AUTOMATIC TEXT SUMMARIZATION USING ANFIS

FONG JIA KANG

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Artificial Intelligence)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL: AUTOMATIC TEXT SUMMARIZATION USING ANFIS

SESI PENGAJIAN: 15/16

Saya FONG JIA KANG

(HURUF BESAR)

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

_____ SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

_____ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

__/__ TIDAK TERHAD

(TANDATANGAN PENULIS)
FONG JIA KANG

Alamat tetap:
57, Jalan Seremban 1,
Taman Seremban, N.S.D.K.

Tarikh: 16/8/16

(TANDATANGAN PENYELIA)
YOGAN JAYA KUMAR
Nama Penyelia

Tarikh: 16/8/16

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM) ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

DECLARATION

I hereby declare that this project report entitled

AUTOMATIC TEXT SUMMARIZATION USING ANFIS

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT: __________ Date: __________

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Artificial Intelligence) With Honours.

SUPERVISOR: YOGAN JAYA KUMAR Date: 16/8/16

DEDICATION

I dedicate my final year project report to my family and friends. I would like to show my gratitude to my supervisor, Dr Yogan Jaya Kumar for helping me and guiding me in finishing this project. I would like to show my deepest gratitude to my parents for being patient towards my bad behavior while completing this project. To my family members, they have been very supportive and encouraging while giving constructive ideas for this project. On top of that, I would like to dedicate this report to all my friends for being helpful in any ways.

ACKNOWLEDGEMENT

I would like to show my gratitude to the following individuals in helping me throughout the project. Firstly is my supervisor, Dr Yogan Jaya Kumar and the project evaluator, Dr Sakinah who have given me a lot of constructive comments, guidance, and support throughout the project. Besides, I would also like to thank my parents for being so supportive and thoughtful either in a passive or active support. They were so helpful by helping me to reduce the stress accumulated throughout this project and many more. Lastly, I would like to thank my class mate for helping me to solve either technical or logical problems faced while completing this project.

ABSTRACT

The information overload faced by today’s society has created a big challenge for people who want to look for relevant resources from the internet. Automatic text summarization is a field of soft- computing which can help humans in dealing with this abundance of information by extracting the main idea in the documents.

This report attempts to compare the performance of the main technique proposed in this report, ANFIS with Neural Network and Fuzzy Logic. The main technique proposed in this report is ANFIS as it is the hybrid technique of Neural Network and Fuzzy Logic. There are many past researches on text summarization using Neural Network and Fuzzy Logic, but not many researches have been done using ANFIS on text summarization.

Besides, there are limitations of Fuzzy Logic to be applied in text summarization. Fuzzy Logic requires human experts to create a set of fuzzy rules for the fuzzy inference system to work. As such, ANFIS is proposed as the main technique to be experimented on text summarization application to overcome the limitation of Fuzzy Logic.

ABSTRAK

Informasi yang terdapat pada laman web kini telah mewujudkan satu cabaran yang besar bagi masyarakat yang ingin mencari sumber-sumber yang berkaitan daripada internet. Rumusan teks secara automatik daripada bidang pengkomputeran industri yang boleh membantu manusia dalam menangani maklumat yang terdapat di dalam internet dengan mengeluarkan idea utama dalam dokumen tersebut.

Laporan ini akan membandingkan prestasi teknik utama yang dicadangkan dalam laporan ini, iaitu ANFIS dengan Rangkaian Neural dan Logik Kabur. Teknik utama yang dicadangkan dalam laporan ini adalah ANFIS kerana ia adalah teknik hibrid yang didapati daripada gabungan Rangkaian Neural dengan Logik Kabur. Terdapat banyak kajian lepas yang berkaitan dengan rumusan teks menggunakan Rangkaian Neural dan Logik Kabur, tetapi tidak banyak kajian telah dilakukan dengan menggunakan ANFIS untuk menyediakan rumusan teks.

Selain itu, Logic Kabur mempunyai batasan tertentu untuk diaplikasikan dalam rumusan teks. Logik Fuzzy memerlukan pakar manusia untuk mewujudkan satu set peraturan kabur bagi sistem inferens kabur untuk berfungsi. Oleh itu, ANFIS dicadangkan sebagai teknik utama yang akan dieksperimenkan atas bidang teks rumusan untuk mengatasi batasan Logic Kabur.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.0 Introduction

Text summarisation has become a field for researchers to explore as information from online sources has become a new trend for everyone to search for information. A brief summary about the text is useful for humans to extract important information about the text before going through the whole document. Thus, a summary can aid humans in understanding the main idea discussed in the text provided. Automatic summarisation is introduced so that computers can create summaries without having linguistics experts to provide a complete summary. In this project, an experiment on automatic text summarisation using Adaptive Neural Network Fuzzy Inference System (ANFIS) will be conducted to be compared with text summarizers using Neural Network and Fuzzy Logic. With the abundance of text material in the internet, such as e-book, online journal, and e-library, automatic text summarization is required for interpreting the text information as the information in the Internet are usually more than required. Artificial Neural Network (ANN) is a predictive model inspired by biological neural systems. A multilayer perceptron, which is used to represent a neural network, consist of three layers: the input layer, output layer, and the hidden layer. A vector of variable values is presented to the input layer. To train the network, the network is supplied with the binary class of the output. By using the back-propagation technique, the weight for predicting the class of the output will be randomly initialized for all the inputs given. When the value from each input neuron arrived at the hidden layer, the resulting weighted values are summed up and are supplied to a sigmoid function in providing an output value. When the weighted values arrived at the output layer, the value of the neuron layer is again multiplied by a weight and is fed into the final sigmoid function, which is the final value of the output network.

1.1 Problem statement

Humans prefer to use imprecision when commmunicating. Computer could not understand Natural Language conversed by humans. Human experts could not create a summary as fast as a computer if the algorithm is trained smoothly. Overfitting may occur as the dataset contains all sentences in the document with at least five positive outputs (1) while the remaining outputs are negative (0). The problem of using Fuzzy Logic in creating summary sentence is Fuzzy Logic requires human experts to determine the membership function and the number of rules required to create the summary sentences. As such ANFIS is introduced to compare the performance with Fuzzy Logic and Neural Network so that the generating of summary sentences can be done without human experts. Two problems are encountered, that is to search for relevant documents through an abundance of text documents, and to absorb the relevant information from that particular text. Hence, automatic summarization is useful to select and extract key points from the relevant text. Text summarization is also useful since there are many news topic of similar idea or event being discussed at the same time. In this case, text summarization can help to give the overall idea of what have being discussed throughout all the news topic of similar idea.

1.2 Objective

- To perform classification of binary output using ANFIS.
- To generate summary sentence of 200 words using ANFIS model.
- To evaluate the generated summary of the 3 models using ROUGE Evaluation.

1.3 Scope

- This project will be implemented using MATLAB R2015b (64-bit) as this platform supports the extension of ANFIS command.
- Dataset will be stored as ".dat" extension file as MATLAB R2014b (32-bit) could not support ".xls" extension file.
- DUC 2002 Dataset are acquired from online sources. Hence, the dataset may contain missing value or noise which can affect the result of the project.
- Grid partitioning is not suitable for DUC 2002 Dataset as the number of input exceeds five, which will generate a large amount of rules that will slow down the computational time with respect to the ANFIS inputs.
- This project will be implemented using clustering method to generate rules and membership functions as there are 9 ANFIS inputs to be considered.
- ANFIS only supports first or zeroth order Sugeno-type FIS with a single output.
- FIS could not generate all rules required for feature extraction. Hence, human expert is required to generate the rules.

1.4 Project Significance

E-books, online journals or papers have recently become a new trend for students to do research. However, the publications of these online text is increasing vastly, resulting in humans could not find their resources required by them efficiently. Thus, this problem can be countered by using computer to summarise the text so that humans can have a brief summary before hand.

The main objective of text summarisation is to provide the main purpose of the text in a short but precise version of the original text. However, automatic text summarisors faced problem to capture the theme of the text as humans tend to include natural languange in their text. As such, text summary identifier using ANFIS is introduced in this project.

1.5 Expected Output

The development in industrial computing has resulted in more improvement on text summarization. Artificial Neural Network can be combined with Fuzzy Logic to improve the accuracy of a basic algorithm, such as clustering and decision making. The expected result of this project is to show an analysis of performance measures by comparing the proposed algorithm(ANFIS) with Fuzzy Logic and Neural Network used for predicting text summary. The outcome of this project should prove that ANFIS can perform better text prediction if compared to Neural Network and Fuzzy Logic algorithm used for predicting text summary.

1.6 Conclusion

Text summary provides a general statement of a document in enabling the reader to get an overview about the text. Hence, automatic text summarization is important for reader to save time on reading the whole text. However, human tend to include natural language while communicating which results in reduced quality of the text summarised. As such, ANFIS is introduced so that the quality of the text summarized can be improved.

CHAPTER 2

LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

Many researches have been done for text summarisation, including text summarisation using artificial neural network(ANN), genetic algorithm(GA), and vector space model(VSM). Each of these technique can produces summaries in different ways. ANN is trained and modified through feature fusion in discovering the key points of the text. GA is done by generating a set of sentences that maximize the optimality of the quality metrics. VSM is an approach where the sentences of the text are visualised in a N-dimensional vector, where the sentences are weighted according to the similarity of the text. However, there are limitations faced when these techniques are used in text summarisation. GA could generate bad chromosome which can reduce the effeciency of creating a good summary. ANN cannot be retrained which will lead to time consumption when adding a new data into the network. VSM cannot summarise long text as a VSM matches similarities to produce a summary. Thus, Adaptive Neuro Fuzzy Inference System(ANFIS) technique is proposed to evaluate which is the best technique for text summarisation.

2.2 Facts and findings

2.2.1 Domain

2.2.1.1 ANFIS

ANFIS is used in the field of decision making to predict the correct summary sentence according to the features provided. The datasets used to train the system are important in evaluating the performance of the system. Sugeno type ANFIS with clustering method will be designed and different datasets will be provided to train and test the system by using MATLAB. Sugeno system uses a mathematical function to define the rule based system. Sugeno systems are more compact and computationally efficient if compared to Mamdani system as Sugeno system can compute large dimension of input whereas Mamdani system can only compute a fixed number of input dimensions given by the membership function. ANFIS system allows the user to modify the parameters of the membership functions based on the data. The parameters are adjusted automatically by the neuro adaptive learning techniques like back propagation algorithm or hybrid method, which is a combination of back propagation and least squares method. These techniques allow the fuzzy inference system to learn information about the data set. During the learning process, the parameters of the membership functions will be changed. The computations of these parameters can be controlled by using the optimization procedure which is defined by the sum of squared difference between actual and desired outputs.

2.2.1.2 Neural Network

Neural network are predictive models which is useful to express a wide range of decision making problems. A multilayer perceptron (MLP) which is use to refer to a neural network, consists of 3 layers: the input layer, hidden layer, and the output layer. The features of each instance are presented in the input layer, and the target output is presented at the output layer. One hidden layer is usually sufficient for all problems. With these 3 layers, neural network can start the training process. Back propagation (BP) algorithm is chosen as the learning algorithm for multilayer feed-forward network. BP enables the reduction of errors between the training example target values and the network outputs. The algorithm follows the cycle as stated below:

1. Randomly initialize weights to the inputs
2. Compute differences between target output and predicted output
3. Compute average error
4. Propagate error and compute change in error after refining the weight
5. Refine the weight to reduce error

Overfitting may occur when the number of neurons used in the hidden layer is too much. The model would show a low performance when new input is given to the model. However, when the number of neurons used in the hidden layer is too low, the model would not be able to model a complex data. Cross validation method is used to measure the number of neurons to be used in the model.

2.2.1.3 Fuzzy Logic

Sentence extraction can also be done by fuzzy logic where fuzzy rules and fuzzy set are applied to select the summary sentences based on their features. Fuzzy logic uses the approximate reasoning technique to provide decision support for deciding whether the particular sentence is a summary sentence or not. The reasoning capabilities of fuzzy logic have overcome the problem of imprecision in human life, where much of the logic cannot be classified as binary values or multi-valued logic. The fuzzy set proposed by Zadeh is a mathematical tool to deal with uncertainty, vagueness and ambiguity. The feature extraction technique can be done by applying fuzzy rules to obtain the important sentences in the text. For example, the feature of sentence length should return high value if the sentence is long and vice versa. This is because long sentences carry more ideas in that sentence where else short sentences seldom carry any meaning towards the text. Each sentence is composed from 9 features. With all the features in the sentence, the score of each sentence is determined by using fuzzy scoring, which is composed of fuzzy rules and membership function. The fuzzy rules are in the form of condition rules IF-THEN. The membership function fuzzifies each score into 3 values that is LOW, MEDIUM and HIGH. Hence, the sentence can now be classified into sentence which is unimportant, average and important. For example IF (F1 is L) and (F2 is L) and (F3 is M) and (F4 is H) and (F5 is L) and (F6 is L) and (F7 is H) and (F8 is L) THEN (sentence is not important).

2.2.2 Existing System

Text summarization using Fuzzy Logic approach has been implemented by Pallavi in 2014 based on fuzzy logic scoring. Four stages of implementing the system are done, which is pre-processing, feature extraction, fuzzy logic scoring and sentence selection. Preprocessing is used to divide the document into sentences, removing stop words, and converting every word into its basic form. Feature extraction is used to give a score to the sentence by judging the title word, sentence length, and sentence position. Sentence with similarity, numerical data, and thematic words will also get high score as the sentence may reflect important statistics of the text. The scores for each sentence are then derived using fuzzy logic scoring. The sentences of the document are then arranged in descending order based on their fuzzy score. The sentence with the highest fuzzy score will be placed as the first sentence of the summary, followed by the second sentence in rank of the fuzzy score. Multi-document text summarization using Neural Network presented by M.KarthiKeyan&K.G.Srinivasagan in 2012 is based on selection of features. The network is trained based on the importance of the sentences in a paragraph according to a human reader. The network is trained using Mcculloch- Pitt's model as the linear threshold gate classifies the set of inputs into 2 binary classes. The input files are converted into a vector form and the output of the neuron is the weighted sum of the input. Training is done when the network is presented with some sample data and modification of weights is done for better approximate of the desired function.

2.2.3 Technique

2.2.3.1 ANFIS

ANFIS technique will be used as the main technique in conducting the project as ANFIS is the hybrid technique of FIS and Neural Network. ANFIS is selected as the main technique of this project as ANFIS fulfils the requirement of using it as a technique for text summarization. However, the main restriction of using ANFIS in text summarization is the limited number of inputs that can be processed. The is due to the process of grid partitioning which will increase the rules to be created with respect to the power of the inputs provided. This problem can be overcome by using clustering method to generate rule where rules are created according to the radius of the cluster. Sugeno fuzzy inference system is selected as Sugeno system defines a mathematical equation which can be used for predicting the output, where else Mamdani system compute a fixed number of input dimensions given by the membership function.

2.2.3.2 Neural Network

Neural network consists of 3 layers with an activation function. When the input exceeds the activation function, the neuron will fire and gives an output to the output layer. Patterns are presented through the input layer and the model will predict the output of the given patterns through the output layer. Since there are 9 features to be extracted, 9 inputs will be created to store the values of each feature. There are two phase in the process of neural network training, that is to identify the sentence to be selected as the summary sentence and the second phase is to rank the sentences from the highest score to the lowest score and select the top sentences reaching 200 and 400 words accordingly to produce the final summary. The first step is to train the network to learn which sentence should be included in the summary sentence. The network will learn the patterns of which sentence is taken as a summary sentence or vice versa. The activation function (threshold condition) will be the same as ANFIS so that result will not differ too much from ANFIS.

2.2.3.3 Fuzzy Logic

To obtain the summary sentence in the text, the text is break up into sentences. Numerical values for all features in the sentence are assigned and the summary sentence is obtained by matching all numerical values with fuzzy rules. The main steps for Fuzzy Logic to do text summarization are as follow:

1. Read the dataset into the Fuzzy Inference System.
2. Each sentence is supplied with 9 numerical values of features, where the values are computed from the content of the sentence.
3. The features are calculated to obtain a fuzzy score for all the sentences.
4. A set of highest score sentences are extracted as a summary.

2.3 Project Methodology

2.3.1 ANFIS

ANFIS is introduced for text summarization in this project. This method enables more number of inputs features to be included into the system using clustering method. This method groups similar document into clusters and each sentence in the documents are clustered according to the features. Sentence with the best score are selected as the summary for the document.

In order to perform ANFIS, the dataset have to be preprossessed first. The output of each sentence, which is either a summary sentence or null, has to be determined and included into the dataset. The instances in the dataset are then reduced for balancing the number of output with 1 and 0. This is to avoid overfitting where the system is trained with more outputs of 0 compared with 1. Features which are irrelevant for training are also removed in this step. The dataset are then loaded into the system and an analysis on the performance of ANFIS in predicting the output will be included in this project.

2.3.2 Neural Network

The dataset that have been preprocessed earlier when performing ANFIS method will be used for training the network. The trained network should be able to classify the type of sentence that is included as a summary sentence. The network will randomly permutate the dataset with 75% of training data and 15% of testing data. With the trained network, a new dataset will be supplied to the network. The trained network will now apply the pattern trained before to predict the scoring of the sentence. The sentence will be sorted in descending order, starting from the sentence with the highest score. After the sorting is complete, the system will insert the top sentences into the summary cell array. The system will be in a loop until it reaches a summary with 200 and 400 words.

2.3.3 Fuzzy Logic

The idea of a Fuzzy Inference System (FIS) is to select the fuzzy rules and the membership function. The performance of the FIS is affected by the selection of fuzzy rules and membership functions. FIS consists of four components: fuzzifier, inference engine, defuzzifier, and the fuzzy knowledge base. In the fuzzifier, crisp input (1 or 0) is fuzzified into fuzzy values by using a membership function to determine the fuzziness of that value. For example, IF (S1 is 1) and (S2 is 0), THEN

(F1 is 0.4445). After fuzzifying the inputs, the inference engine will refer to the fuzzy knowledge base which contains all the rules created to compute an output for all the fuzzy inputs derived earlier. In the last step, the fuzzy output from the inference engine are converted into a final crisp value (1 or 0) using the membership function to represent the final sentence score. Figure 1 shows the architecture of a FIS for summary generation.

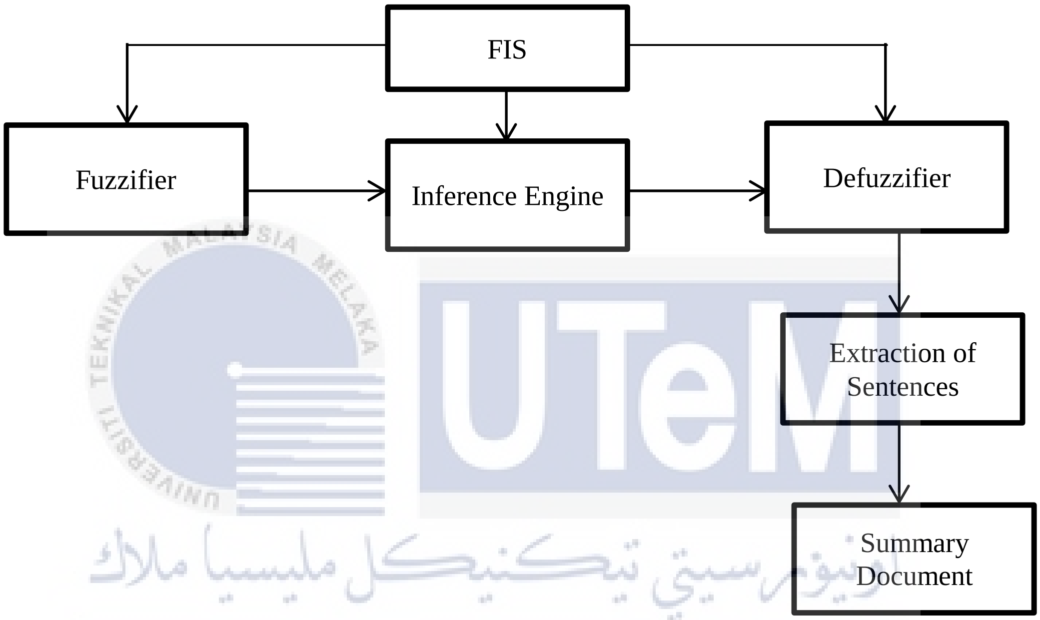

Figure 1: FIS architecture for Summary Generation

2.4 Project Requirements

2.4.1 Software Requirement

- MATLAB R2015b
- Notepad++
- MS Word
- WPS Spreadsheet

2.5 Project Schedule and Milestones

Five action plans are determined to accomplish the milestones stated below, which is the requirement, design, implementation, verification, and maintenance stage. To provide a better understanding on how these action plans work, the waterfall methodology is applied as this approach enables the development of this system in various stages. This approach will be discussed further at stage 3.

The requirement analysis shows us what is needed in order to develop the system. Software such as Matlab R2015b is required as the main developing software used in this project. A dataset is also required before developing the system as dataset are served as the training and testing data. Preprocessing of the dataset is required as there are features that are irrelevant which might influence the result of the data.

The second stage to be performed is to design the prototype of the system. The prototype includes the rough idea of how the system works and all the functions that will be included in the system. Functions that will be included such as creating a dialog box for user input, graphs, regression plot, and performance measures. Dialog box is for user to input the option for creating the FIS, such as the influence radius and the number of epochs. The training and testing graphs which will plot the target and the actual outputs against the number of samples enable us to see which predicted result varies from the actual output. The root mean square of the training and testing data will also be included in the graph for further analysis to be continued. Regression plot is used to show how much the predicted output varies from the target output. Lastly, performance measure which is the accuracy, precision, and f-measure, and recall of the predicted output will be included as the function of the system. The accuracy is used to calculate the overall effectiveness of a classifier. Precision is used to show how many samples that the system predicted with positive labels. Recall is used to show the effectiveness of the system to identify positive labels. F-measure is used to show the relations between data's positive labels and those given by the system.

The third stage is to implement a suitable development methodology for this project. Waterfall approach is chosen as stated above because this approach shows the feasibility of the project, such as how many milestones have been achieved and what should be done in the next stage. Implementing the waterfall system into this project development enables the development of the system in various stages as shown in Figure 2.

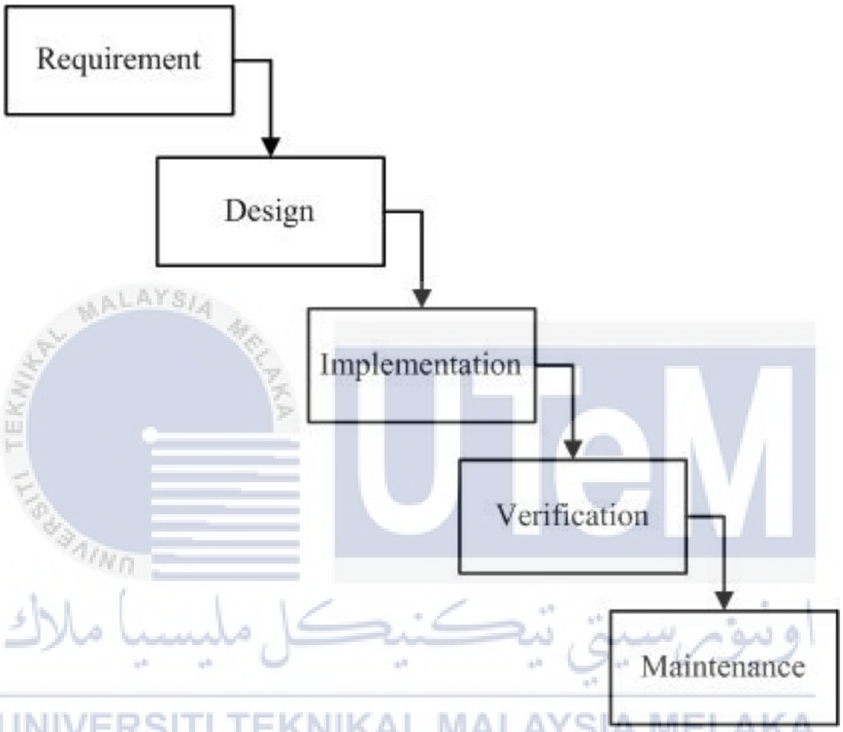

Figure 2: Waterfall Model Diagram

The fourth stage is to verify the system for avoiding overfitting. The model is trained till the results are obtained with minimum error. A proper training and testing data set is required as the testing data set will not validate the model if the training datasets are not selected properly. If the testing data set is completely different from the training dataset, then the model cannot capture any of the features of the testing data. Overfitting occurs when the prediction model is bias to a given output. Hence, testing data are required to prevent overfitting from occurring.

The last stage is to maintain the system so that the system would not fall into runtime and out of memory error. This is to allow the system to store a large amount of input data and generate a FIS to predict the output without crashing.

Table 1: Gantt Chart of the overall project

Task	Feb 21-27	Mac 4-10	Mac 11-17	Mac 18-24	Mac 25-31	Apr 1-7	Apr 8-14	Apr 15-21	Apr 22-28	May 1-7	May 8-14	May 15-21	June 1-14
Proposal confirmation													
Correction on proposal													
Studies on Literature													
Chapter1,2 submission													
System development													
Chapter 3 submission													
Experimental Testing													
Chapter 4 submission													
System demonstration													
Chapter 5 submission													
Result Analysis													
Chapter6,7 submission													
Presentations													
Final Report Due													

2.6 Conclusion

The goal of this chapter is to identify the milestones for this project. In order to achieve the milestones, five action plans are determined so that the project can be completed in time. The first action is to identify the requirements of this project. The result of this project is to create an analysis to compare which methods is the best for text summarization. The second action is to create a prototype on how this system functions. The prototype is a basic system that includes all functions in the system for further development. The third action is to implement a development methodology into this project. In this case, the waterfall approach is chosen as it is more suitable with the requirements of this project. The fourth action is to verify the fitness of this system. This can be done by using a testing set to test the performance of the system for preventing overfitting of the system. The last action is to maintain the system so that the system can store a large amount of inputs without crashing. This can be done by avoiding noise data which will interrupt the process of predicting the output.

CHAPTER 3

ANALYSIS

3.1 Introduction

The Analysis Phase is where the project lifecycle start. The requirements of the project are identified and are classified into more detailed project requirements. Gathering requirements is the main task in the Analysis Phase. Gathering requirements can be done by asking the users what they need and writing their answers down. This phase consists of a group of repeatable processes that utilize certain techniques to manage the requirements of the project, depending on the complexity of the system. This phase consists of four basic steps, which is elicitation, validation, specification, and verification. The analysis phase can be illustrated as shown in Figure 3:

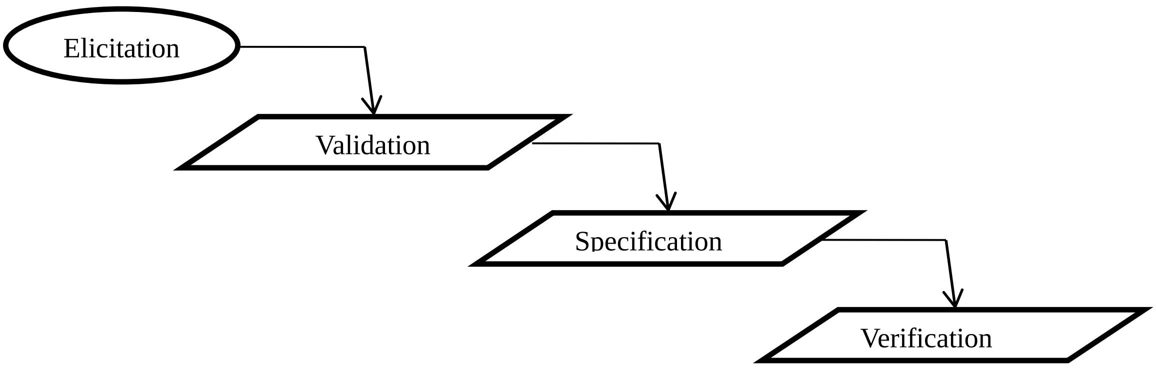

Figure 3: Work flow model for Analysis phase

3.2 Problem Analysis

On the first stage, a research on how the current text summarization (Fuzzy Logic) works is conducted to elicitate how Fuzzy Logic can be applied in text summarization. Text summarization using Fuzzy Logic can be done by completing the following stages, preprocessing, feature extraction, fuzzy logic scoring, and sentence selection. The work flow diagram in figure 4 illustrates how fuzzy logic works on text summarization.

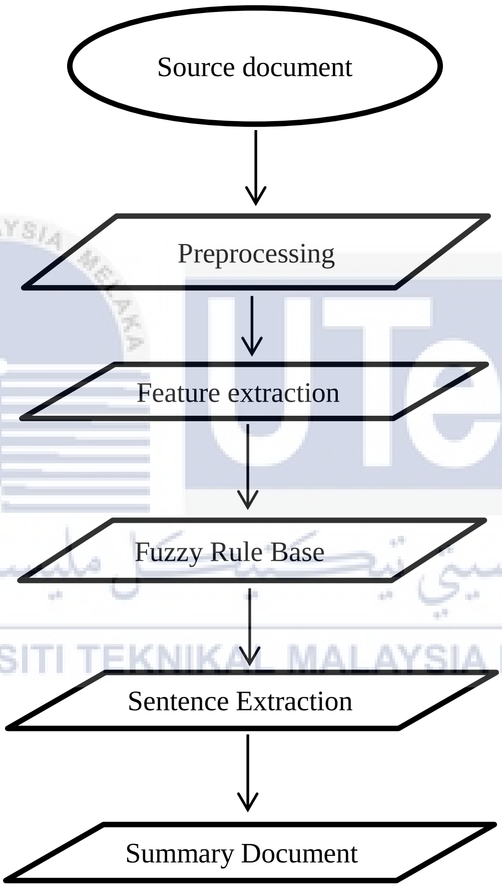

Figure 4: Work flow diagram of text summarization using Fuzzy Logic

The second stage is to validate the model. A few performance measures can be calculated to perform validation. The true positive (TP), true negative (TN), false positive (FP), and false negative (FN) have to be determined first before validation process. The output prediction by the ANFIS is classified as TP if the output prediction and the desired output are both positive. TN is for output prediction and desired output that are both negative. For output prediction that predicts a positive value when the desired output is negative, the output prediction is classified as FP. As for output prediction that predicts a negative value when the desired output is positive, the output prediction is classified as FN. With these matric, performance measures which are the accuracy, recall, f-measure, and precision can be done. The mathematical formulations are stated as below:

Accuracy = $(TP + TN) \div (TP + FN + FP + TN)$

Precision = $TP \div (TP + FP)$

Recall = $TP \div (TP + FN)$

F-measure = $2 * (\frac{precision*recall}{precision+recall})$

On the third stage, the requirements of ANFIS have to be identified. The ANFIS model should work according to the following processes, which are the dataset acquisition, feature extraction, compute training dataset, compute ANFIS model, compute testing data set, ANFIS output, and generate text summary. The work flow diagram in figure 5 illustrates the process of ANFIS in text summarization.

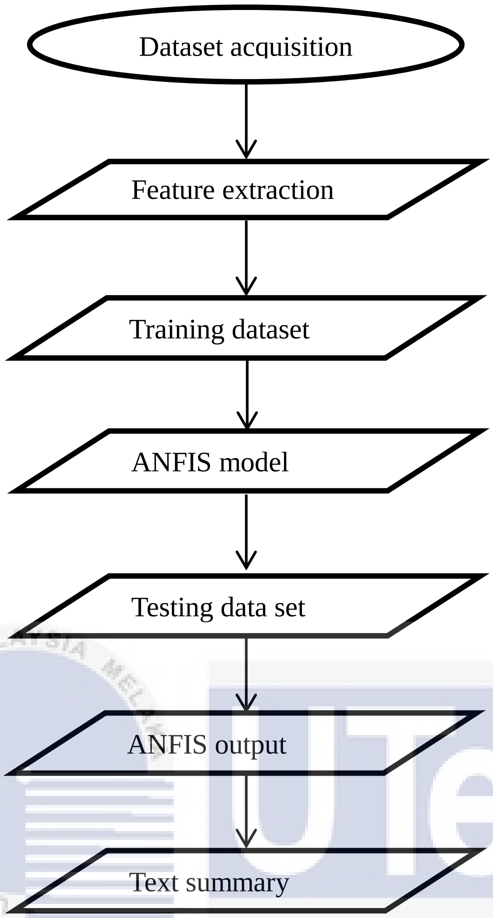

Figure 5: Work Flow Diagram of ANFIS

On the fourth stage, verification is required. To verify the new ANFIS model, the model will be tested with a multi- document file. Lastly, a text summary will be produced by ANFIS. Recall-Oriented Understudy for Gisting Evaluation (ROUGE) will be used as it has become the standards of automatic evaluation of summaries. ROUGE compares the summary generated by the model with the human- generated summaries. Calculating the root mean square error (RMSE) is also a part of validating the model trained. RMSE is used to compute the error of the desired output target with the predicted output. Lower RMSE will produce a result that is closer to the human summary.

3.3 Requirement Analysis

3.3.1 Data Requirement

3.3.1.1 DUC 2002 Dataset

The dataset which is used is the DUC 2002 Dataset. The dataset above contains 22 articles with all features extracted from the articles. The dataset above provides the documents with sentence breaks, summaries for all the documents, human summaries, and a training data. To proceed with training the data, feature extraction is required. There are 10 features provided in the DUC 2002 Dataset. In this case, 9 main features are selected as the input for ANFIS, which is the title, similarity, location, numerical data, temporal, length, proper noun, semantic term weight, and the term frequency- inverse document frequency (TF-IDF). The number of nouns and verbs is not selected as the feature for text summarization as the input does not cause much effect on the output. In fact, it could cause the predicted output to be less accurate. The dataset is trained with 75% of random training data, and 25% of random testing data. After that, an ANFIS model is produced. The article for the dataset falls under 8 categories: retailer, politics, warfare, immigration, events, disease, social problems, and natural disaster. The number of articles for each category is listed in the table 2.

Table 2: Category and Number of Books Used in the Dataset

Category	Number of documents
Retailer	2
Politics	5
Warfare	1
Immigration	1
Events	2
Disease	1
Social problems	2
Natural disaster	8

3.3.1.2 Crisp Rules Dataset

Fuzzy logic requires rules and membership functions for the system to function. Hence, a dataset for crisp rules is required to be loaded into the FIS. Each input membership function has 2 categories, which is either low (L) or high (H). There are nine features, which is F1, F2, F3, F4, F5, F6, F7, F8 and F9 to be computed and 1 output, which is O11 to be predicted. An example of the rule works as follows:

IF

(F1 is L) and (F2 is L) and (F3 is M) and (F4 is H) and (F5 is L) and (F6 is L) and (F7 is H) and (F8 is L) and (F9 is L)

THEN

(O11 is L).

Table 3 shows the rules dataset created by experts.

Table 3: Crisp rules dataset

Rule	F4	F5	F6	F7	F8	F9		F11
1	L	L	L	L	L	L		VL
2	L	L	L	L	L	H		VL
3	L	L	L	L	H	H		L
4	L	L	L	H	H	H		L
5	L	L	H	H	H	H		M
6	L	H	H	H	H	H		M
7	H	H	H	H	H	H		H
8	H	H	H	H	H	H		H
9	H	H	H	H	H	H		VH
10	H	H	H	H	H	H		VH
11	L	L	L	L	H	L		VL
12	L	L	L	H	L	L		VL
13	L	L	H	L	L	L		VL
14	L	H	L	L	L	L		VL
15	H	L	L	L	L	L		VL
16	L	L	L	L	L	L		VL
17	L	L	L	L	L	L		VL
18	L	L	L	L	L	L		VL
19	L	L	L	H	H	L		L
20	L	L	H	H	L	L		L
21	L	H	H	L	L	L		L
22	H	H	L	L	L	L		L
23	H	L	L	L	L	L		L
24	L	L	L	L	L	L		L
25	L	L	L	L	L	L		L
26	L	H	H	H	H	L		M
27	L	L	H	H	H	L		L

To avoid overfitting, the dataset has to be balanced with a significant amount of outputs for each membership functions. The output membership function is classified into 5 categories, which is very low (VL), low (L), medium (M), high (H), and very high (VH). Table 4 shows the numbers of output for each membership functions.

Table 4: Number of Outputs for each Membership Function

Membership functions	Number of outputs
VL	10
L	15
M	12
H	23
VH	10

3.3.1.3 Training and Testing Dataset

The training dataset comes from the DUC Dataset where the dataset is split into 75% as training dataset and 25% as testing dataset. To avoid overfitting from occurring, these two datasets are divided into two partially balanced amount of class 1 and class 0. Table 5 and table 6 show the number of data for both training and testing dataset.

Table 5: Number of training data

Class	1	0
Number of data	163	114

Table 6: Number of testing data

Class	1	0
Number of data	42	50

3.3.2 Functional Requirement

The function of the system is to create a summary of 200 words by using ANFIS to predict the summary sentence. At first, the dataset is loaded into MATLAB. The number of data is recorded so that permutation can be performed to shuffle the data. The user can enter the influence radius of the clustering method and the maximum number of epochs. By default, the value for influence radius is 0.55 and 40 for maximum number of epochs. After that, ANFIS will begin to train the data. The trained FIS is named as 'Outputs'. With the trained FIS, a new text document will be inserted into ANFIS to predict the summary sentence. The inputs for each feature are stated in table 7.

Table 7: Input Features Extracted From the Dataset

Input	Features
F1	Title
F2	PAS to PAS similarity
F3	Location
F4	Proper Noun
F5	Numerical Data
F6	Temporal
F7	Length
F8	TF-IDF
F9	Semantic term weight

1. Input F1 is to determine how similar the sentence towards the title. High score will be given if the sentence has much similarity with the title.
2. Input F2 is to compute the similarity between each sentence. High score will be given if the sentence has much similarity between each sentence.

3. Input F3 is to rank the sentence based on the location in the paragraph. High score will be given if the sentence is located at the top of the paragraph.
4. Input F4 is to calculate the number of proper nouns in the sentence. The sentence with the maximum number of proper noun is said to be an important sentence.
5. Input F5 is to rank the sentence based on the numerical data in the sentence. High score will be given to the sentence if there is significant numerical data in the sentence.
6. Input F6 is to rank the sentence according to the temporal value of the sentence. The sentence with a temporal value (eg. year 2014) will be given a high score.
7. Input F7 is to filter out short sentences as short sentences does not contain important idea of the document. The score is given by the ratio of length of the sentence over the length of the longest sentence in the document.
8. Input F8 is to determine the commonness of a term. The score is given by dividing the value by the maximum thematic score in a collection of thematic scores.
9. Input F9 is to calculate the importance of a sentence. The occurrence of an important term is summed up and is divided with the maximum summation values of all sentences in a document.

3.3.3 Non-functional Requirement

The performance of the system will be measured by using the precision, recall, f-measure, and accuracy. The root mean square error (RMSE) is computed before evaluating on its performance measures. RMSE is to determine the difference of desired output and the predicted output. If the RMSE of both training and testing data is low, the performance measures should show a better result. The accuracy is the overall effectiveness of the output predictor. Precision is mainly how many positive values are recorded true by the output predictor. Recall is the effectiveness of the FIS to identify positive labels. F-measure is to show the relations between the data's positive labels and those given by the output predictor.

3.3.4 Other Requirements

Other software requirements used is the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) to compare the summary generated by the model with the human- generated summaries. ROUGE is used as it has become the standards of automatic evaluation of summaries.

3.4 Conclusion

Four steps for Analysis Phase are introduced to ensure the system can accomplish all functional requirements as stated above, which is the elicitation, validation, specification, and verification. Non-functional requirements are important for analyzing the system so that the system can be used with less error produced. ROUGE is selected as the parameters for evaluating the system as it is the standards for evaluating automatic evaluation of summaries.

CHAPTER 4

THE PROPOSED TECHNIQUE

4.1 Introduction

This chapter indicates the result of the analysis of the preliminary design and the result of the detailed design. The preliminary design includes all experimental designs such as including all data and features into the system. The detailed design is the final design that has balanced data without irrelevant features. This chapter will compare both preliminary and final result to investigate the difference in both designs.

4.2 High-Level Design

4.2.1 System Architecture

This system architecture provides an overview of this text summarization technique. Two different architectural views, which is the use-case view is used to show different aspects of the system. The use-case diagram in Figure 5 is used to describe the set of scenarios which is used to represent the functionality of the system.

4.2.1.1 ANFIS

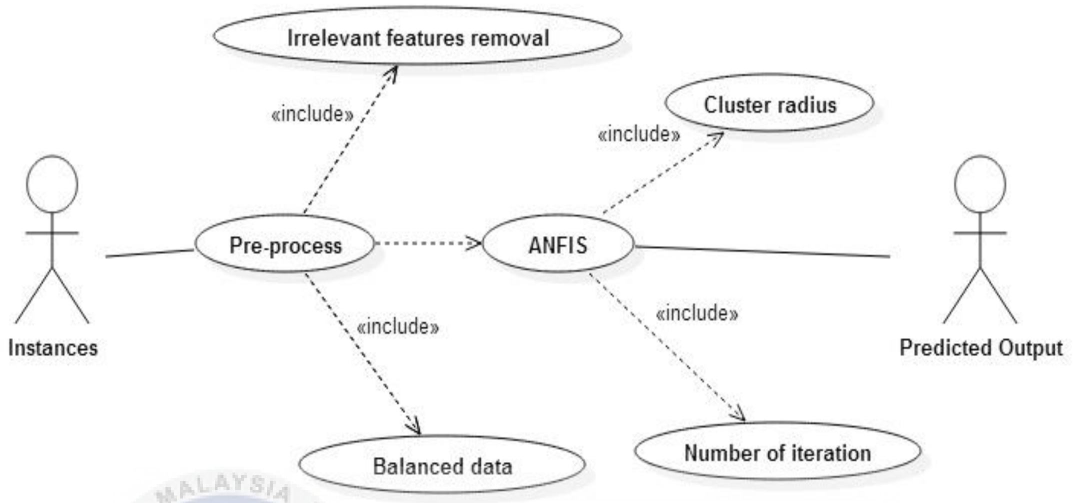

Figure 5: Use Case Diagram for ANFIS

The data instances will be pre-processed to obtain instances without irrelevant features. Balanced data can be obtained by balancing the data with the exact amount of 0 and 1 output data. The data can now be processed by ANFIS. ANFIS will enquire the user to enter the cluster radius and number of iteration input. With the inputs given by the user, ANFIS will produce the predicted output.

4.2.1.2 Neural Network

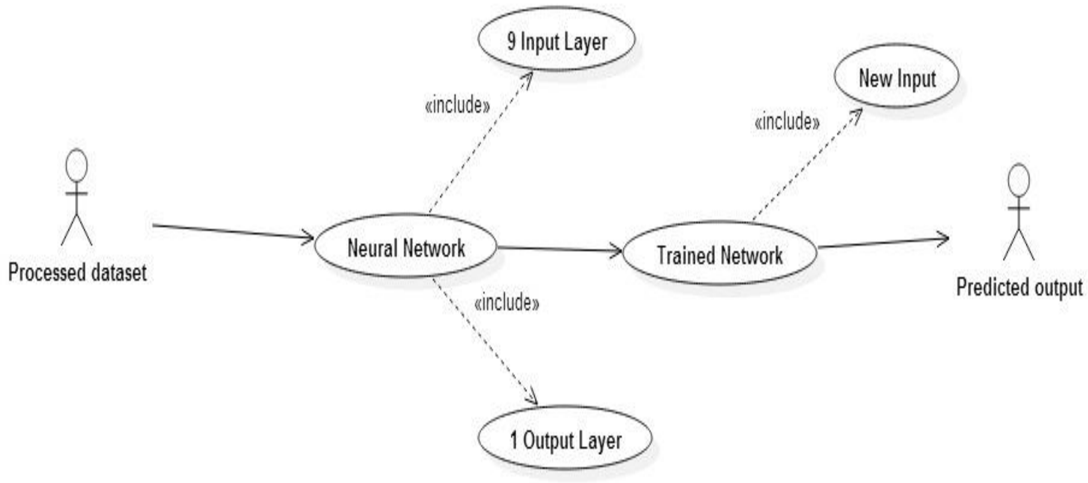

Figure 6: Use Case Diagram for Neural Network

The pre-processed dataset which is used on ANFIS will be imported into the Neural Network system. The neural network will have 9 input layers, 4 hidden layers and 1 output layer. Once the network is trained, a new input will be supplied to the network to predict the summary sentence as shown in figure 6.

4.2.2 User Interface Design

Three buttons will be used for user to select which text summarizer summaries they want to views. After selecting the text summarizer, a new window will be opened. The user can now select which dataset from the dropdown box to be viewed.

4.2.3 Technical Design

4.2.3.1 ANFIS

ANFIS is used to generate the text summary. Sugeno type ANFIS is selected as Sugeno system uses a mathematical function to define the rule based system. Sugeno systems are more compact and computationally efficient if compared to Mamdani system as Sugeno system can compute large dimension of input whereas Mamdani system can only compute a fixed number of input dimensions given by the membership function.

Clustering function produces accurate output values by using a large number of membership functions. This requires more simulation time. The idea of fuzzy clustering is to divide the data space into fuzzy clusters, each representing one specific part of the system behaviour. After projecting the clusters onto the input space, the antecedent parts of the fuzzy rules can be found. The number of rules is decided by an expert in a fuzzy inference system, which is the limitation for using fuzzy reasoning in text summarization. However in ANFIS model, no expert are required to manipulate the number of rules as the number of rules can be generated automatically by using subtractive clustering method.

Subtractive clustering algorithm estimates the cluster number and cluster location automatically. Each instance is seen as a potential cluster centre and the instances that have a value that is in the range of the first cluster will be included as the first cluster. Else, the instance will form a new cluster. The process will repeat until all instances are included in the clusters. Figure below shows the process of subtractive clustering method.The purpose of using ANFIS in text summarization is to reduce the usage of human experts in creating the crisp rules for fuzzification by using the back- propagation technique to identify the membership functions for each input. Figure 7 shows an example of membership function designed by ANFIS.

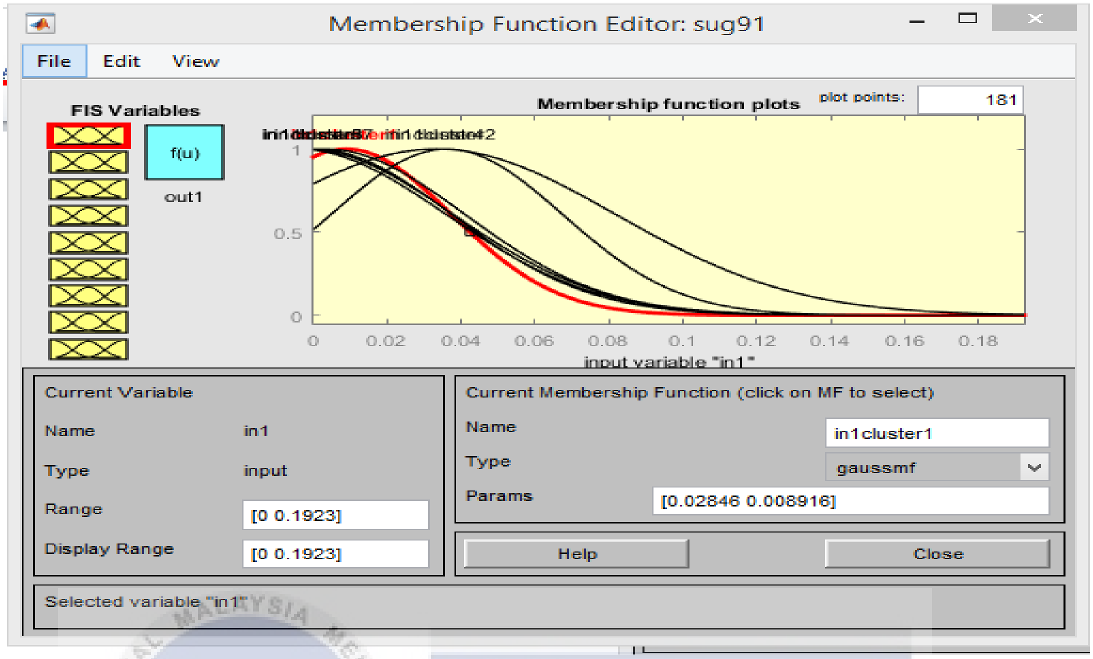

Figure 7: Membership function designed by ANFIS

The training options have been selected as follows: 40 epochs is set to stop the training process, and the cluster radius is set to be 0.55. These options are the default options for training ANFIS, so no modifications are required.

4.2.3.2 Neural Network

Neural network is developed from the principle of the human neural system. The network consists of 9 input layers, 4 hidden layers, and 1 output layer. The input layers are determined by the number of inputs that is used in the network. 4 hidden layers are selected by experimental findings where 4 hidden layers provide the highest accuracy among other numbers of hidden layers. The output layer is chosen as 1 because there is only 1 output for this project (0 or 1). Levenberg-Marquardt (LM) back propagation technique will be used as it is the default setting for training a neural network. LM is proven to be one of the fastest and efficient algorithms for training small and medium sized feed-forward neural network patterns where LM can train any network with all the weights, inputs, and transfer function is derived properly. LM also suits well with Mean Squared Error (MSE) performance index.

4.2.3.3 Fuzzy Logic

To apply fuzzy logic into text summarization, the fuzzy rules and membership function have to be set first. The fuzzy rules for the features are set with two parameters, which are Low (L) and High (H). This is to reduce the rule size as the rule size is directly proportional to the number of inputs, following the formula below:

$$R = M^I$$

where R= number of rules

M= number of membership function

I = number of inputs

Hence, following the formula above, the number of rules will be large if 3 membership functions are used for the input data. For instance, reducing the number of membership function would still lead to large number of rules generated. As such, human experts are required to determine the number of rules for this system. Figure 8 shows an example of membership function designed by the FIS.

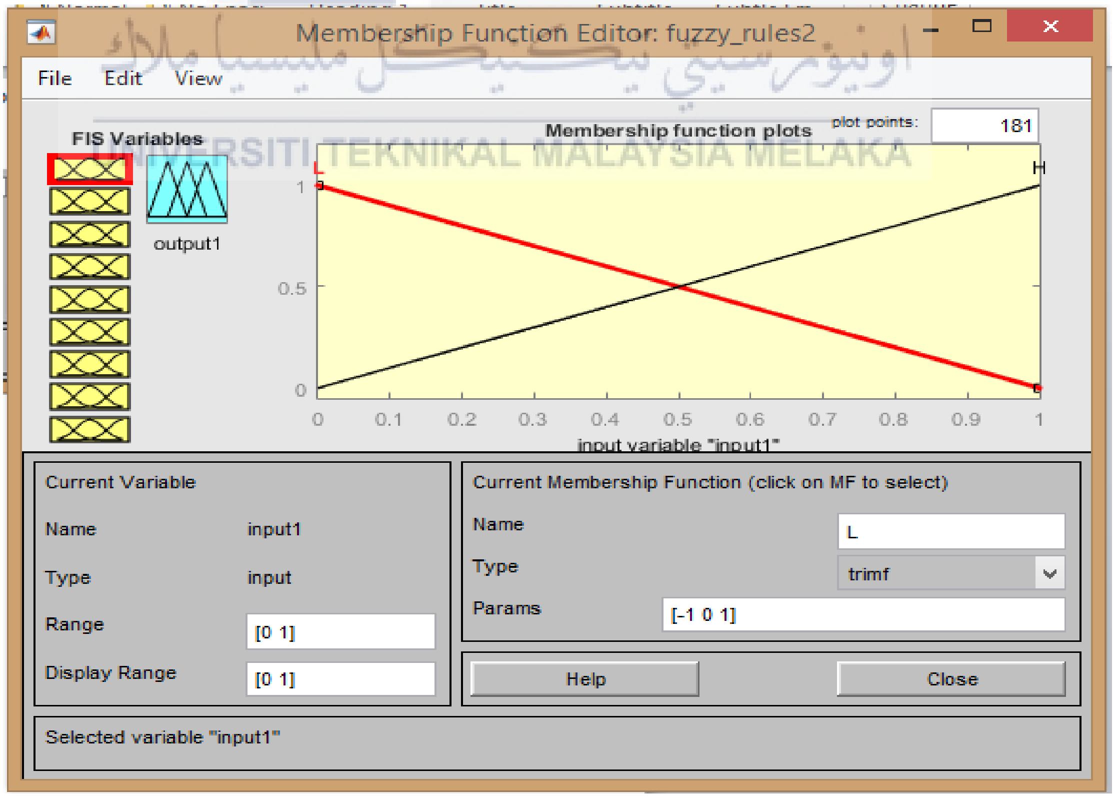

Figure 8: Membership function designed by FIS

4.2.4 Learning Algorithm

4.2.4.1 ANFIS

4.2.4.1.1 Backpropagation method

Backpropagation method originates from multilayer feedforward neural networks where the network is computed by using the gradient descent method to minimize the sum of squared errors. Backpropagation works by each input weights having their own learning rate, where the learning rate will change over time for each iteration. Backpropagation algorithm is used to find the values of the squared errors and the negative gradient for a given configuration.

4.2.4.1.2 Hybrid method

The hybrid method provided by MATLAB is a combination of least-squares and backpropagation gradient descent method. Least-squares Estimate (LSE) is used to minimize the squared error of the actual output and the target output. Then the backpropagation method is combined with the LSE to update the inputs of the membership functions. A forward pass and a backward pass is included in the hybrid optimization method where forward pass is used to calculate the error measure. The error rates are propagated from the output end towards the input end in the backward pass, where all parameters are updated. The combination of fuzzy inference to represent knowledge in the form of fuzzy rules and membership functions with the learning ability of neural network enables the membership functions parameters to be adjusted directly from the output data.

4.2.4.2 Neural Network

Neural network uses back propagation method for training the data. Back propagation is discussed earlier at the subtopic on ANFIS in 4.2.4.1.1.

4.2.4.3 Fuzzy Logic

Fuzzy Logic does not have any training algorithm since Fuzzy Logic functions with fuzzy rules and membership functions. The fuzzy rules and membership functions is discussed in 4.2.3.3.

4.2.5 Output Design

4.2.4.1 ANFIS

To achieve the optimum classification output, some testing of classification conditions has to be done. The testing conditions include:

1. If ANFIS output is lesser than 0.4, ANFIS will classify it as 1.
2. If ANFIS output is lesser than 0.5, ANFIS will classify it as 1.
3. If ANFIS output is lesser than 0.6, ANFIS will classify it as 1.
4. If ANFIS output is lesser than 0.7, ANFIS will classify it as 1.
5. If ANFIS output is lesser than 0.8, ANFIS will classify it as 1.
6. If ANFIS output is lesser than 0.9, ANFIS will classify it as 1.

The output for all test conditions is included in appendix. The result shows that ANFIS can get the optimum result with the fourth condition where the root mean square error (RMSE) is the least. Hence, the fourth condition will be used to set the classification rule for ANFIS to classify the output into crisp value (1 or 0).

4.2.4.2 Neural network

The network is trained using Levenberg-Marquardt technique whereby this technique is used to solve non-linear least squares problem. In this project, the error or the accuracy of a model can be calculated by using the least squares analysis. Hence, Levenberg-Marquardt technique is selected as the training technique for this project. The network is also trained with the same parameters which are selected by ANFIS, where the data is divided randomly and the training ratio is fixed at 0.75. Once the network is trained, the performance of the network has to be tested. The training performance can be tested by using the mean squared error to calculate the error between the actual output and the target output. The performance of the network is also determined by a test set, where a set of similar data unused during training is selected as the testing set.

4.2.4.3 Fuzzy Logic

The FIS is designed using Sugeno Inference System, where the fuzzification and the fuzzy inference process is the same as Mamdani method. The main difference between Sugeno and Mamdani is that Sugeno computes the output membership function with mathematical operations, which will provide an output either linear or constant. An example of Sugeno Inference System computes an output is given below:

IF

(input1 = a) and (input2 = b) and (input3 = c) and (input4 = d) and (input 5= e) and (input6 = f) and (input7 = g) and (input8 = h) and (input9 = i)

THEN

(output = zj + bk + cl + dm + en + fo + gp + hq + ir + c)

For a zero-order Sugeno model, each rule weights its output level by the firing strength of the rule. The final output of the system will be the weighted average of all rule outputs. A Sugeno rule operator is shown on Figure 9.

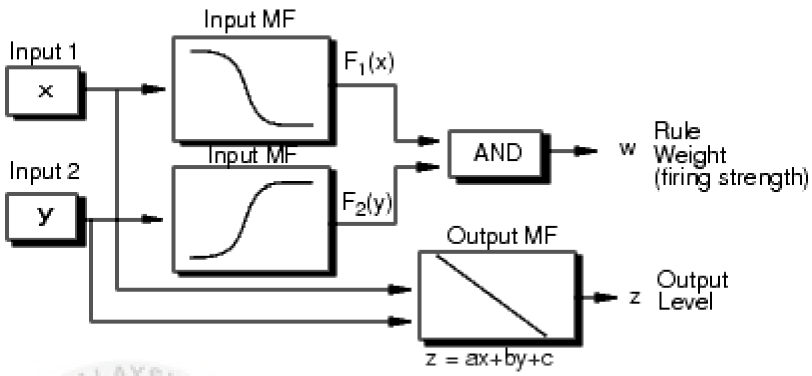

Figure 9: Sugeno rule operator

4.3 Detailed Design

4.3.1 Software Design

Firstly, the system should load the input and output of the data into the MATLAB workspace. The system will compute the total number of inputs and outputs to divide them into training and testing data. After that, 2 dialog boxes will appear for user to input the influence radius and the number of epochs. ANFIS will begin to train the data loaded into the workspace. After training of the data is complete, ANFIS will begin to predict the test output using the model trained. With the predicted output supplied by ANFIS, ANFIS will sort the data in descending order where the data with higher score will be included as a summary sentence. The system will stop including data into summary sentence when the length of the summary reached 200 words and 400 words. To defuzzify the predicted output into crisp value, 6 experiments are done to obtain the optimum classification condition. The predicted output will be classify according to the condition if the predicted output is greater than 0.7, the predicted output will be classified as a summary sentence (1) and vice versa. With the defuzzified output, the performance measure can be computed. The performance measures that will be included are the accuracy, precision, recall, and f-measure. At last, 3 graphs will be plotted, which is the training data against the number of instances graph, testing data against the number of instances graph and the overall data against the number of instances will be included for analyzing the system. A regression curve will be used to show how much the predicted output varies from the target output. Figure 10 shows the activity diagram of the system.

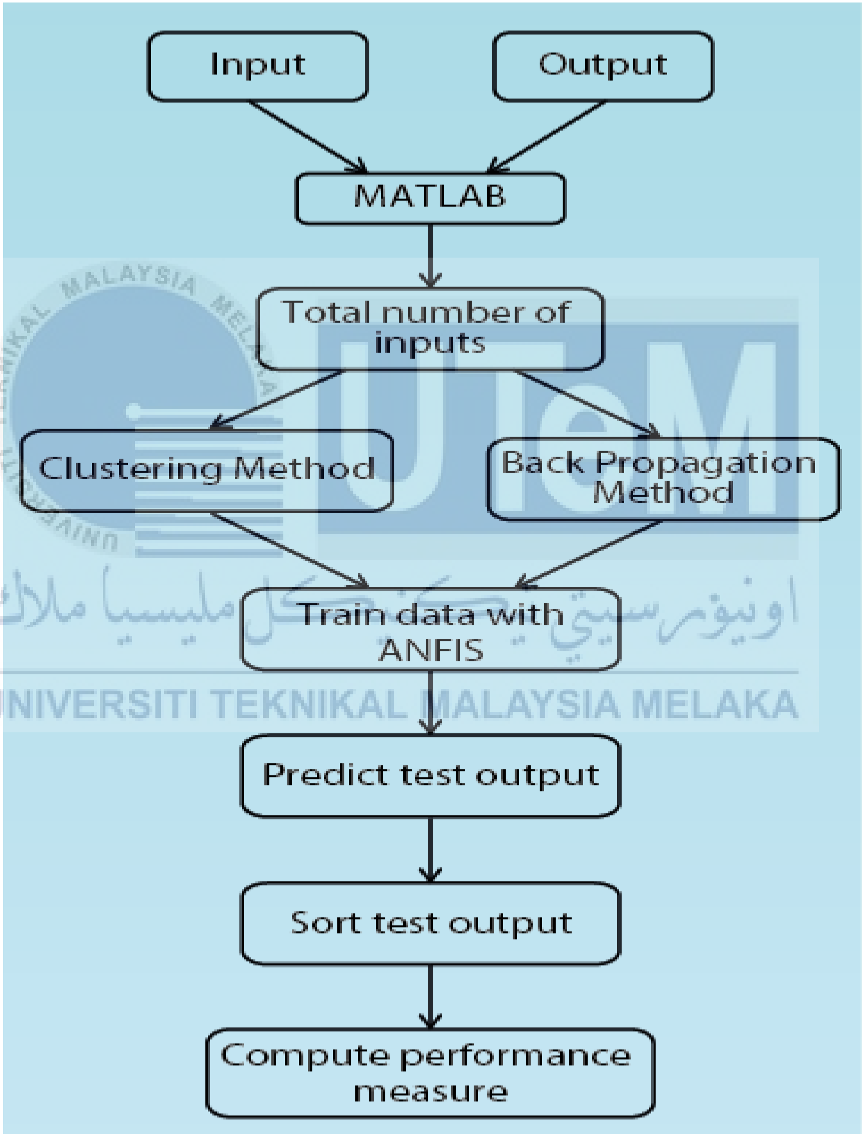

Figure 10: Activity diagram of the system

4.4 Conclusion

This chapter provides an overview of the system, using 2 different architecture views to present different aspects of the system. The Use-Case View is important for selection of different set of scenarios to represent the functionality of each object in the Use-Case Diagram. The Deployment View is used to describe the physical nodes for each of the platform configurations.

Many experiments have been done in order to achieve the optimum output design. With the result from these experiments, ANFIS can classify the predicted output with lesser error. Other experiments such as balancing the dataset and removing irrelevant features are also important for ANFIS to classify the output with least error.

For the User Interface Design, the AI technique and methods for text summarization are introduced. Sugeno type ANFIS with clustering method will be designed and different datasets will be provided to train the system by using MATLAB. By using clustering method, the membership degree of a data point will be directly computed in this projected cluster according to its distance from the projected cluster center.

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

Many commitments have been made for text summarization purpose. This project is to implement a new method in producing summary sentence, which is ANFIS. To implement this system, research based on previous studies is done to provide more details on how ANFIS works. The research shows that ANFIS is suitable for text summarization. Hence, ANFIS will be implemented as the classification model for this project.

5.2 Software Development Environment Setup

This system will be developed using MATLAB R2014b as there are many libraries that can be accessed through this software. A deployment view as shown in Figure 11 is used to describe the mapping of the software and shows the systems distributed aspects.

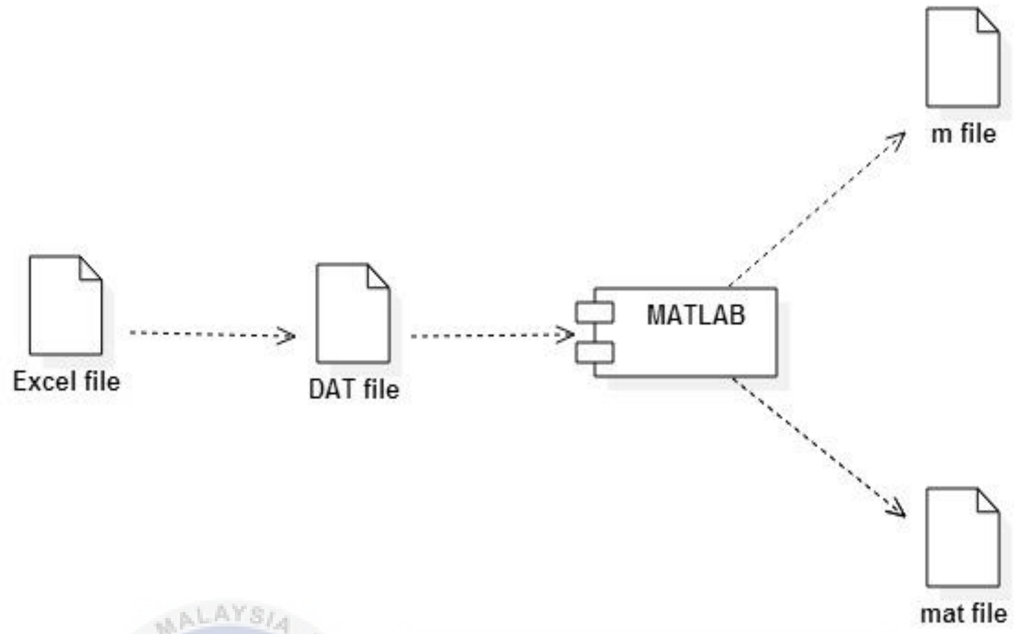

Figure 11: Deployment View Diagram

The excel file (.xls) contains the documents with all the instances and input features. This file will be converted into text file (.dat) as MATLAB 32bit version could not read excel file. 2 new text file will be created to store the input features and target output. These 2 files will be loaded into MATLAB for the system to process the summary sentence. ".m file" is used to save the source code and ".mat" file is used to store the variables derived in the workspace. The program will not run if either 1 from these 2 files are missing.

5.3 Software Configuration Management

5.3.1 Configuration environment setup

Software configuration management is important for creating design and system testing of the project. Software configuration is a control tool to be developed as the project progresses. Each time the process is changed has to be recorded and reported to maintain a status record of all items in the system. This will provide traceability of all changes to the system.

5.3.2 Version Control Procedure

Version control is important for a system to be applied systematically. Version control enables the tracking of how the system is developed during drafting process. The common version control system will be used in this project where the version control simply gives a number to each version of the document. The versions for this project are as follow:

'Proposal Version 0.1.docx'	Draft Versions
'Proposal Version 0.2.doc'	Finalized proposal that will be handed to the committee.
'FYP Version 1.0.doc'	At this stage the document has been reviewed by supervisor.
'FYP Version 1.1.doc'	At this stage any changes towards the system are recorded.
'FYP Version 1.2.doc'	Changes are made due to the result of the system.
'FYP Version 2.0.doc'	A major revamp is done when the document is reviewed by supervisor.

5.4 Implementation Status

Implementation begins when a solution is approved. The planning, executing, and deploying changes are included into implementation progress. The planning phase is to plan how to set the conditions for the system to classify the predicted output. For the execution phase, all the conditions are tested to achieve the best result out of all the conditions. Changes are made if the conditions did not affect the result of the predicted output. Figure 12 presents an activity flowchart for the Implementation phase.

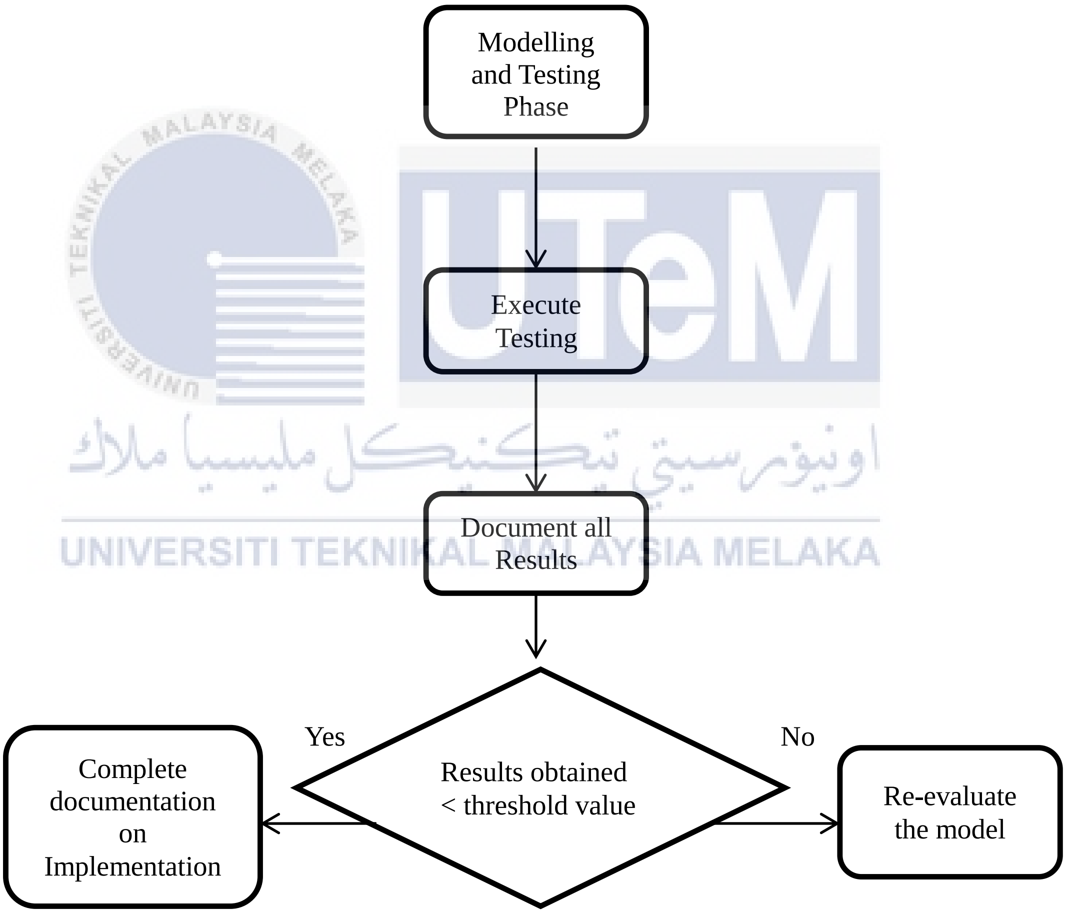

Figure 12: Modelling and Testing Phase Activities

5.5 Conclusion

The implementation of a new method for text summarization requires many testing to be done. Hence, the activity diagram for the implementation phase is important to record all testing that have been done. Version control is required for keep tracking how the system is started from an initial idea. Version control is also served as an evidence for all progress that have been done throughout the project.

CHAPTER 6

TESTING

6.1 Introduction

Testing is a process to satisfy the project requirements. The code is executed to check that it meets the requirements of the project. Testing is also used to demonstrate that the project is fit for purpose where this project could help the users to carry out text summarization. Testing is required to detect faulty codes where the codes will cause failure when it is operational. There are two types of testing which is the static and dynamic testing. Static test is done by finding defects without executing the code where else dynamic test is done by finding defects by executing the code. In this project, dynamic test is used often to obtain an optimal result.

6.2 Test plan

6.2.1 Test Environment

The system will be implemented and tested using MATLAB R2015a as it is a standard workspace for research analysis. The test is done by using a new multi-document and applies it on both techniques. The sample feature of the dataset is shown in Figure 13. The rows represent the sentences and columns represents the feature score for each sentence.

title	PAS to PAS similarity	Location	proper noun	numerical data	temporal	length
0.125	0.79282	1	0.18182	0	0.04545	0.66667
0	0.92476	0.9375	0	0.18182	0	0.33333
0	0.92267	0.875	0.23529	0	0.11765	0.51515
0.06667	0.92762	0.8125	0.3	0	0	0.30303
0	0.81421	0.75	0.17647	0.17647	0	0.51515
0.05882	0.80483	0.6875	0.16667	0	0.16667	0.36364
0.03125	0.80016	0.625	0.21429	0.14286	0.07143	0.84848
0.03448	0.83587	0.5625	0.33333	0.04167	0	0.72727
0	0.83918	0.5	0.25	0	0.125	0.48485
0.04545	0.84072	0.4375	0.15789	0.05263	0	0.57576
0	0.92698	0.375	0	0	0	0.09091
0	0.79595	0.3125	0.14286	0	0.14286	0.42424
0.10526	0.80707	0.25	0.26667	0	0.06667	0.45455
0	0.91817	0.1875	0	0.08333	0	0.36364
0	0.81981	0.125	0.15385	0	0	0.39394
0	0.91021	0.0625	0.08333	0	0.08333	0.36364
0.07692	0.75759	1	0.3	0.05	0	0.60606
0	0.84406	0.96875	0	0	0	0.12121
0	0.79411	0.9375	0	0	0	0.09091
0	0.79716	0.90625	0.4375	0	0	0.48485
0.0625	1	0.875	0	0.11111	0	0.27273
0	0.87919	0.84375	0	0	0	0.63636
0.03846	0.70773	0.8125	0.25	0	0.05	0.60606
0.03571	0.76915	0.78125	0.28571	0.04762	0	0.63636
0	0.73947	0.75	0.25	0	0	0.60606
0	0.77523	0.71875	0.15385	0.03846	0	0.78788
0	0.76694	0.6875	0.5	0	0	0.24242

Figure 13: Input dataset that is loaded into MATLAB

6.2.2 Test schedule

For all the 5 conditions listed below, the hidden layer number with the highest accuracy is selected as the fittest hidden layer number. For instance, the hidden layer number of 5 shows the highest accuracy. The testing continues by mutating the hidden layer number 5 to 4 and 6. The hidden layer with the highest accuracy, for instance, 4 is selected as the fittest among all. The testing continues with hidden layer number of 1, 2, 3 and 7, 8, 9 but still hidden layer number of 4 is tested to be the hidden layer with the highest accuracy.

6.3 Test Strategy

6.3.1 Dynamic strategy

There are few test strategy that are widely used, namely analytical, model-based, methodical, dynamic and many more. Dynamic strategy is selected as the test strategy for this project as dynamic strategy focus on finding as many defects as possible during text execution. This strategy allows the location of defects to be identified before the program is coded completely. This strategy also allows the programmer to identify the defects before moving on to another function of the program.

6.3.2 White box technique

White box technique is implemented in this project as this technique focus on the internal structure of the program. The tester knows what the code looks like and how to execute the testing methods with certain parameters. The program is run with predetermined inputs to check if there are any defaults produced. There are several testing that can be done using white box technique, namely variable coverage and condition coverage.

6.3.2.1 Variable coverage

Variable coverage is a measure for all variables that is used did not overlap with another variable. In this project, there are many variables that have their own unique values. Hence if the value of the variable is assigned wrongly, the result will be faulty. Figure 14 shows the variables in the project with unique names and values.

Name	Value
dataFromExcel	121x4 double
dataInput	369x9 double
dataInput067	122x9 double
dataInput0671	9x122 double
dataOutput	369x1 double
dataOutput067	122x1 double
DefaultValues	1x1 cell
DisplayError	1
DisplayFinalResult	1
DisplayInfo	1
DisplayOptions	1x4 logical
DisplayStepSize	1
errors	1x369 double
ExcelRow2	121x1 double
FN	3
FP	4
h	121x1 cell
h1	121x1 double
hiddenLayerSize	4
in_fis	1x1 struct
index	20x1 double
MaxEpoch	40
n	50
nData	369
net	1x1 network
nInd067	121

Figure 14 : Variable names and values.

6.3.2.2 Condition coverage

Condition coverage measures the outcome of the sub expression written in the code. For example, the expression {if (a+ b==3) return true} should return the Boolean value of 1 if the summation of a and b is 3. This coverage is to test all the conditions in the program to avoid faulty conditions. Figure 15 shows an example of condition coverage in the program.

```
rrTestInd=size(r1TestOutputs,1);
for tData = drange(1:tData)
if tTestOutputs(tData,:) > 0.7
    rrrrTestOutputs(tData,:) =1 ;
else
    rrrrTestOutputs(tData,:)=0 ;
end

end
```

Figure 15: An example of condition coverage in the program

6.4 Test Implementation

6.4.1 Test Description

The test activities are controlled and recorded to compare which control are the best for this project. The conditions for classification are tested accordingly and the results with the lowest error will be selected as the threshold for the classification conditions as stated in Chapter 4. The selection for the number of hidden layer is tested to achieve the highest possible accuracy. The selection is tested by reduction method, where:

1. the first hidden layer number is 100
2. the second random guess is 50
3. the third random number is 25
4. the fourth random hidden layer number is 10
5. the fifth random hidden layer number is 5

6.4.2 Test Data

There are several data that is required for performance measure, which is the number of True Positive, False Positive, True Negative, and False Negative. These data are collected and will be shown in the confusion matrix as shown in Table 8.

Table 8: Confusion Matrix of ANFIS

		Predicted Class 1	Predicted Class 0
Actual Class	1	20	22
	0	8	42

The confusion matrix above shows the number of actual class for class 1, which is 20(TP) +22(FN) = 42 and the number of actual class for class 0, which is 8(FP) +42(TN) = 50. ANFIS has predicted a total of 28 for class 1(20+8), and a total of 64 for class 0(22+42).

Table 9: Confusion Matrix of Neural Network

		Predicted Class 1	Predicted Class 0
Actual Class	1	30	12
	0	25	25

The confusion matrix shown in table 9 is the number of actual class for class 1, which is 30(TP) +12(FN) = 42 and the number of actual class for class 0, which is 25(FP) +25(TN) = 50. Neural Network has predicted a total of 55 for class 1(30+25), and a total of 37 for class 0(12+25).

Table 10: Confusion Matrix of Fuzzy Logic

		Predicted Class 1	Predicted Class 0
Actual Class	1	5	37
	0	0	50

The confusion matrix above shown in Table 10 is the number of actual class for class 1, which is 5(TP) + 37(FN) = 42 and the number of actual class for class 0, which is 0(FP) + 50(TN) = 50. Fuzzy Logic has computed a total of 5 for class 1(5+0), and a total of 87 for class 0(37+50).

The true positive (TP), true negative (TN), false positive (FP), and false negative (FN) have to be determined first. The output prediction by the ANFIS is classified as TP if the output prediction and the desired output are both positive. TN is for output prediction and desired output that are both negative. For output prediction that predicts a positive value when the desired output is negative, the output prediction is classified as FP. As for output prediction that predicts a negative value when the desired output is positive, the output prediction is classified as FN. With these matric, performance measures which are the accuracy, recall, f-measure, and precision can be done. The mathematical formulations are stated as below:

Accuracy = $(TP + TN) \div (TP + FN + FP + TN)$

Precision = $TP \div (TP + FP)$

Recall = $TP \div (TP + FN)$

F-measure = $2 * (\frac{precision * recall}{precision + recall})$

6.5 Test Results and Analysis

6.5.1 ANFIS

The test data for ANFIS, which is the performance measure with the 6 conditions of classification are collected and is shown in the Table 11.

Table 11: Data collection on classification with 6 conditions

Condition	Accuracy	Average precision	Average Recall
TestOutput > 0.4	0.2131	0.5556	0.5637
TestOutput > 0.5	0.2623	0.5588	0.5909
TestOutput > 0.6	0.2951	0.5612	0.6091
TestOutput > 0.7	0.4098	0.5715	0.6728
TestOutput > 0.8	0.5328	0.5461	0.6296
TestOutput > 0.9	0.6230	0.5391	0.6053

With the test data above, it is clearly shown that the classification condition of rounding up the test output when the test output is greater than 0.7 compute the best performance. This is due to the amount of class 0 is more than class 1. Hence, a higher threshold value will show a result with least error. However, the threshold of 0.8 shows a decrease in the precision and recall if compared to threshold of 0.7 as there are instances which is supposedly to be labelled as class 1 are incorrectly labelled as class 0.

6.5.2 Neural Network

Table 12: Performance Measure for Each Number of Hidden Neurons Used

Number of hidden layers	Accuracy	Average Precision	Average Recall
100	0.2893	0.5612	0.6055
50	0.3306	0.5645	0.6285
20	0.2 397	0.5577	0.578
10	0.2645	0.5328	0.5547
5	0.1983	0.5157	0.518

The accuracy of the test outputs with different number of hidden layers are shown in Table 12. Although the number of hidden layers with 50 provides a high accuracy, it is not selected as the optimal condition for this project as it exceeds the summation of the neurons in the input and output layer. The number of hidden nodes should not be twice the number of input layer. Hence, the neurons number of 10 in the hidden layers is selected as the optimal condition for this project as it also can provide a high accuracy.

6.5.3 Hold out function

The hold out function in MATLAB is used to permutate the dataset into equally amount of classes inside the dataset. This function enables the permutation of different training and testing data using different fold of data in each iteration. Hence, multiple results and performance measure can be obtained from different testing data created using hold out function.

6.6 Results and Discussion

Table 13: Performance Measure for class 1 without hold out

Text Summarizer	Precision	Recall
ANFIS	0.7143	0.4762
Neural Network	0.5455	0.7143
Fuzzy Logic	1.0	0.1190

Table 13 shows the performance measure for class 1 without hold out function. The results show that ANFIS provides the best performance among the 3 models. However, Fuzzy Logic shows a higher precision when compared with ANFIS.

Table 14: Performance Measure for class 0 without hold out

Text Summarizer	Precision	Recall
ANFIS	0.6563	0.8400
Neural Network	0.6757	0.5
Fuzzy Logic	0.5747	1.0

Table 13 and Table 14 show the comparison for the performance measure for all the 3 text summarizers. ANFIS provides a better results if compared to Fuzzy Logic and Neural Network. The results show that the precision and recall is high in ANFIS when compared with Neural Network and Fuzzy Logic.

Table 15: Performance Measure for class 1 with hold out

Text Summarizer	Precision	Recall
ANFIS	I11: 0.8286	I11: 0.5370
	I12: 0.6923	I12: 0.5000
	I13: 0.8000	I13: 0.6667
	I14: 0.6923	I14: 0.5000
	I15: 0.7381	I15: 0.5741
Neural Network	I11: 0.6545	I11: 0.7500

	I12: 0.6226	I12: 0.6875
	I13: 0.6154	I13: 0.5000
	I14: 0.5846	I14: 0.7917
	I15: 0.6250	I15: 0.5208
Fuzzy Logic	I11: 1.0000	I11: 0.0185
	I12: 1.0000	I12: 0.0370
	I13: 1.0000	I13: 0.0000
	I14: 0.0000	I14: 0.0370
	I15: 1.0000	I15: 0.0556

Table 15 shows the performance measure for class 1 with hold out. The result shows ANFIS shows the best result for class 1 with hold out function. However, the precision for Fuzzy Logic is the highest when compared with ANFIS.

Table 16: Performance Measure for class 0 with hold out

Text Summarizer	Precision	Recall
ANFIS	I11: 0.5556	I11: 0.4412
	I12: 0.6087	I12: 0.7778
	I13: 0.7143	I13: 0.8333
	I14: 0.6087	I14: 0.7778

	I15: 0.6515	I15: 0.7963
Neural Network	I11: 0.5556	I11: 0.4412
	I12: 0.4828	I12: 0.4118
	I13: 0.4419	I13: 0.5588
	I14: 0.4118	I14: 0.2059
	I15: 0.4524	I15: 0.5588
Fuzzy Logic	I11: 0.5047	I11: 1.0000
	I12: 0.5094	I12: 1.0000
	I13: 0.5000	I13: 1.0000
	I14: 0.5094	I14: 1.0000
	I15: 0.5143	I15: 1.0000

Table 16 shows the performance measure for class 0 with hold out. The result shows ANFIS shows the best result for class 0 with hold out function.

Table 17: Comparison between Performance Measure of ANFIS, Neural Network, and Fuzzy Logic with hold out

Model	Accuracy	F-Measure	Precision	Recall
ANFIS	I11: 0.7130	I11: 0.7277	0.7431	0.7130
	I12: 0.6389	I12: 0.6446	0.6505	0.6389
	I13: 0.7500	I13: 0.7536	0.7571	0.7500
	I14: 0.6389	I14: 0.7110	0.7184	0.7037
	I15: 0.6852	I15: 0.6900	0.6948	0.6852
Neural Network	I11: 0.6220	I11: 0.6003	0.6051	0.5956
	I12: 0.5732	I12: 0.5512	0.5527	0.5496
	I13: 0.5244	I13: 0.5290	0.4419	0.5294
	I14: 0.5488	I14: 0.4985	0.4982	0.4988
	I15: 0.5366	I15: 0.5393	0.5381	0.5398
Fuzzy Logic	I11: 0.5093	I11: 0.6074	0.7523	0.5093
	I12: 0.5185	I12: 0.6147	0.7547	0.5185
	I13: 0.5000	I13: 0.3333	0.2500	0.5000
	I14: 0.5185	I14: 0.6147	0.7547	0.5185
	I15: 0.5278	I15: 0.6220	0.7571	0.5278

Table 17 shows the average performance measure of the 3 models with hold out. The result shows ANFIS shows the best average result with hold out function.

Table 18: Comparison between Performance Measure of ANFIS, Neural Network, and Fuzzy Logic without hold out

Text Summarizer	Accuracy	F-Measure	Average Precision	Average Recall
ANFIS	0.6739	0.6714	0.6853	0.6581
Neural Network	0.5978	0.6088	0.6106	0.6072
Fuzzy Logic	0.5978	0.6542	0.7874	0.5595

Table 18 shows the comparison between performance measure of ANFIS, Neural Network and Fuzzy Logic. The results show that ANFIS can perform the best within the 3 models above. However, the precision of Fuzzy Logic is higher when compared with ANFIS.

Table 19: Comparison between Performance Measures of ANFIS With and Without Hold Out

Hold Out	Accuracy	F-Measure	Average Precision	Average Recall
Yes	0.6852	0.7054	0.7128	0.6982
No	0.6739	0.6714	0.6853	0.6581

Table 19 shows a higher result of the performance of ANFIS with hold out function. The precision of the result is higher when the testing set is computed with hold out function.

Table 20: Comparison between Performance Measure of ANFIS, Neural Network, and Fuzzy Logic with hold out

Text Summarizer	Accuracy	F-Measure	Average Precision	Average Recall
ANFIS	0.6852	0.7054	0.7128	0.6982
Neural Network	0.5610	0.5437	0.5272	0.5426
Fuzzy Logic	0.5148	0.5584	0.6538	0.5148

From the results above, it is clearly shown that testing with five different balanced testing data using the holdout function provides a better result and performance measure for each AI technique proposed above. Hence, training and testing data have to be balanced in order to perform with better results.

6.7 Conclusion

Testing phase is important to meet the requirements for a project. The system is tested with a multi- document which can be collected from the DUC 2002 dataset. Testing phase is required to reduce faulty errors before we move on to the next stage. In each function of the program, testing phase has to be done to ensure all variables and conditions are well defined. The testing data are collected for further performance measure to be done. Both methods are tested and the results show that Fuzzy Logic can produce a higher reasoning capacity when compared with Neural Network method, while ANFIS can provide the best result based on performance measure.

CHAPTER 7

ROUGE EVALUATION

7.1 Introduction

ROUGE (Recall-Oriented Understudy for Gisting Evalaution) is a measurement to determine the quality of a summary generated automatically by machines. ROUGE compares the summary generated with summaries created by human experts. An ideal automated summary is measured by counting the number of overlapping units such as word sequences and word pairs between the automated summary and the human summary. 3 experiments have been conducted to identify the factors that will affect the quality of the summary. 4 ROUGE measures are introduced, namely ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU* in this project.

7.1.1 ROUGE-L: Longest Common Subsequence (LCS)

LCS is used to identify cognate candidates during construction of the best translation lexicon from parallel text. The length of LCS of two words and the length of the longer word of the two words are used to measure the cognateness between them. The summary sentence is viewed as a sequence of words when applying LCS in summarization evaluation. The longer the LCS of the two summary sentences is, the more similar the two summaries are.

7.1.2 ROUGE-W: Weighted Longest Common Subsequence

ROUGE-W is introduced as LCS has a problem in differentiating spatial relations within their embedding sequences. The length of consecutive matches is considered to improve the basic LCS method. For example:

X : [A B C D E F G]

Y1: [A B C D H I K]

Y2: [A H B K C I D]

Y1 and Y2 have the same ROUGE-L score. However, Y1 shows a better choice as Y1 has consecutive matches. In this case, evaluation using ROUGE-W will provide a higher score of Y1 than Y2.

7.1.3 ROUGE-S: Skip-Bigram Co-Occurrence Statistics

ROUGE-S allows arbitrary gaps in any pairs of words in the sentence. Skip-bigram measures the overlap between a candidate translation and a set of reference translations. For example:

S1. police killed the gunman
S2. police kill the gunman
S3. the gunman kill police
S4. the gunman police killed

Each sentence has 6 skip-bigrams. For example, the skip- bigram for S1 is (“police killed”, “police the”, “police gunman”, “killed the”, “killed gunman”, “the gunman”), while S2 has 3 skip- bigram that matches with S1, which is (“police the”, “police gunman”, “the gunman”). S3 has one skip-bigram match with S1 (“the gunman”), and S4 has two skip-bigram matches with S1 (“police killed”, “the gunman”). In this case, S2 is said to be more similar to S1 as the number of skip- bigram in S2 is the highest among the 3 sentences.

7.1.4 ROUGE-SU: Extension of ROUGE- S

The problem found in ROUGE-S is that no credit will be given to the sentence that does not have any skip- bigram pairing. For example, the sentence S5 has a ROUGE-S score of 0.

S5. gunman the killed police

S5 is the exact reverse sentence of S1 but there is no skip bigram match between them. ROUGE-S is extended with the addition of unigram as counting unit to differentiate sentences that do not have word occurrence with S1. ROUGE-SU can be obtained from ROUGE-S by adding a begin-of-sentence marker at the beginning of candidate and reference sentences.

7.2 Experiment 1

The first experiment is done by using 10 different summary sets generated by machine to evaluate the performance of the 3 models created. This experiment is done to investigate whether the number of summary sets would affect the results of the evaluation. The result of the first experiment is shown in table 21, table 22, and table 23.

Table 21: ROUGE Evaluation of Summary Generated by ANFIS

	Average- Recall	Average- Precision	F- measure
ROUGE-L	0.20530	0.23183	0.21719
ROUGE-W	0.07830	0.15173	0.10294
ROUGE-S	0.04442	0.05692	0.04945
ROUGE-SU	0.04726	0.06043	0.05256

Table 26 shows the ROUGE evaluation of summary generated by ANFIS with 10 datasets. ROUGE- L shows the highest result among the 4 ROUGE evaluations.

Table 22: ROUGE Evaluation of Summary Generated by Fuzzy Logic

	Average- Recall	Average- Precision	F- measure
ROUGE-L	0.36279	0.39777	0.37591
ROUGE-W	0.13957	0.26302	0.18049
ROUGE-S	0.12548	0.14885	0.13285
ROUGE-SU	0.12961	0.15382	0.13720

Table 22 shows the ROUGE evaluation of summary generated by Fuzzy Logic with 10 datasets. ROUGE- L shows the highest result among the 4 ROUGE evaluations.

Table 23: ROUGE Evaluation of Summary Generated by Neural Network

	Average- Recall	Average- Precision	F- measure
ROUGE-L	0.21617	0.24753	0.22872
ROUGE-W	0.07745	0.14719	0.10068
ROUGE-S	0.04854	0.06530	0.05346
ROUGE-SU	0.05150	0.06902	0.05669

Table 23 shows the ROUGE evaluation of summary generated by Neural Network with 10 datasets. ROUGE- L shows the highest result among the 4 ROUGE evaluations.

7.3 Experiment 2

The second experiment is done by using all 30 summary sets generated by machine to evaluate the performance of the 3 models created. This experiment is done to compare the results collected in experiment 1. The result of the second experiment is shown in table 24, table 25, and table 26.

Table 24: ROUGE Evaluation of Summary Generated by ANFIS

	Average- Recall	Average- Precision	F- measure
ROUGE-L	0.25988	0.27790	0.26751
ROUGE-W	0.10152	0.18602	0.13073
ROUGE-S	0.06947	0.08225	0.07417
ROUGE-SU	0.07303	0.08613	0.07785

Table 24 shows the ROUGE evaluation of summary generated by ANFIS with 30 datasets. ROUGE- L shows the highest result among the 4 ROUGE evaluations.

Table 25: ROUGE Evaluation of Summary Generated by Fuzzy Logic

	Average- Recall	Average- Precision	F- measure
ROUGE-L	0.27111	0.29208	0.28012
ROUGE-W	0.10594	0.19656	0.13710
ROUGE-S	0.07738	0.08867	0.08166
ROUGE-SU	0.08098	0.09273	0.08543

Table 25 shows the ROUGE evaluation of summary generated by Fuzzy Logic with 30 datasets. ROUGE- L shows the highest result among the 4 ROUGE evaluations.

Table 26: ROUGE Evaluation of Summary Generated by Neural Network

	Average- Recall	Average- Precision	F- measure
ROUGE-L	0.26187	0.27332	0.26656
ROUGE-W	0.10457	0.18795	0.13390
ROUGE-S	0.06921	0.07477	0.07099
ROUGE-SU	0.07282	0.07863	0.07468

Table 26 shows the ROUGE evaluation of summary generated by Neural Network with 30 datasets. ROUGE- L shows the highest result among the 4 ROUGE evaluations.

7.4 Experiment 3

The third experiment is done by filtering out overlapping sentences with similar ideas being discussed. This experiment is to investigate whether overlapping sentences would affect the quality of the sentences as human experts would not include sentences with similar ideas into the summary. The result of the third experiments is shown in table 27, table 28, and table 29.

Table 27: ROUGE Evaluation of Summary Generated by ANFIS with Word Overlap Reduction

	Average- Recall	Average- Precision	F- measure
ROUGE-L	0.20000	0.28244	0.23870
ROUGE-W	0.08095	0.18458	0.11157
ROUGE-S	0.04699	0.08728	0.05921
ROUGE-SU	0.05004	0.09233	0.06293

Table 27 shows the evaluation of summary generated by ANFIS with word overlap reduction. The result on ROUGE-L shows the highest result among the four ROUGE evaluations.

Table 28: ROUGE Evaluation of Summary Generated by Neural Network with Word Overlap Reduction

	Average- Recall	Average- Precision	F- measure
ROUGE-L	0.23617	0.29702	0.26143
ROUGE-W	0.09617	0.20832	0.13075
ROUGE-S	0.05691	0.08970	0.06804
ROUGE-SU	0.06023	0.09455	0.07192

Table 28 shows the evaluation of summary generated by Neural Network with word overlap reduction. The result on ROUGE-L shows the highest result among the four ROUGE evaluations.

Table 29: ROUGE Evaluation of Summary Generated by Fuzzy Logic with Word Overlap Reduction

	Average- Recall	Average- Precision	F- measure
ROUGE-L	0.23594	0.29781	0.26157
ROUGE-W	0.09343	0.20195	0.12692
ROUGE-S	0.05879	0.09208	0.07031
ROUGE-SU	0.06208	0.09693	0.07416

Table 29 shows the evaluation of summary generated by Fuzzy Logic with word overlap reduction. The result on ROUGE-L shows the highest result among the four ROUGE evaluations.

7.5 Comparison of ANFIS performance among the 3 experiments

The experiments above show a different result based on performance measure. For Experiment 1 and Experiment 2, the performance measure shows a huge improvement which proved that the number of dataset used in evaluation will affect the performance of the model. However for Experiment 3, the result shows a slight decrease when compared with Experiment 2. This shows that removing overlapping sentences would not improve the results as it may lead to important sentences being removed. Table 30, table 31, table 32 and table 33 shows the performance measure of ANFIS on all the 3 experiments.

Table 30: Evaluation of Summary Generated by ANFIS using ROUGE-L

	Average- Recall	Average- Precision	F- measure
Experiment 1	0.20530	0.23183	0.21719
Experiment 2	0.25988	0.27790	0.26751
Experiment 3	0.20992	0.28244	0.23870

Table 30 shows the evaluation results of summary generated by ANFIS using ROUGE-L. The results show that for experiment 2, the result is the highest among the 3 experiments. However, the precision for experiment 3 is the highest among the 3 experiments.

Table 31: Evaluation of Summary Generated by ANFIS using ROUGE-W

	Average- Recall	Average- Precision	F- measure
Experiment 1	0.07830	0.15173	0.10294
Experiment 2	0.10152	0.18602	0.13073
Experiment 3	0.08095	0.18458	0.11157

Table 31 shows the evaluation results of summary generated by ANFIS using ROUGE-W. The results show that for experiment 2, the result is the highest among the 3 experiments.

Table 32: Evaluation of Summary Generated by ANFIS using ROUGE-S

	Average- Recall	Average- Precision	F- measure
Experiment 1	0.04442	0.05692	0.04945
Experiment 2	0.06947	0.08225	0.07417
Experiment 3	0.04699	0.08728	0.05921

Table 32 shows the evaluation results of summary generated by ANFIS using ROUGE-S. The results show that for experiment 2, the result is the highest among the 3 experiments.

Table 33: Evaluation of Summary Generated by ANFIS using ROUGE-SU

	Average- Recall	Average- Precision	F- measure
Experiment 1	0.04726	0.06043	0.05256
Experiment 2	0.07303	0.08613	0.07785
Experiment 3	0.05004	0.09233	0.06293

Table 33 shows the evaluation results of summary generated by ANFIS using ROUGE-SU. The results show that for experiment 2, the result is the highest among the 3 experiments. However, the precision for experiment 3 is the highest among the 3 experiments.

7.6 Discussion and Conclusion

From the experiments above, the evaluation result of ANFIS is increased from experiment 1 to experiment 2. This shows that the number of dataset used for evaluation will affect the result of the experiment. However, the result of ANFIS still did not perform better than Fuzzy Logic. This could be due to the fuzzy rules created by human expert favors the parameters presented in ROUGE. Due to the time constraint, the parameters of ROUGE are not managed to be explored as there are many parameters in ROUGE required to be explored. Hence, the default parameters in ROUGE are applied which caused the result of evaluation to be more favorable towards Fuzzy Logic. The evaluation result decreased when experiment 3 is conducted. Removing word overlap sentences could remove some important sentences which will lead to low performance on evaluation.

Further study should be done on adjusting the parameters of ROUGE to help improve the result of the evaluation. Hence, the number of datasets should be increased for the evaluation of ANFIS to be improved.

CHAPTER 8

CONCLUSION

8.1 Observation on Weakness and Strengths

Researches have been done on Fuzzy Logic to perform classification task. However, the existing Fuzzy Logic text summarizer requires human experts to manually construct the fuzzy rules for the model to generate summary sentence. After all experiments have been done, ANFIS is found to provide the best classification result. However, ANFIS could not perform well on ROUGE evaluation when compared with Fuzzy Logic model. This could be due to the parameters of ROUGE did not favor the classification done by ANFIS.

8.2 Propositions for Improvement

Further research on other parameters of ROUGE to evaluate a summary should be done so that these parameters can help in improving the evaluation result. Other evaluation tools should also be used so that the evaluation results can be compared to determine which summarizer can provide the best summary.

8.3 Project Contribution

This project is completed with the contribution of Universiti Teknikal Malaysia Melaka (UTeM) under the Faculty of Information and Communication Technology for providing an adequate place to conduct this project. This project is also completed under the supervision of Dr. Yogan Jaya Kumar who contributes a lot of ideas and solutions when a problem is found. The appendices can be found after this page. Comments from panels also have contributed a lot in improving this project.

8.4 Conclusion

The objectives of this project are met where classification is performed on each of the techniques above. Summaries are generated using the techniques above which met the second objectives in this project. For the third objective of this project, ANFIS is concluded as the best binary classifier, but ANFIS shows a lower evaluation result when compared with Fuzzy Logic. As a conclusion, ANFIS shows a good potential of generating summaries. Further studies on how to improve the quality of summaries generated by ANFIS should be done so that the process of generating summaries can be fully automated by using ANFIS.

APPENDICES

Figure 1: Turnitin Report for Final Year Project 2 Report

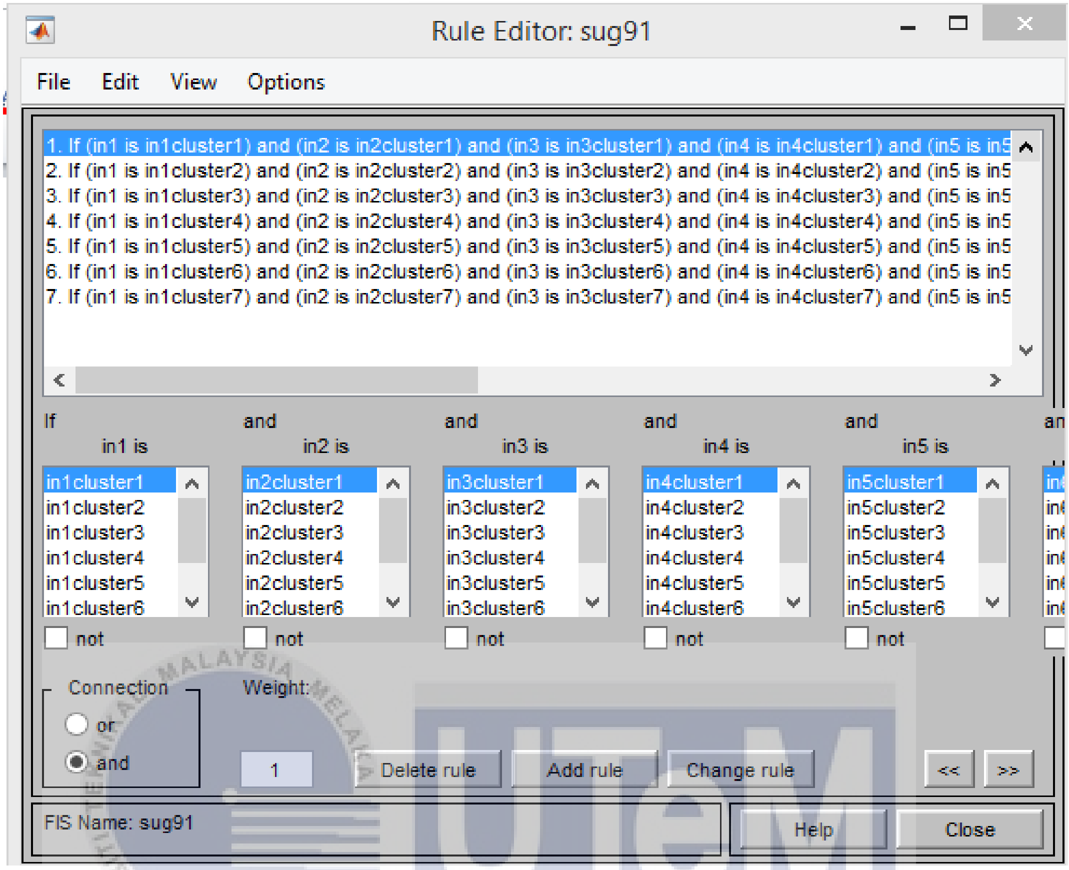

Figure 2: Rules Generated Automatically by ANFIS using Clustering Method

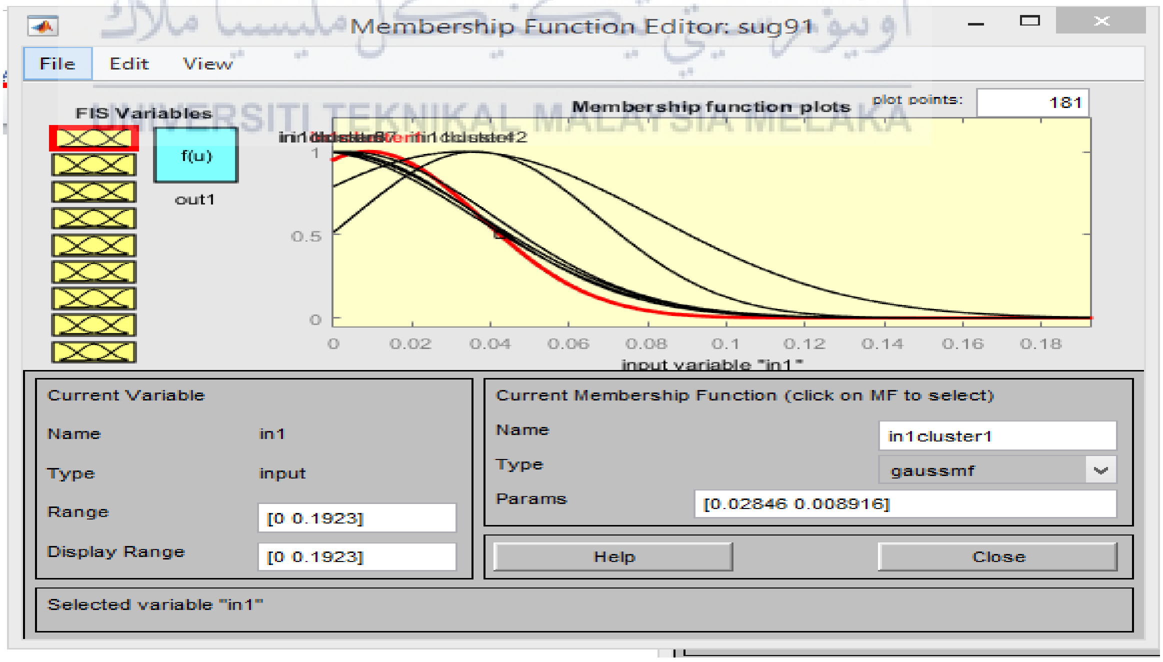

Figure 3: ANFIS Membership Functions for the First Input

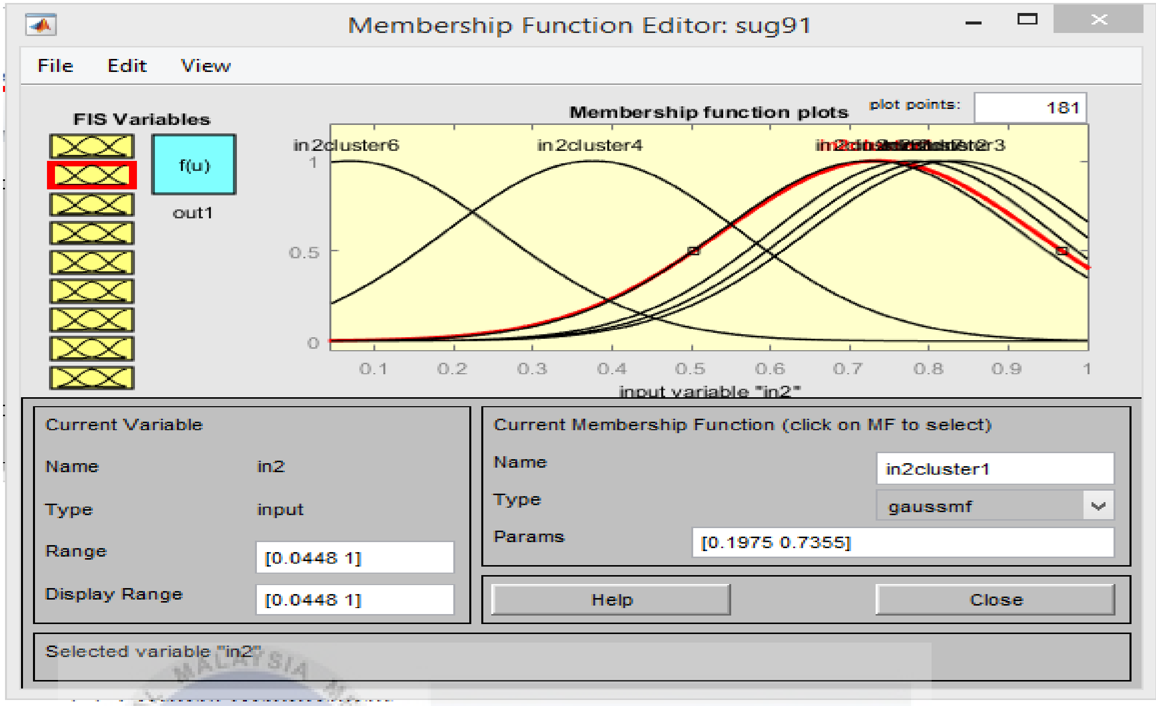

Figure 4: ANFIS Membership Functions for the Second Input

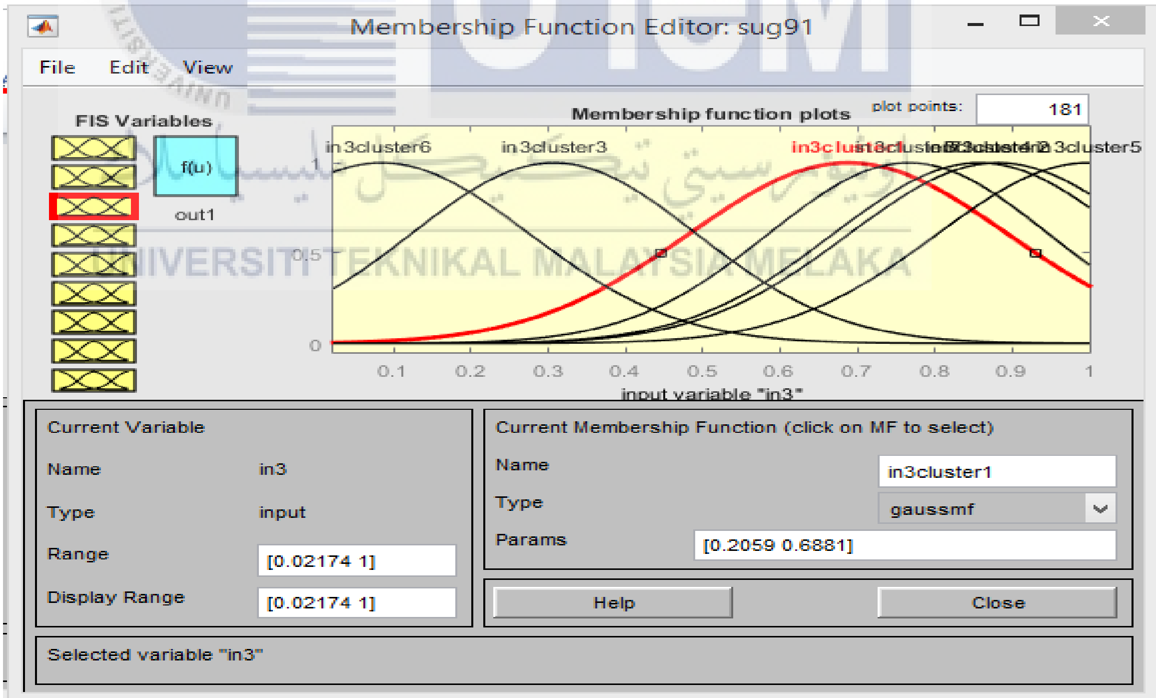

Figure 5: ANFIS Membership Functions for the Third Input

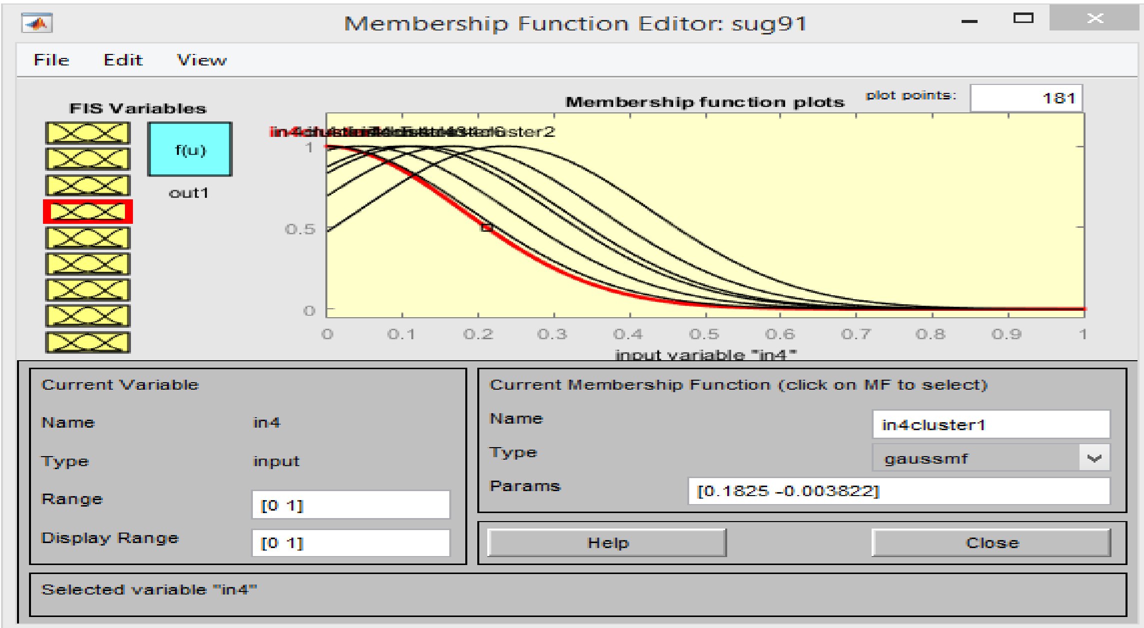

Figure 6: ANFIS Membership Functions for the Fourth Input

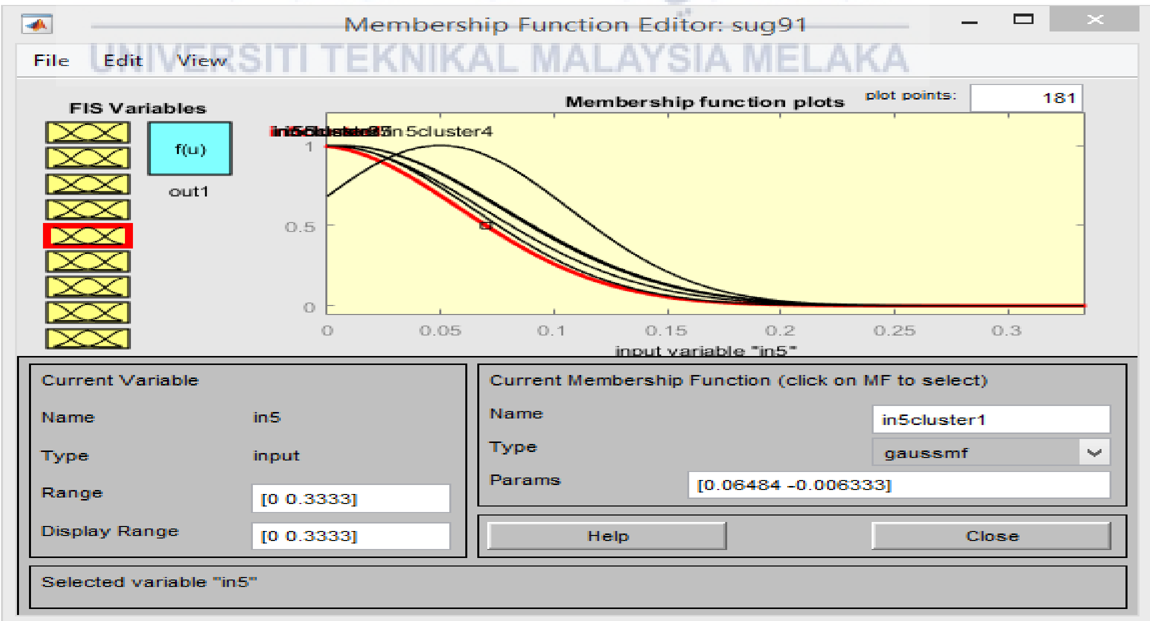

Figure 7: ANFIS Membership Functions for the Fifth Input

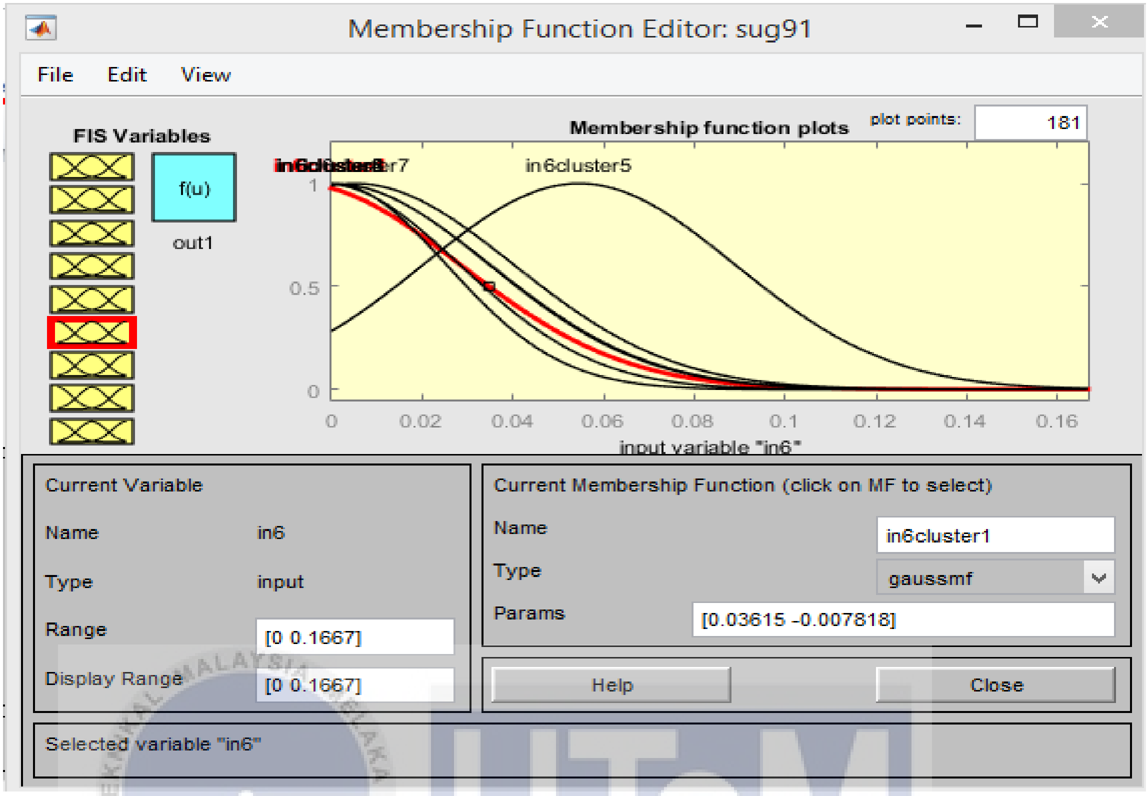

Figure 8: ANFIS Membership Functions for the Sixth Input

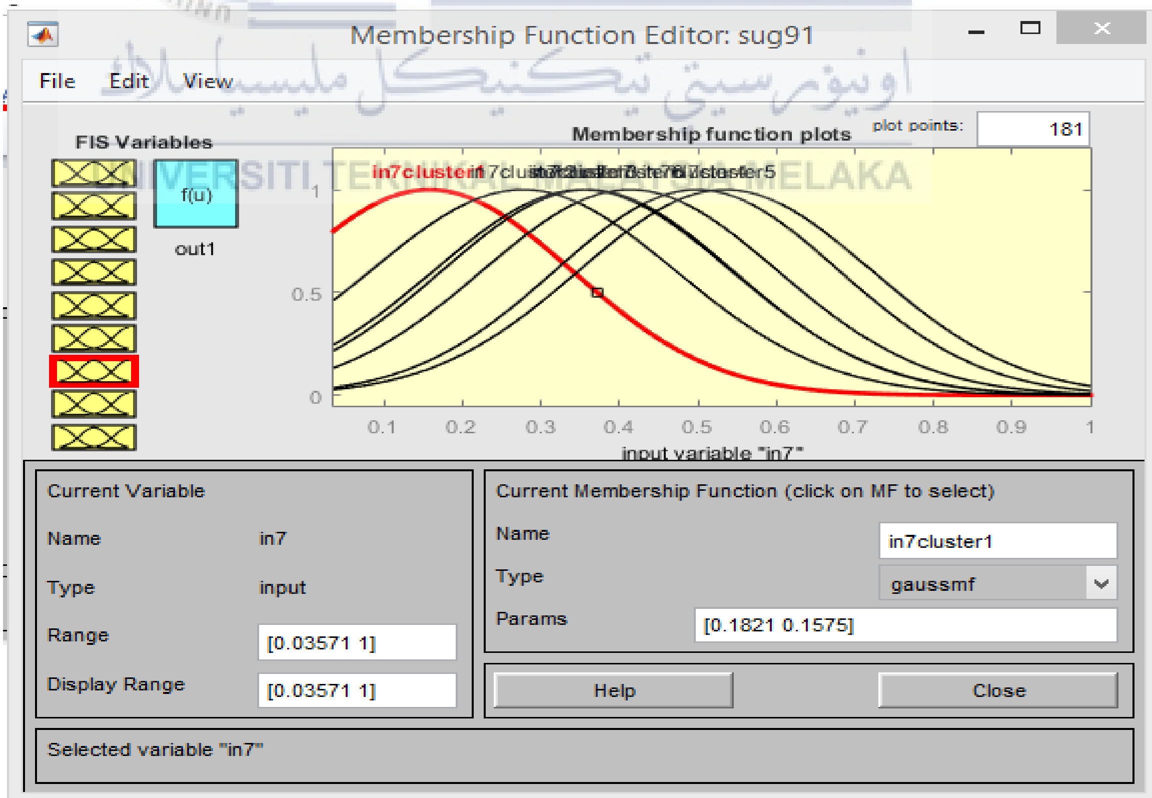

Figure 9: ANFIS Membership Functions for the Seventh Input

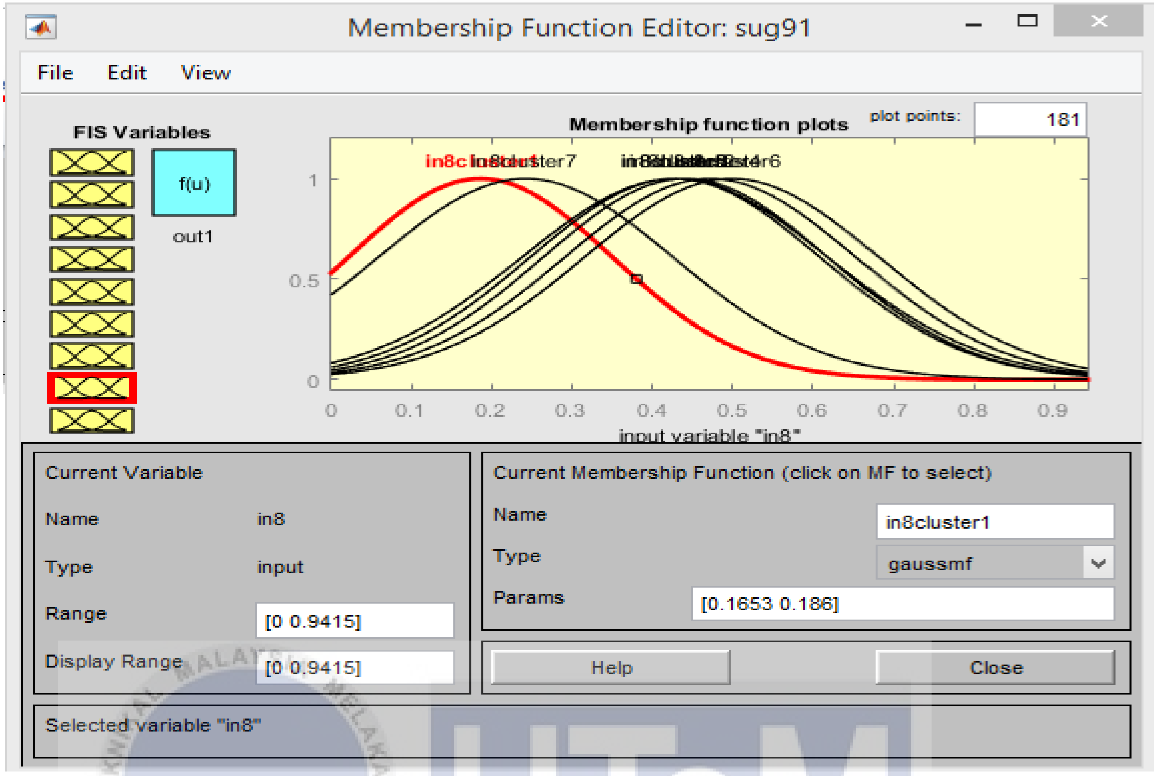

Figure 10: ANFIS Membership Functions for the Eighth Input

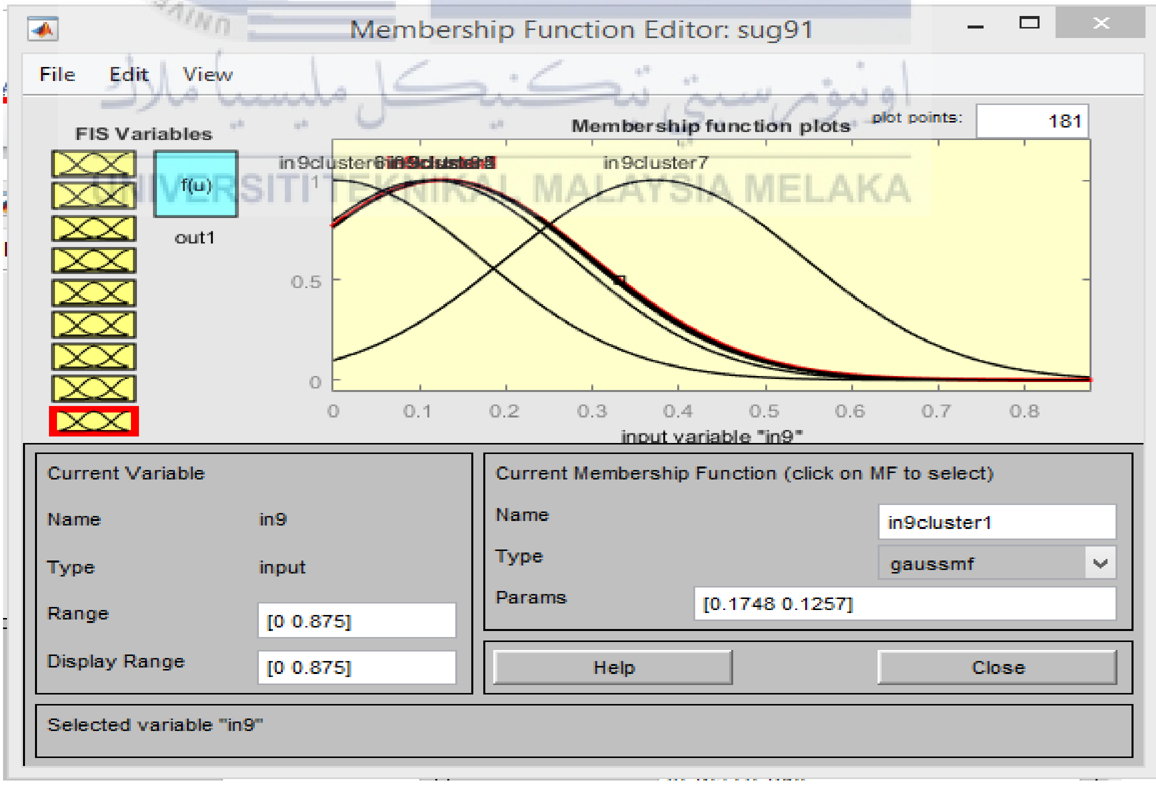

Figure 11: ANFIS Membership Functions for the Ninth Input

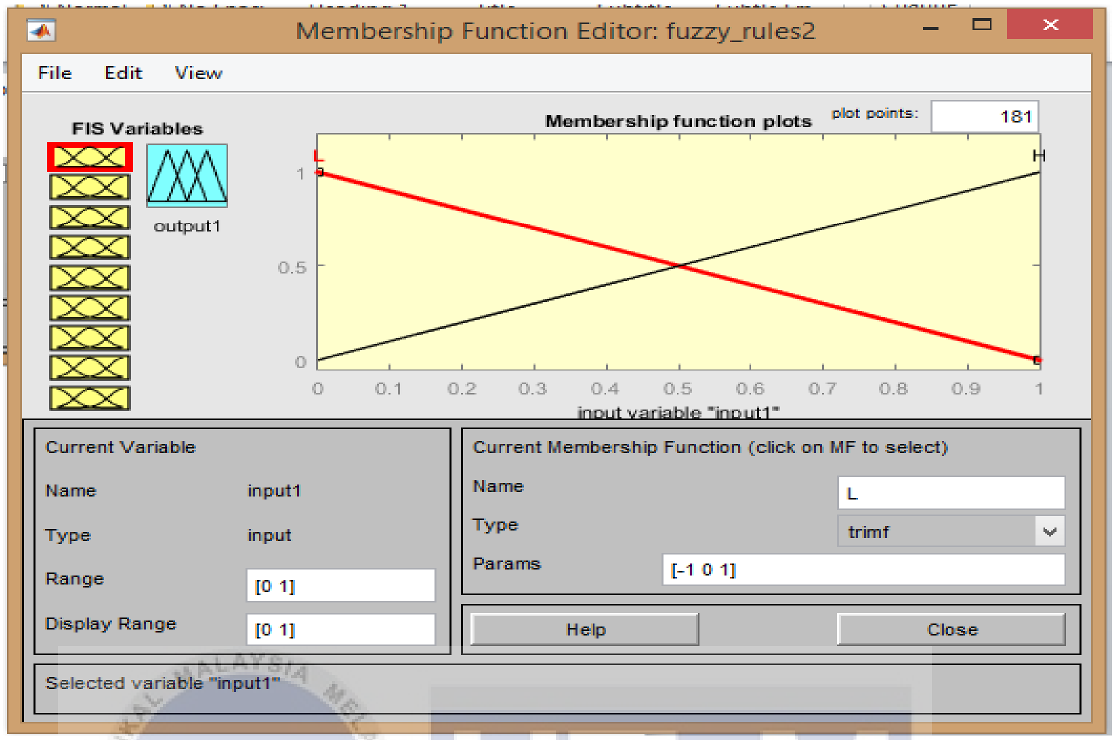

Figure 12: Fuzzy Logic Membership Function for the First Input

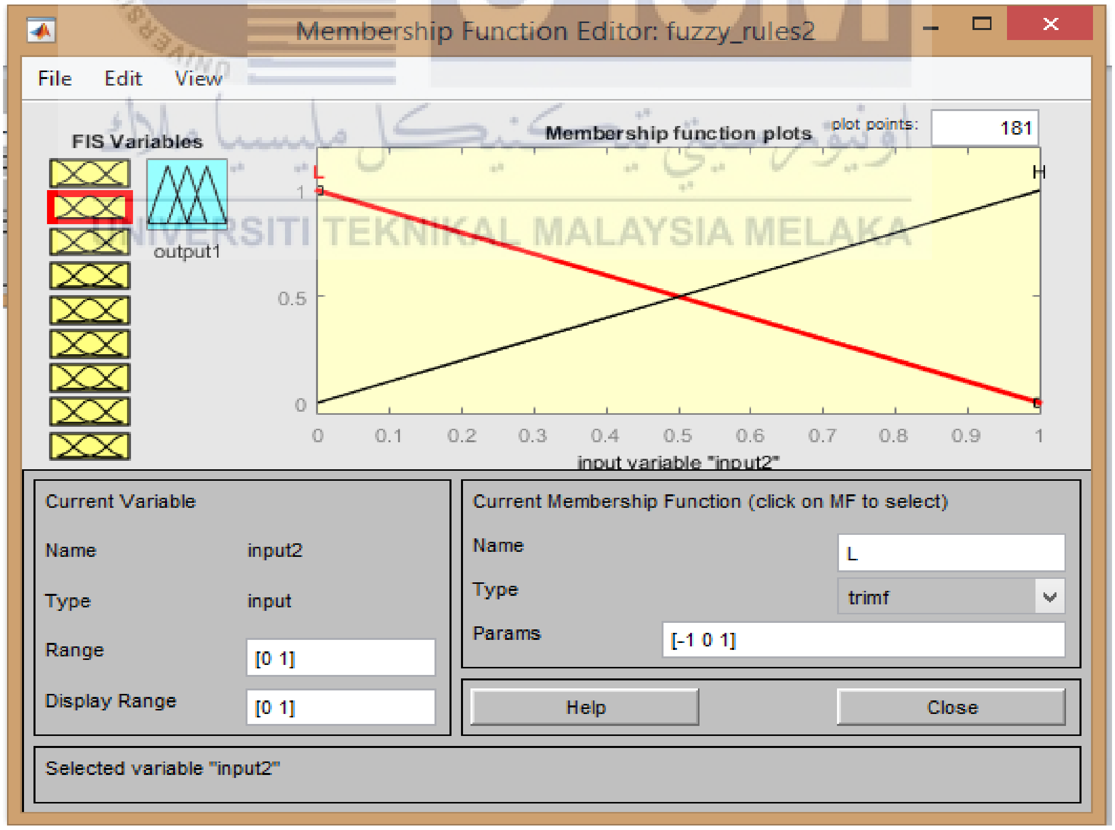

Figure 13: Fuzzy Logic Membership Function for the Second Input

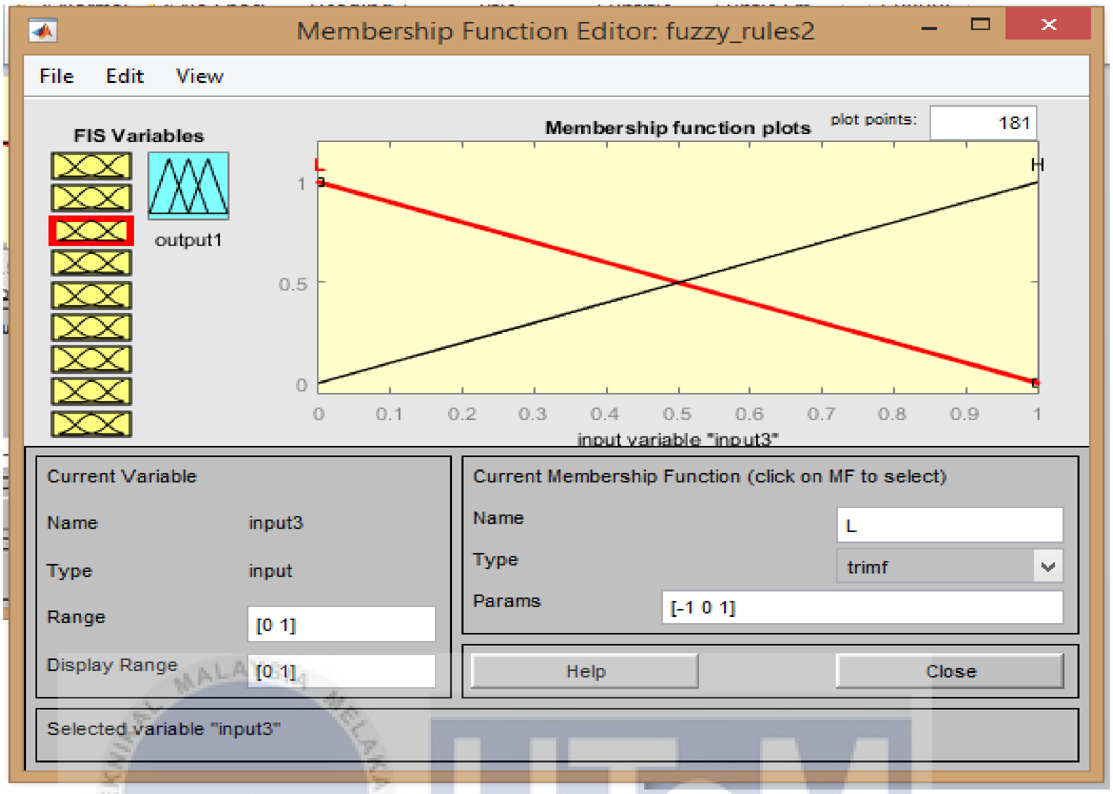

Figure 14: Fuzzy Logic Membership Function for the Third Input

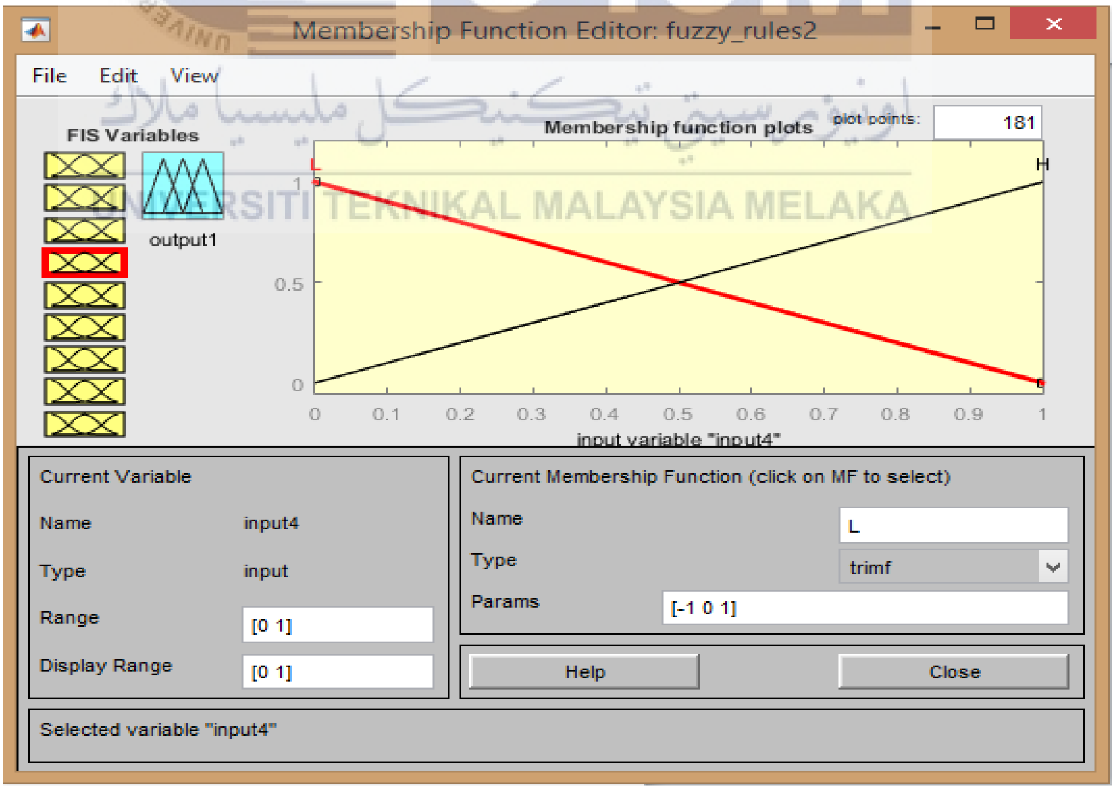

Figure 15: Fuzzy Logic Membership Function for the Fourth Input

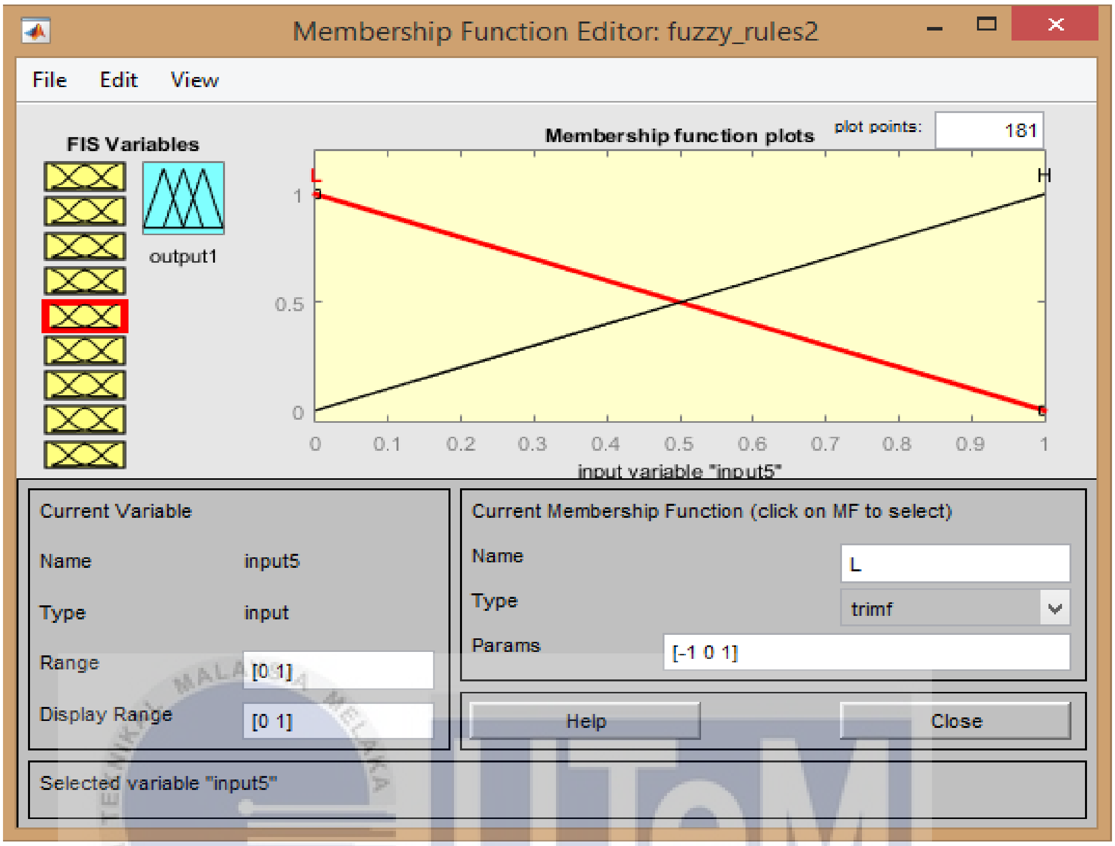

Figure 16: Fuzzy Logic Membership Function for the Fifth Input

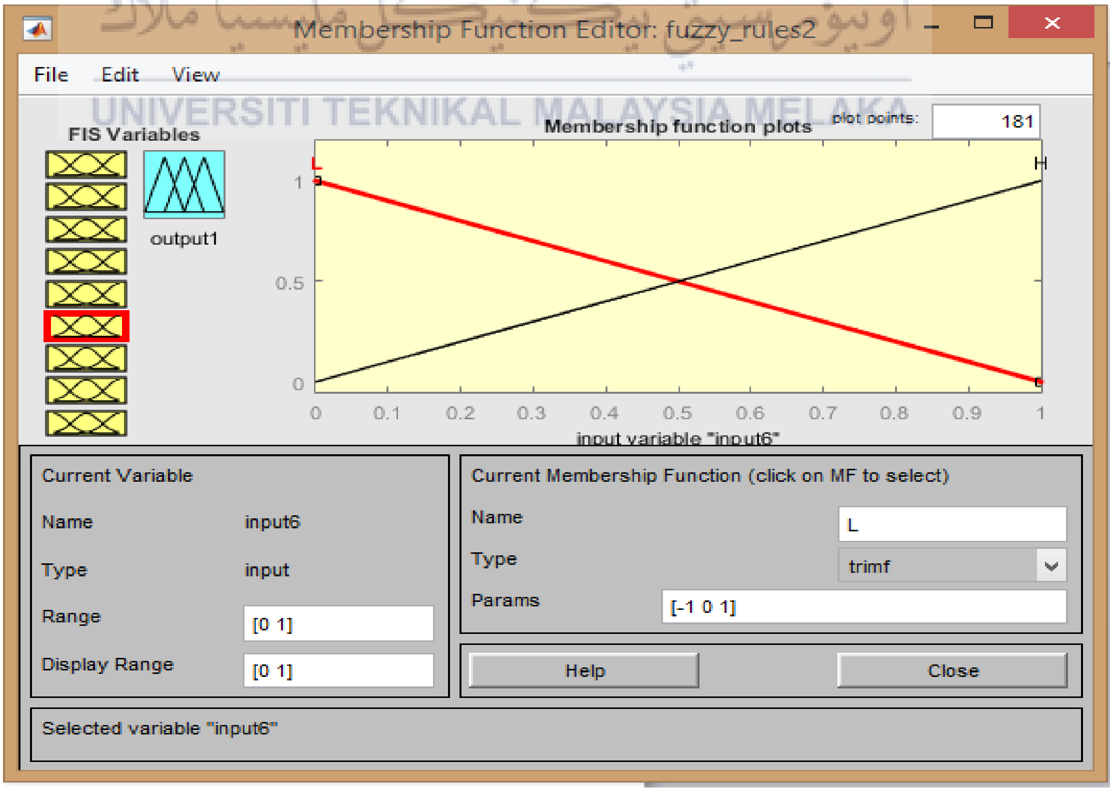

Figure 17: Fuzzy Logic Membership Function for the Sixth Input

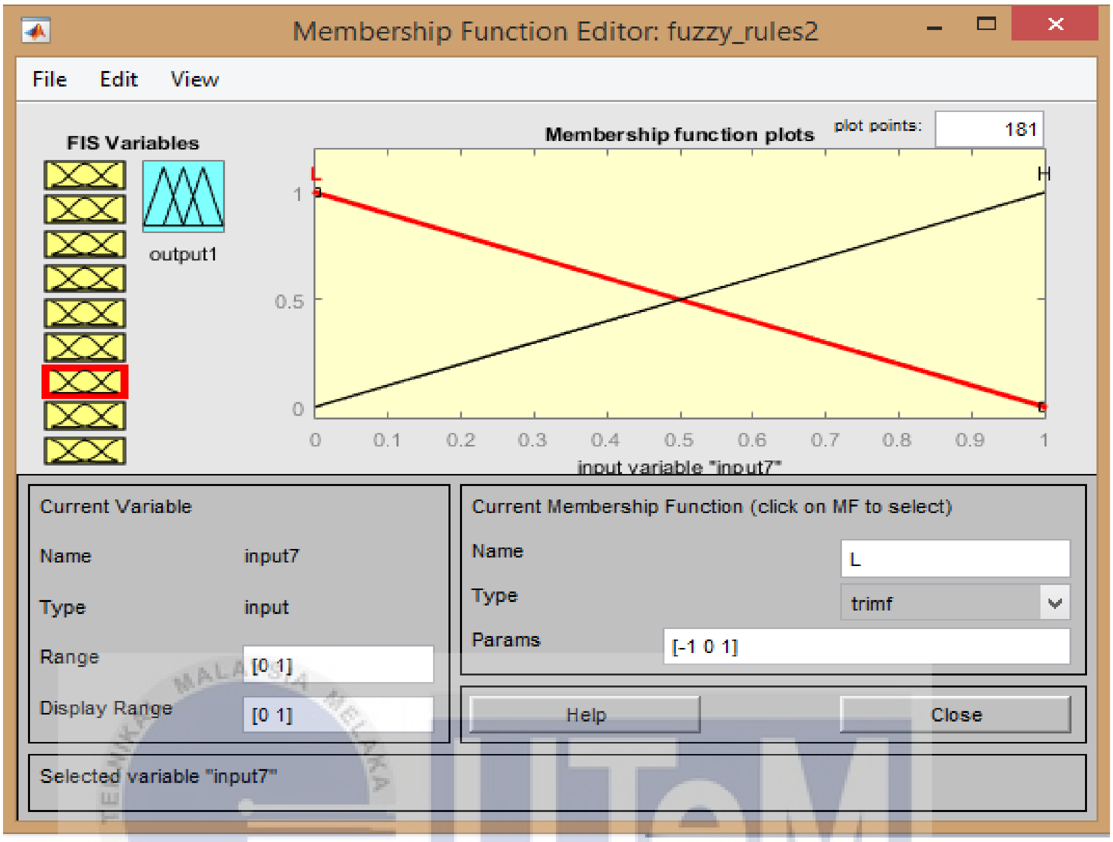

Figure 18: Fuzzy Logic Membership Function for the Seventh Input

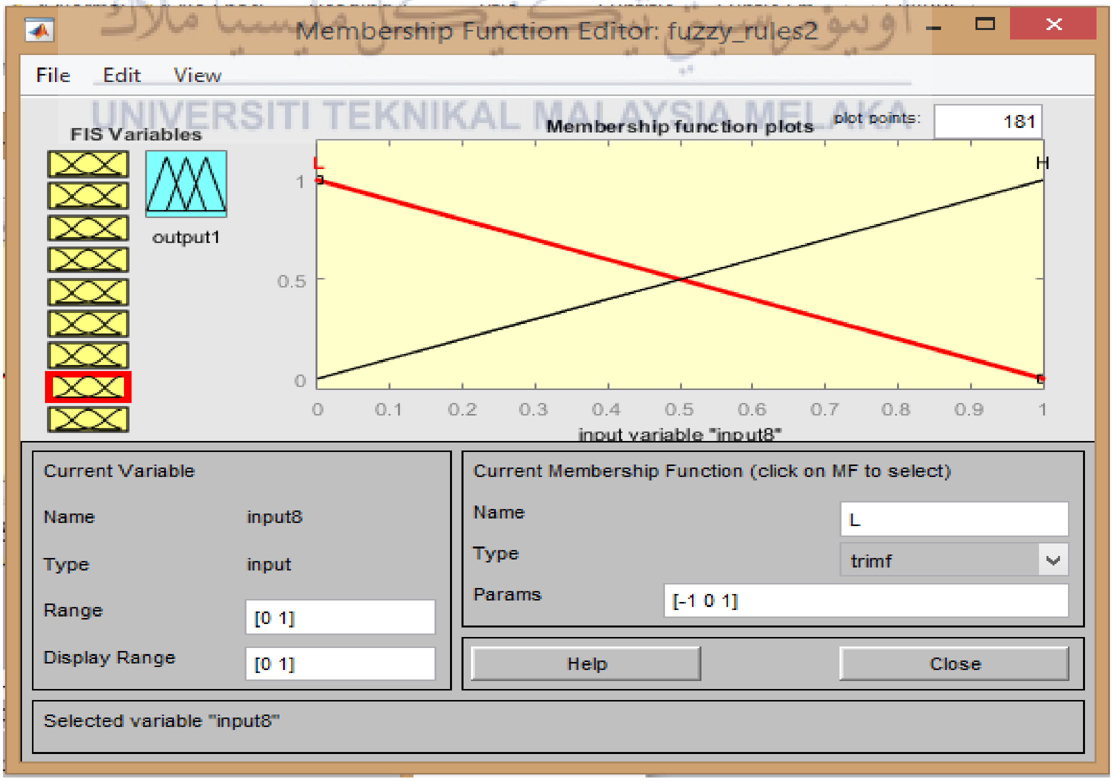

Figure 19: Fuzzy Logic Membership Function for the Eighth Input

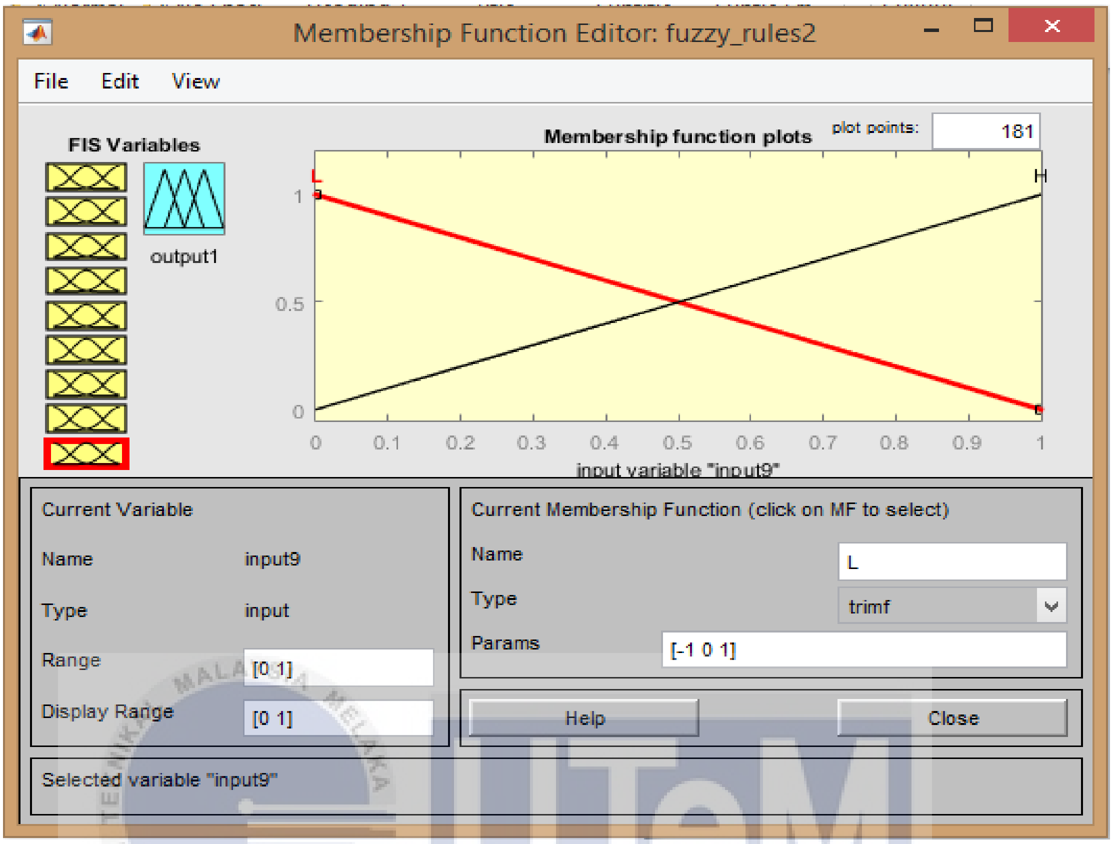

Figure 20: Fuzzy Logic Membership Function for the Ninth Input

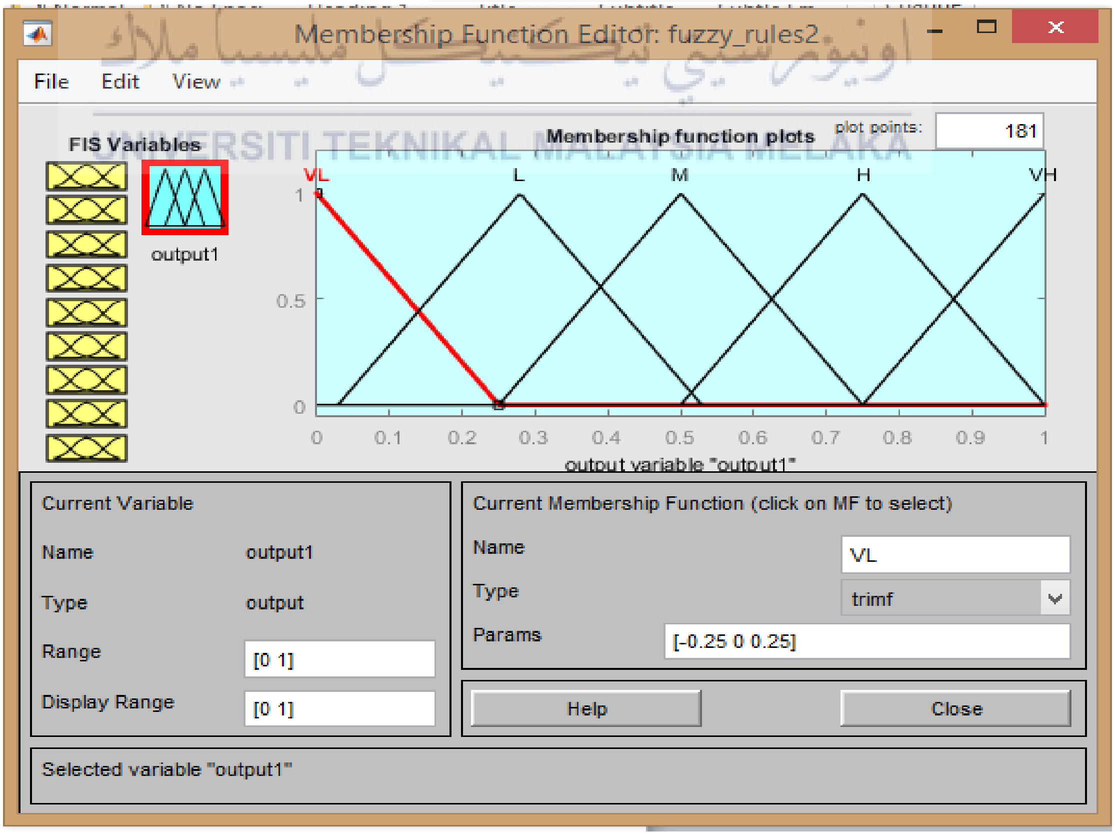

Figure 21: Fuzzy Logic Output Membership Functions

	A	B	C	D	E	F	G	H	I
1	0	0.74946	0.8125	0	0	0	0.23684	0.36849	0
2	0	0.71354	0.78125	0	0	0	0.26316	0.31415	0.125
3	0	0.715	0.75	0	0	0	0.15789	0.22063	0
4	0	0.73906	0.6875	0	0	0	0.15789	0.19505	0.125
5	0	0.72007	0.65625	0	0.2	0	0.13158	0.27363	0
6	0	0.75437	0.625	0	0	0	0.34211	0.30953	0.125
7	0	0.73859	0.5625	0	0	0	0.18421	0.21134	0.25
8	0	0.74172	0.46875	0.2	0.2	0	0.26316	0.29204	0.125
9	0	0.73353	0.4375	0	0	0	0.31579	0.34566	0.125
10	0	0.75375	0.40625	0.08333	0	0	0.31579	0.43607	0
11	0.1	0.67945	1	0	0.13333	0	0.39474	0.46467	0.375
12	0	0.65678	0.72093	0	0.11111	0	0.23684	0.35656	0.25
13	0.03226	0.63409	0.5814	0.30769	0	0	0.68421	0.79261	0.125
14	0.04167	0.50094	1	0.2381	0.04762	0	0.55263	0.58179	0.125
15	0	0.50736	0.77778	0	0	0.05	0.52632	0.52604	0.375
16	0	0.48064	0.66667	0.29412	0.11765	0	0.44737	0.50538	0
17	0	0.49137	0.25	0.35714	0	0	0.36842	0.62117	0.125
18	0.03571	0.5325	0.16667	0.16	0.04	0	0.65789	0.58581	0.25
19	0	0.37109	0.96154	0.17391	0.04348	0	0.60526	0.94149	0.125
20	0.04	0.21267	0.06667	0.05556	0.05556	0	0.47368	0.45916	0.5
21	0.06061	0.1482	1	0.08	0.12	0.04	0.65789	0.57482	0.25
22	0	0.14951	0.44444	0.06897	0	0	0.76316	0.79225	0
23	0.03704	0.1387	0.35556	0.11111	0	0	0.47368	0.45132	0.25
24	0	0.14991	0.13333	0.14286	0	0.09524	0.55263	0.52698	0.125
25	0	0.83527	0.90476	0.15789	0.05263	0.05263	0.5	0.4947	0.25
26	0	0.87353	1	0.04545	0	0	0.48889	0.27387	0.25
27	0	0.92457	0.55556	0	0	0	0.2	0.17431	0.25
28	0	0.89549	0.22222	0.25	0	0	0.44444	0.26743	0.625

Figure 22: Training Data Used for the 3 Systems Used for Experiments

0	0.6344	0.89189	0.25	0.25	0	0.14286	0.13024	0
0	0.63794	0.86486	0.25	0.25	0	0.14286	0.12558	0
0	0.63321	0.83784	0	0.16667	0	0.10714	0.06032	0.125
0	0.62544	0.78378	0	0.16667	0	0.10714	0.05985	0.125
0	0.62423	0.72973	0.25	0.25	0	0.14286	0.09684	0.125
0	0.73671	0.7027	0	0	0	0.19643	0.08554	0
0	0.64214	0.67568	0	0.16667	0	0.10714	0.06534	0.125
0	0.86454	0.64865	0	0.25	0	0.07143	0.04119	0
0	0.62323	0.62162	0	0.16667	0	0.10714	0.05904	0.25
0	0.61974	0.59459	0.28571	0.07143	0	0.25	0.21928	0
0	0.64329	0.56757	0	0.16667	0	0.10714	0.08428	0
0	0.86454	0.54054	0.25	0.25	0	0.07143	0.04524	0
0	0.62475	0.51351	0	0.14286	0	0.125	0.07738	0.125
0	0.62127	0.48649	0.2	0.1	0	0.17857	0.20977	0
0	0.64292	0.45946	0.21429	0.14286	0	0.25	0.12449	0.25
0	0.62683	0.43243	0.2	0.06667	0	0.26786	0.2136	0
0	0.64444	0.40541	0.18182	0.09091	0	0.19643	0.12922	0.125
0	0.63416	0.37838	0.28571	0.04762	0	0.375	0.34589	0.125
0	0.64262	0.35135	0	0.25	0	0.14286	0.10419	0
0	0.6066	0.32432	0.28571	0	0	0.125	0.13432	0
0	0.6148	0.2973	0	0	0	0.07143	0.07735	0
0	0.6292	0.27027	0	0.125	0	0.14286	0.13289	0
0	0.86354	0.24324	1	0	0	0.03571	0.07465	0
0	0.63261	0.21622	0	0	0	0.14286	0.13362	0
0	0.62436	0.18919	0.55556	0.11111	0	0.16071	0.23913	0
0	0.64438	0.10811	0	0.08333	0	0.21429	0.1451	0.125
0	0.7222	0.08108	0.4	0	0	0.17857	0.21519	0
0	0.63718	0.05405	0	0.11111	0	0.16071	0.10219	0.125

Figure 23: Testing Data Used for the 3 Systems Used for Experiments

ATTACHMENTS

1.1	Log book
1.2	User Manual

REFERENCES

Loganathan,C and Girija,K. (2014). “Investigations on Hybrid Learning in ANFIS.” International Journal of Engineering Research and Applications.

Battiti, R. (1989). “Accelerated Backpropagation Learning.” USA: California Institute of Technology, Pasadena, CA 91125.

Hamdan, H. (2013). “An Exploration of the Adaptive Neuro-Fuzzy Inference System (ANFIS) in Modelling Survival.” UK: School of Computer Science, Nottingham University, pp. 80-90.

Papli´nski.A (2005). “Neuro-Fuzzy Computing.”

Nawi, N., Khan, A., Rehman, M. (2013). “A New Levenberg Marquardt Based Back Propagation Algorithm Trained with Cuckoo Search.” Proceedings of the 4th ICEEI.

Sarkar, K. and Nasipuri, Ghose, S. (2010). “A New Approach to Keyphrase Extraction Using Neural Networks.” IJCSI, VoL7.

Keyan, M and Srinivasagan, K. (2012). “Multi-Document and Multi-Lingual Summarization using Neural Networks.” IJCA.

Kaikhah,K. “Text Summarization Using Neural Network.” Department of Computer Science, Texas State University.

Gung, J and Kalita, J. (2012). “Summarization of Historical Articles Using Temporal Event Clustering.” Department of Computer Science, Miami University and Colorado University.

Pallavi, D and Kulkarni, N. (2014). “Text Summarization Using Fuzzy Logic.” IJIRAE, VoL 1, Issue 3.

Anjali, D and Lobo, L. (2013). “Text Summarization Using Clustering Technique.” IJEET, VoL 4, Issue 8.

Suanmali, L, Salim, N and Binwahlan, M. (2009). “Fuzzy Logic Based Method for Improving Text Summarization”. IJCSIS, VoL 2.

Rucha, S and Apte, S. (2012). “Improvement of Text Summarization using Fuzzy Logic Based Method.” IOSRJCE, VoL 5, Issue 6.

Kyoomarsi, F, Khosravi, H, Eslami, E and Davoudi, M. (2010). “Extraction- Based Text Summarization Using Fuzzy Analysis.” Iranian Journal of Fuzzy System, VoL 7, pp.15-32.

Ladekar,A , Mujumdar, A , Nipane, P, Titar, S and Kavitha, S. (2012). “Automatic Text Summarization Using Fuzzy GA-GP.” IJERA, VoL 2.

Megala, S, Kavitha, A, and Marimuthu, A. (2014). “Enriching Text Summarization Using Fuzzy Logic.” IJCSIT, VoL 5.

Sarda, A and Kulkarni, A (2015). “Text Summarization Using Neural Network and Rhetorical Structure Theory.” IJARCCE, Vol 4, Issue 6.