figure 11: Lower-Bound Filtration

After done filtering the upper-bound data, another diagram also has to be made for lower-bound filtration to ensure the output data is in the range. Lower-Bound filtration is basically a reversal of Upper-Bound filtration process. The block diagram use also similar except the connection of properties. Lower-bound filtration cut off the high value data up to certain limit. The deleted data also will be replaced with the last valid data.

The connections set up for this diagram different in position for inputs port. The delay unit which previously set up at third port now changes to first port. The second and third port now received the value from data input.

At the control port (second port), the threshold value must be entered and the condition to let the first input pass through is “always greater than the threshold”.

- If a valid data feed into the system, the value is connected to second and third port. Thus, the control port would determine whether or not to let the first port pass through. Valid data will not pass the control port criterion for Lower-Bound filtration and let the third port pass through. The data that pass through the switch should be in the range of operating condition after passing Upper and Lower filtration.
- If an invalid data feed into the system, the value should fulfill the criterion of control port which is “always greater than threshold”. So, it will let the first port pass through which eventually take the previous valid data as the current data.

4.2.2 .1 Data Filtration Output Graph

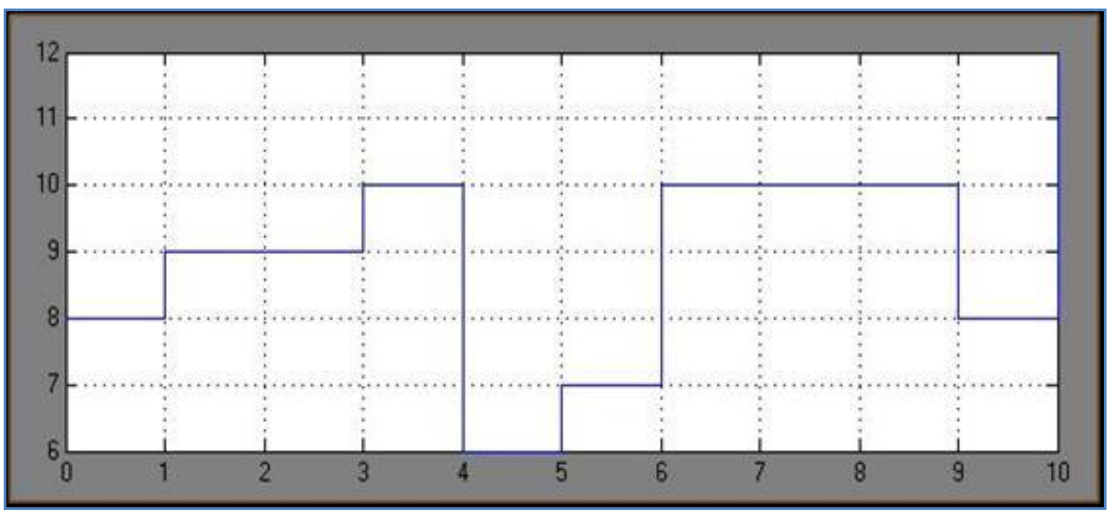

Figure 12: Upper-Bound Filtration Output

A set of data created on purpose to test the Upper and Lower-bound Filtration. A combination of high and low out of range value included to see whether or not the system could detect the presence of any of these values. For this test, a value ranging from 0 to 10 uses as the taken data but the valid data would range from 6 to 8.

The author decides to let the data pass through the Upper-Bound Filtration first before the Lower-Bound Filtration. The order of filtration does not affect the output data as long as it passes both type of filtration. As we could see from figure 12, there are few data that well below the range and been replaced with valid data especially on timeline of 6 to 9.

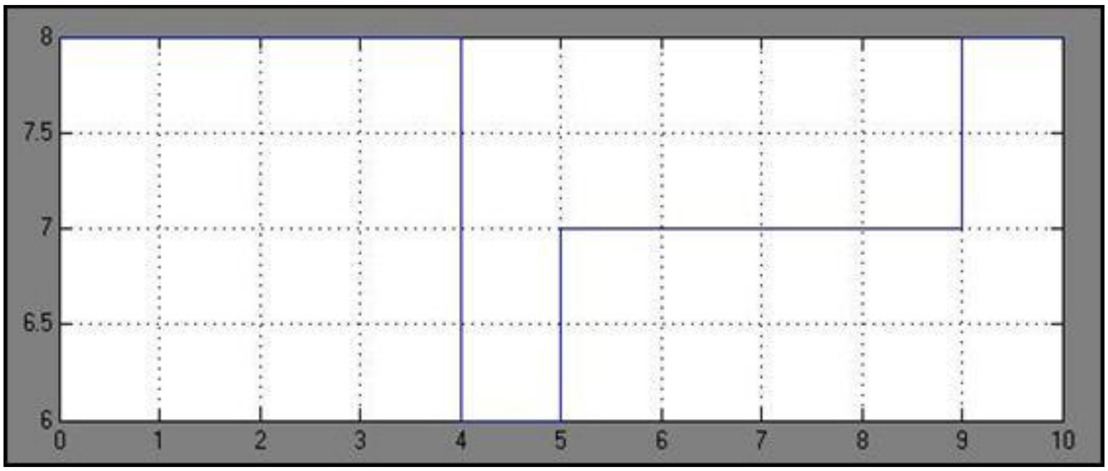

Figure 13: Upper and Lower-Bound Filtration Output

After done with the Upper-Bound Filtration, the cycle needs to go to Lower-Bound Filtration to complete the cycle. From Figure 13, we could see the value of high range will not exceed 8 and the value of low range will not drop below 6. This satisfies the system objective to let the out of range value eliminated by replacing it with the last valid data. This thus will help the system to get more efficient input to be trained or validate using the neural network system and producing more accurate result.

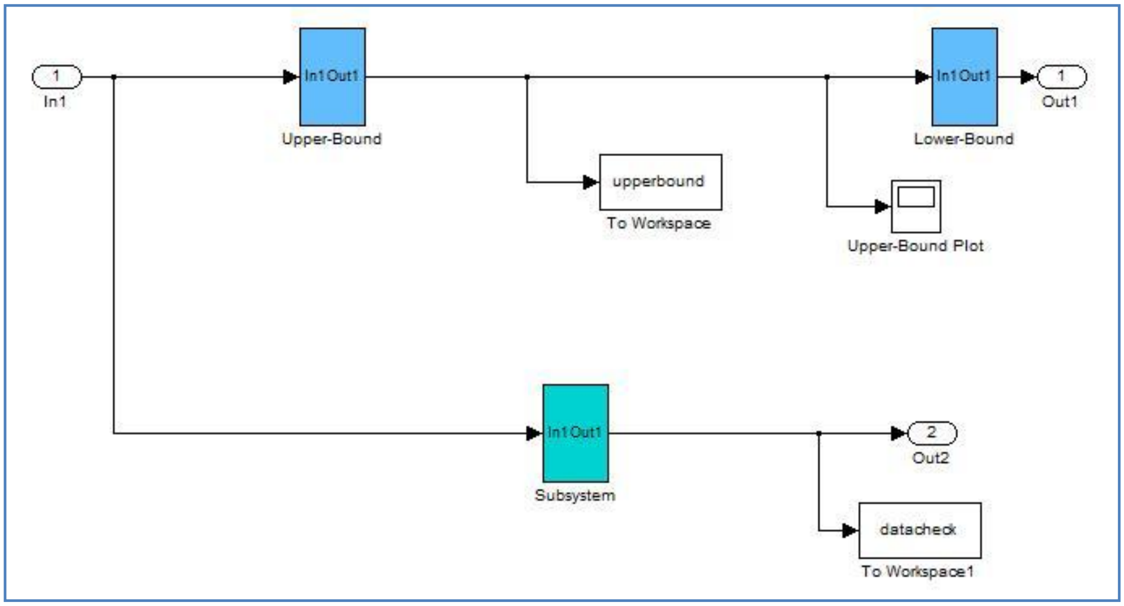

Figure 14: Check Range and Upper/Lower-Bound Filtration System

The diagram from Figure 14 shows the complete illustration on where the input goes using the system. The input will go to check range system and data filtration simultaneously to extract data and information. Both systems are very useful to the user. Check range system will make the user alert to any irrelevant changes and adapt to it while data filtration give the best possible range for data input to the main system.

4.2.3 Data Freeze

Data freeze is a common data error which is cause by instrument malfunction that will lead to redundant process value in a series of time. It is very crucial to detect this kind of data because we do not want the system later on to learn from a false set of data. In order to do avoid it, a subsystem created by Simulink to detect the freeze data and exclude it from the data set.

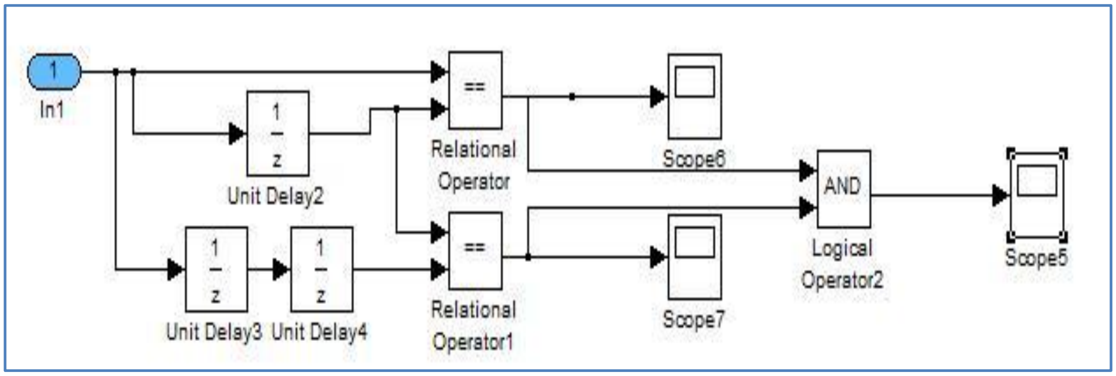

Figure 15: Data Freeze Detection System

From the Figure 15, we could see the block diagram for freeze detection which consists of three major parts which are the unit delay bloc, relational operator and the AND gate.

Unit Delay

- Sample and hold with one sample period delay. It takes the sample of current data and delays it by a period of time. Using multiple block will multiple the delays of the data. Data will be paired up with previous data before been feed into the relational operator block.

Relational Operator Block

- Relational operator block for this subsystem use two inputs which current (t) and delayed value (t-1). Equal value will trigger the block and gives output of "1" and not equal value will give value of "0". The output then will be carried out to the AND gate.

AND Gate

- The AND gate is a digital gate that will be triggered as true if only both the inputs are true. If both the inputs were "1", then the output will be "1" otherwise "0". This is the last output to determine whether there is freeze value in the system or not.

4.2.3.1 Data Freeze Output Graph

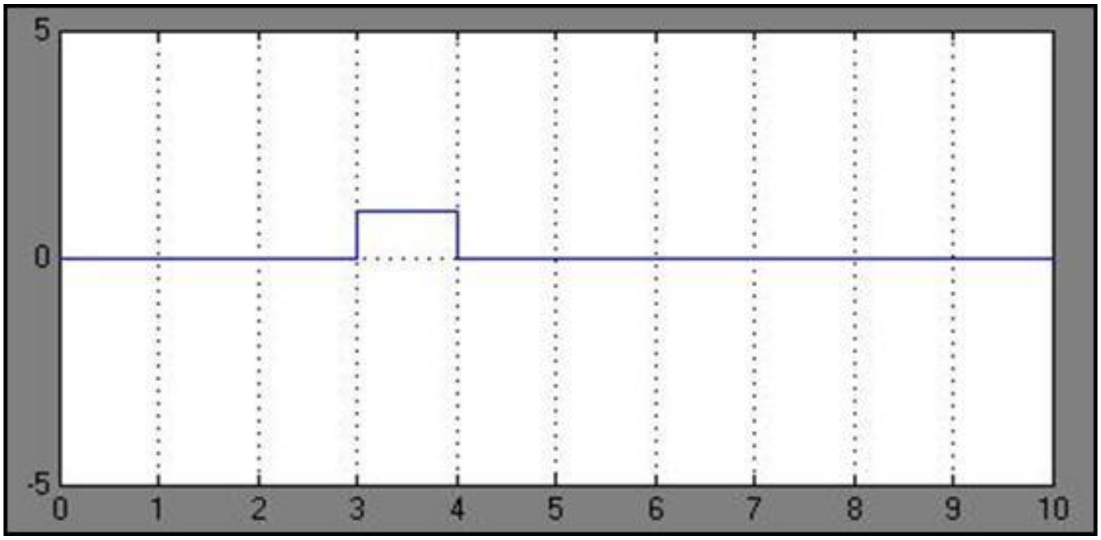

Figure 16: Data Freeze Detection Output

From the Figure 16, we can see from the time series of 10 units there is a triggered value in time 4. It shows that the freeze start at time 4 because the value exactly equal to previous and after value. This program works to detect after 3 times of repetition. The number of repetition before detect can be adjusted depending on the demand.

The number of graph is separated for each input because for each input parameters as they were taken from different devices. So it is necessary to detect for individual input rather than all together.

4.2.4 Row of Zeros (Missing Data)

Row of zeros or missing data commonly occurs inside an industrial plant. This is because of miscommunication between control room and field devices due to bad wiring or device failure. It may not occur very often because it may recover after sometimes but the loss of communication will give no value of data from input parameters. So the program must able to detect any instant value that loss from data.

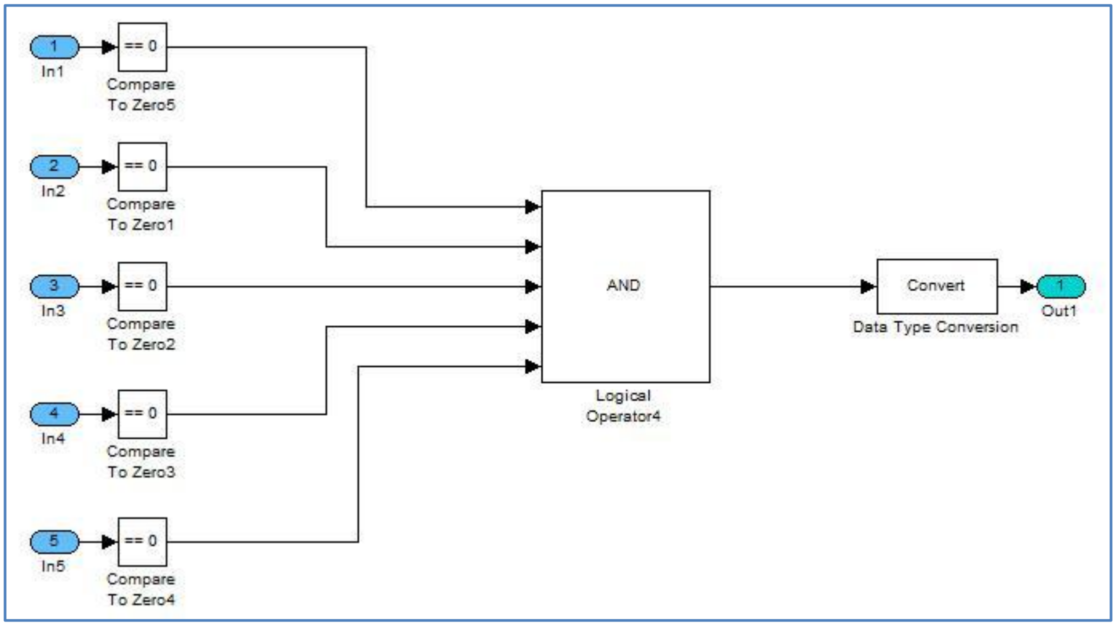

Figure 17: Row of Zeros Detection System

From Figure 17, we can define the system by two major blocks which are the parameter comparison block and the AND gate.

Compare to Constant Block

- As the missing communication input value always return as zero, this block use the zero value as a reference figure for any value of input from the set of data. In the diagram, all five inputs will be connected to each comparison block and any data equal to zero will trigger the block as “1”, otherwise “0”. The inputs then connected to next block, AND gate.

AND Gate

- The AND Gate connected with all five inputs and produce one output. To triggered the output as “1” or true, all five inputs must be equal to zero or state “1” as the inputs, otherwise the output will “0” or false. Depending on the system requirement, not all five inputs should be connects to the AND gate.

4.2.3.1 Row of Zeros Output Graph

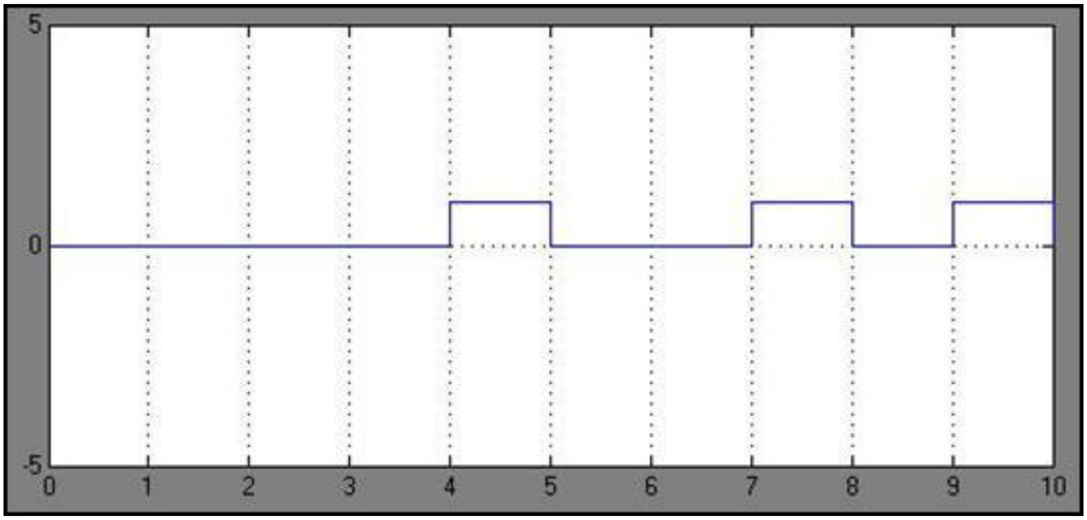

Figure 18: Row of Zeros Detection Output

Figure 18 shows the output graph on a time series of 10 units. We could see detection occurs on row 5, 8 and 10. We can conclude from the system that each input for each rows that triggered are equal to zeros. If larger set of data given, we can easily detect the rows that got zeros value by detecting from time series value.

4.3 Neural Network Model Structure

In this part, we will discuss on layer of neural network, the activation function and neuron that will be used in this model.

Neural Network Layer

- In this project, we will use one hidden layer for inputs given and one layer of output. The hidden layer for inputs and output connected as one hidden layer in between it. The output use only one neuron for hidden layer while the major part for hidden layer comes after the input layer. So in sequence, the neural network structure should be input layer, hidden layer and the output.

Input & Output

- The correlation between input and output for this network is very straightforward which is placing the input to produce an output.
- There are five inputs use for this project which are the gross volume (*Vg*), temperature (*T*), pressure (*P*), calorific value (*CV*) and specific gravity (*sg*). Of all other inputs, these five inputs give a significant indicator of output changes.
- The output for this system is gas prediction in term of energy. The inputs given will provide the prediction of gas energy in comparison with actual energy.

Activation Function

- Activation function use for training for in this project are continuous log-sigmoid function (logsig), continuous tan-sigmoid function (tansig) and pure-linear.
- For input layer, logsig and tagsig will be used alternately as activation function.
- For output layer, all function; tansig, logsig and pure-linear will be used alternately.
- The best input/output correlation will be plotted to find for the best model.

Number of Neuron

- Number of neuron is significant in order to get the optimum result; in this case, the objective is to find the lowest root mean square error (RMSE). The number of neuron should not be too small or too big.
- Neuron will be tested for a range of number and the optimum value of performance will be plotted and observed.
- Number of neuron with the best performance will be used in a model.

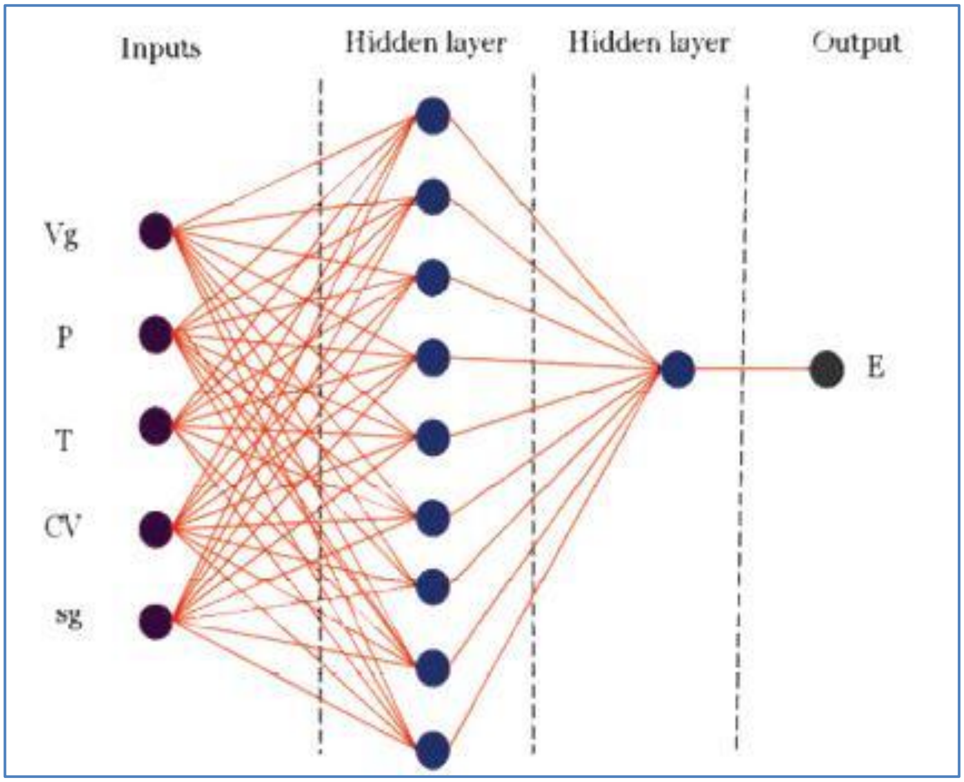

Figure 19: Neural network for gas prediction structure

4.4 Examine Neural Network Model Properties

In this part, we will discuss on training algorithm, number of neuron selected and proportional data training. Discussion and selection will be based on performance or root mean square error value (RMSE).

Table 1: Neural network training with different algorithm and activation function

Learning Algorithm	Activation Function (Hidden Layer)	Activation Function (Output)	RMSE (Training Data)	RMSE (Validation Data)	No Of Epoch(s)
Levenberg-Marquardt (LM)	Tan-Sigmoid	Pure-Linear	30.4008	117.5005	33
		Tan-Sigmoid	42.8156	173.7837	66
		Log-Sigmoid	191.1625	268.9092	169
	Log-Sigmoid	Pure-Linear	28.7533	67.6118	60
		Tan-Sigmoid	36.5324	105.5351	26
		Log-Sigmoid	102.7337	180.2886	20
Bayesian Regulation (BR)	Tan-Sigmoid	Pure-Linear	25.9090	48.9495	69
		Tan-Sigmoid	134.0851	159.4038	15
		Log-Sigmoid	214.3603	149.5438	10
	Log-Sigmoid	Pure-Linear	153.1591	238.9517	53
		Tan-Sigmoid	28.4655	74.1926	38
		Log-Sigmoid	94.5694	169.3381	48
Resilient Backpropagation (RP)	Tan-Sigmoid	Pure-Linear	133.4265	226.9997	77
		Tan-Sigmoid	147.4132	171.6051	145
		Log-Sigmoid	214.9055	236.5062	162
	Log-Sigmoid	Pure-Linear	178.4400	265.7708	22
		Tan-Sigmoid	177.6864	259.4646	20
		Log-Sigmoid	120.1617	203.8692	128
Gradient Descent With Adaptive Lr (GDA)	Tan-Sigmoid	Pure-Linear	207.8986	332.9721	61
		Tan-Sigmoid	1110.5044	1550.0122	1
		Log-Sigmoid	764.9414	538.4926	1
	Log-Sigmoid	Pure-Linear	401.6037	316.5956	71
		Tan-Sigmoid	262.5530	401.6037	1
		Log-Sigmoid	262.5530	401.6037	1

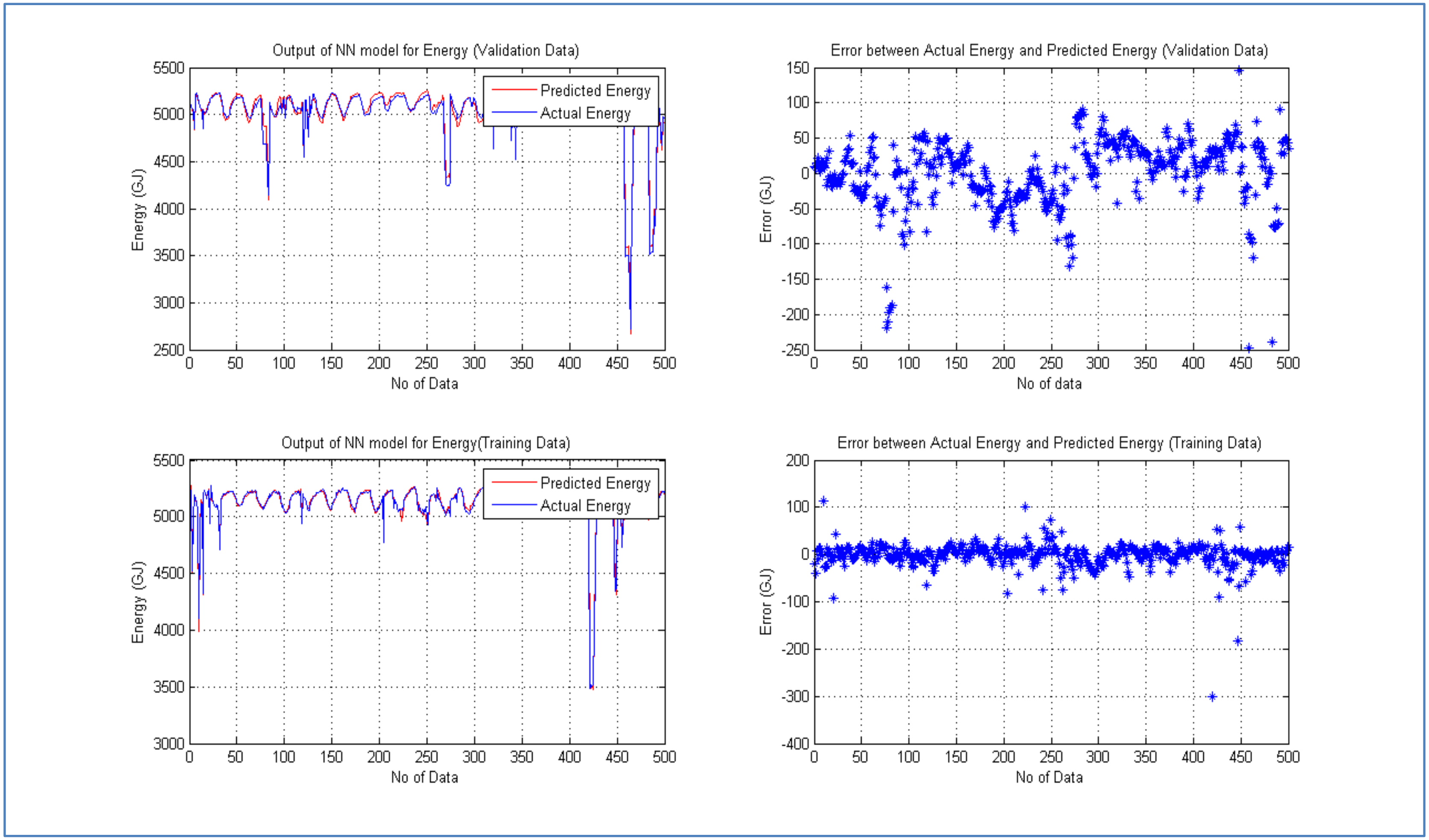

Figure 20: Bayesian Regularization with tan-sigmoid/purelinear training

From the observation of Table 1, it can be seen that the learning algorithm LM and BR gives more promising result rather than the other learning algorithms. The RMSE obtained from RP and GDA is not consistent, shows no real trend and gives quite high RMSE. Between the LM and BR, the RMSE is low and has not much different, just that using the BR, the RMSE is lower. Meanwhile, using the activation function tan-sigmoid for hidden layer always give better result than log-sigmoid which is due to data distribution lies between -1 and 1 that coincide with the tan-sigmoid properties with smaller error.

Out of these four learning algorithms, the most compromising result is obtained from the learning algorithm of Bayesian Regularization with activation function of tan-sigmoid for hidden layer and pure-linear for output layer. The RMSE is 48.9495 for the validation data and 25.9090 for the training data.

Table 2: Neural network Bayesian Regularization algorithm with different neurons

RMSE (Training Data)	RMSE (Validation Data)	No Of Neuron(s)	No Of Epoch(s)
62.9119	126.7948	1	23
34.8994	66.9562	2	25
26.2926	59.9573	3	29
28.7506	51.0344	4	40
24.2049	77.2284	5	108
28.9207	61.7735	6	52
25.0222	58.7301	7	74
24.8542	50.8466	8	59
28.5592	45.6301	9	42
28.7865	59.7451	10	76
24.6342	87.2330	11	91
33.8665	47.0096	12	48
28.3120	63.5375	13	63
38.2580	53.4371	14	42
28.8960	79.5008	15	83
32.7589	66.6003	16	49
27.7694	81.1022	17	53
33.7145	59.3281	18	30
34.6118	75.1326	19	48
32.6868	118.9493	20	36

One other objective of this investigation is to determine the number of neurons that gives the least RMSE value for both training and validation data set. The constants made are the learning algorithm; Bayesian Regularization, the activation functions tan-sigmoid for hidden layer and pure-linear for output layer. Neural network is tested and modelled from 1 neuron to 20 neurons. This model is tested out starting with one neuron up to number of neurons that gives the least value of RMSE and is also tested for more than the optimum number of neurons.

The number of neurons manipulated only for the first hidden layer because the second hidden layer is made constant to one neuron. It can be seen from the Table 2 that almost all neuron gives similar RMSE in training but the different become more obvious during validation.

Selection should be based on low RMSE during training and validation. The range or different between training and validation also should be low because we do not want too large margin for error during the training and validation. Neuron 4, 8 and 9 gives a promising result of performance. 9 neurons are chosen because it got a low RMSE on training and validation. The range of error between training and validation also is the smallest of all neurons.

Table 3: 9 neurons neural network Bayesian Regularization algorithm with different data proportion

Data Proportion	RMSE (Training Data)	RMSE (Validation Data)
Data X (Training 75%, Validation 25%)	28.9126 28.9265 28.4282	44.3166 47.8464 42.4326
Data Y (Training 50%, Validation 50%)	34.4935 33.7804 34.6104	85.4835 50.9066 69.1360
Data Z (Training 25%, Validation 75%)	20.4643 21.3008 25.1562	76.2648 82.3196 100.2729

Data X from Table 3 is observed and able to get a low RMSE value which proves that the more training data provided, the less RMSE resulted. This also means that better energy prediction model can be developed using neural network with higher accuracy and reliability.

Data Y shows RMSE for both training and validation, the value of RMSE increases from Data X. It has small increases on training and the obvious part come on validation where the RMSE increase almost double from Data X.

For Data Z, by looking at the trending, the RMSE for validation is higher compared to other division of data. The RMSE for training at the other hand is smaller compared to the other two data division results. This is because smaller number of data for training gives less information for validation process. Thus, it will increase the RMSE for the validation part. It is recommended that the 9 neurons neural network energy prediction model to be chosen as the best model with 75 percent training data and 25 percent validation data.

CHAPTER 5
CONCLUSION

In this chapter, the author will discuss the overall conclusion and recommendation to be made for this project.

5.1 Conclusion

The model is developed by creating a functional system for data error detection and investigating a few parameters which are the learning algorithm, the activation function, the training and validation data division and the number of neurons. For data error detection, all the data will be detected for its operating range, freeze data and row of zeros data (missing data). From the neural network developed on predicting the energy consumption with 5 inputs, the most simple and reliable model suggested and decided is a neural network model with 9 neurons for first hidden layer and 1 neuron for second hidden layer trained using Bayesian Regularization algorithm. The activation function used is the tan-sigmoid for hidden layer and pure-linear for output layer. To get more accurate prediction, it is suggested that the number of training data should be higher than the number of validation data. As for this investigation, the proposed division is 75 percent of data is used for training and 25 percent for validation. A reliable neural network model for energy prediction is essential in forecasting accurately and precisely. Implementing this model will not only help the distributor in gas transmission smoothness but also might bring profits to the distributor.

5.2 Recommendation

For future improvement, it is highly recommended to use different method of neural network for gas prediction. A recurrent neural network (feedback neural network) should be an ideal method of implementing it. Learning algorithm also should be considered as there are a lot of algorithm can be used for prediction model. Data error detection also could be added such as spiking data within range of operating range and a lot more. This could help on improving the set of data for neural network training purpose.

REFERENCES

[1] Salvador Gil, Jorge Deferrari (2006). Generalized Model of Prediction of Natural Gas Consumption. Journal of Energy Resources Technology. Distribution Division of ENERGAS (Natural Gas Regulatory Agency of Argentina).

[2] Edward R. Jones (2004). An Introduction to Neural Networks. Visual Numerics, Inc.

[3] Christopher Bishop (1995). Neural Networks for Pattern Recognition, Oxford.

[4] B. Hammer, T. Villmann (2003). Mathematical Aspects of Neural Networks.

[5] Warren S.Sarle (1994). Neural Networks and Statistical Models, SAS Institute Inc., Cary, NC, USA.

[6] Dejan Ivezic (2006). Short-Term Natural Gas consumption Forecast, Faculty of Mechanical Engineering, University of Belgrade.

[7] Andes Krogh (2008). What are artificial neural networks? ; Nature Biotechnology Volume 26, February 2008.

[8] Simon Haykin (1999). Neural Networks, A Comprehensive Foundation, 2nd Edition, Prentice Hall, Inc.

[9] Kate A. Smith (1999). Neural Networks for Combinatorial Optimization, INFORMS Journal on Computing, Vol. 11, No. 1, Winter 1999.

[10] Michael I. Jordan, Christopher Bishop (1996). Neural Networks, ACM Computing Surveys, Volume 28 No.1.

[11] Anangela Garcia, Shahab D. Mohaghegh (2004). Forecasting US Natural Gas Production into year 2020, Society of Petroleum Engineers Inc.

[12] “Neural Networks at Works”, IEEE Spectrum June 1993.

[13] C. G. Atkeson (1991). ‘”Using locally weighted regression for robot learning," IEEE Robotics Conference on Automation and Robotics.

[14] Domagoj Peharda, Marko Delimar, Sven LonEariC (2001). “Short term hourly forecasting of gas consumption using neural networks”, Faculty of Electrical Engineering and Computing, University of Zagreb.

[15] Alireza Khotanzad, Hassan Elragal, Tsun-Liang Lu (2000). “Combination of Artificial Neural-Network Forecasters for Prediction of Natural Gas Consumption”, IEEE Transactions on Neural Networks, Vol. 11, No. 2, March 2000.

[16] Ronald H. Brown, Paul Kharouf, Xin Feng, Luc P. Piessens, Dick Nestofl (1994). “Development of Feed-Forward Network Model to Predict Gas Consumption”, Department of Electrical and Computer Engineering Marquette University, Milwaukee, WI.

[17] Milan Šimůnek, Emil Pelikán (2008). “Temperatures Data Preprocessing for Short-Term Gas Consumption Forecast”, Institute of Computer Science, Academy of Sciences of the Czech Republic.

[18] Recep Kızılaslan and Bekir Karlık (2008). “Comparison Neural Networks Models for Short Term Forecasting of Natural Gas Consumption in Istanbul”, Department of Industrial Engineering and Computer Engineering, Fatih University, Istanbul, Turkey.

[19] Nguyen Hoang Viet, Jacek Mandziuk (2003). “Neural and Fuzzy Neural Networks for Natural Gas Consumption Prediction”, Institute of Fundamental Technological Research, Polish Academy of Sciences and Faculty of Mathematics & Information Science, Warsaw University of Technology, Warsaw, Poland.

[20] Ronald H. Brown and Iftekhar Matin (1995). “Development of Artificial Neural Network Models to Predict Daily Gas Consumption”, Department of Electrical and Computer Engineering Marquette University, Milwaukee, WI.

[21] R. L. Zhang, D. J. Walton, W. D. Hoskins (1993). “Prediction of Natural Gas Consumption”, Department of Computer Science University of Manitoba Winnipeg, Manitoba, Canada.

[22] Petr Musilek, Emil Pelik´an, Tom´aˇs Brabec, Milan ˇSim°unek (2006). “Recurrent Neural Network Based Gating for Natural Gas Load Prediction System”, Department of Electrical & Computer Engineering, University of Alberta, Edmonton and Institute of Computer Science, Academy of Sciences of the Czech Republic.

[23] Alireza Khotanzad and Hassan Elragal (1999). “Natural Gas Load Forecasting with Combination of Adaptive Neural Networks”, Electrical Engineering Department, Southern Methodist University, Dallas, Texas

APPENDICES

```
%-----------------------------------------------%
%Neural Network model by Vishal Nanji Patel
%February 2009
%2 layer network with normalization [0,1]
%-----------------------------------------------%
%Revision 2 by Mohammad Sholeh Bin Abdullah
%02 November 2009
%-----------------------------------------------%
%Clear workspace and command window
clear all;
close all;
clc;

load datajanuarymay;

%load data from workspace
x = datajanmay(:,1:5)';
y = datajanmay(:,6)';

%-------------------------------------------------------------------
-----%
%divide data into TRAINING and VALIDATION
%-------------------------------------------------------------------
-----%
%get the number of input and number of data
train_data = 250; %number of TRAINING data
validation_data =750; %number of VALIDATION data
numofvar = size(x,1); %number of input
numofout = size(y,1); %number of input

for m=1:numofvar
    for n=1:train_data
    x_t(m,n)=x(m,n);
    end
end

for m=1:numofvar
    for n=1:validation_data
    x_v(m,n)=x(m,n+train_data);
    end
end

for m=1:numofout
    for n=1:train_data
    y_t(m,n)=y(m,n);
    end
end
```

```
for m=1:numofout
    for n=1:validation_data
    y_v(m,n)=y(m,n+train_data);
    end
end

%-------------------------------------------------------------------
-----%
%prepocess the input and output [-1,1]
%-------------------------------------------------------------------
-----%
[x_t1,x_s1] = mapminmax(x_t);   %INPUT training data
% [y_t1,y_s1] = mapminmax(y_t);   %OUTPUT training data
[x_v1,x_s2] = mapminmax(x_v);   %INPUT validation data
% [y_v1,y_s2] = mapminmax(y_v);   %OUTPUT validation data
%maximum and minimum value of TRAINING data
t = minmax(x_t1);
%-------------------------------------------------------------------
-----%
%set network properties
%-------------------------------------------------------------------
-----%
%number of neurons for layer 1 and layer 2
neuron_1 = 9; %number of neurons for layer 1
neuron_2 = 1; %number of neurons for layer 2

%network and parameters
net=newff(x_t1,y_t,neuron_1,{'tansig','purelin'},'trainbr');
net.trainParam.show = 50;
net.trainParam.lr = 0.1;
net.trainParam.epochs = 1000;
net.trainParam.goal = 0.001;
net=init(net);

%set the weights and biases to 0
%set the weights for 1st layer to 0
for m=1:neuron_1
    for n=1:numofvar
    w_1(m,n)=1;
    end
end
net.IW{1,1}=w_1;

%set the weights for 2nd layer to 0
for m=1:neuron_2
    for n=1:neuron_1
    w_2(m,n)=0;
    end
end
net.LW{2,1}=w_2;
```

```
%set the bias for 1st layer to 0
for m=1:neuron_1
    b_1(m,1)=0;
end
net.b{1}=b_1;

%set the bias for 2nd layer to 0
for m=1:neuron_2
    b_2(m,1)=0;
end
net.b{2}=b_2;

%checking the weights and biases (make sure all are 0)
net.IW{1,1}; %weights of 1st layer
net.LW{2,1}; %weights of 2nd layer
net.b{1}; %bias of 1st layer
net.b{2}; %bias of 2nd layer
%--------------------------------------------------------------------
-----%
%train the network
%--------------------------------------------------------------------
-----%
[net,tr]=train(net,x_t1,y_t);
%--------------------------------------------------------------------
-----%
%simulate the network
%--------------------------------------------------------------------
-----%
%simulate the network with TRAINING data
xtest_t = mapminmax('apply',x_t,x_s1);  %prepare input data for
training
ytrain = sim(net,xtest_t);  %simulate the network
% ytrain1 = mapminmax('reverse',ytrain,y_s1); %descale the output
%calculate the different between the actual and predicted
temperature value
etrain=y_t-ytrain;

%simulate the network with VALIDATION data
xtest_v = mapminmax('apply', x_v, x_s1);    %prepare input data for
training
yvalid=sim(net,xtest_v); %simulate the network
% yvalid1 = mapminmax('reverse',yvalid,y_s1);%descale the output
%calculate the different between the actual and predicted
temperature value
evalid=y_v-yvalid;
%--------------------------------------------------------------------
------%
%plot graph
%--------------------------------------------------------------------
------%
```

```
%plot the actual and predicted pH from VALIDATION data
subplot(2,2,1);
plot (yvalid,'r');
hold on;
plot (y_v,'b');
xlabel('No of Data');
ylabel('Energy (GJ)');
title('Output of NN model for Energy (Validation Data)');
legend('Predicted Energy','Actual Energy');
grid on;




%plot the different between the actual and predicted pH from
VALIDATION data
subplot(2,2,2);
plot(evalid,'*');
xlabel('No of data');
ylabel('Error (GJ)');
title('Error between Actual Energy and Predicted Energy (Validation
Data)');
grid on;




%plot the actual and predicted pH from TRAINING data
subplot(2,2,3);
plot (ytrain,'r');
hold on;
plot (y_t,'b');
xlabel('No of Data');
ylabel('Energy (GJ)');
title('Output of NN model for Energy(Training Data)');
legend('Predicted Energy','Actual Energy');
grid on;




%plot the different between the actual and predicted pH from
TRAINING data
subplot(2,2,4);
plot(etrain,'*');
xlabel('No of Data');
ylabel('Error (GJ)');
title('Error between Actual Energy and Predicted Energy (Training
Data)');
grid on;
%-------------------------------------------------------------------
------%
```

```
%error analysis
%--------------------------------------------------------------------
------%
%error analysis for the VALIDATION data
fit_valid = (1-norm(evalid)/norm(y_v-mean(y_v)))*100 %fit value
rmse_valid = sqrt(mse(evalid)) %mean square error
index_valid = (sum((evalid).^2)/sum((y_v-mean(y_v)).^2))*100 %index
value
correlation = corrcoef (y_v,yvalid)
%actualValid_predictedValid = [y_v' yvalid1']

%error analysis for the TRAINING data
fit_train = (1-norm(etrain)/norm(y_t-mean(y_t)))*100 %fit value
rmse_train = sqrt(mse(etrain)) %mean square error
index_train = (sum((etrain).^2)/sum((y_t-mean(y_t)).^2))*100 %index
value
correlation = corrcoef (y_t,ytrain)
%actualTrain_predictedTrain = [y_t' ytrain1']
```