SELF LEARNING NEURO-FUZZY MODELING USING HYBRID GENETIC PROBABILISTIC APPROACH FOR ENGINE AIR/FUEL RATIO PREDICTION

BAYADIR ABBAS AL-HIMYARI

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2017

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

PERAKUAN KERJA TESIS / DISERTASI
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

BAYADIR ABBAS HUSSEIN

calon untuk Ijazah PhD
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"SELF LEARNING NEURO-FUZZY MODELING USING HYBRID GENETIC PROBABILISTIC APPROACH FOR ENGINE AIR/FUEL RATIO PREDICTION"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : *21 Februari 2016.*
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:
February 21, 2016.

Pengerusi Viva: *(Chairman for VIVA)*	Prof. Dr. Suhaidi Hassan	Tandatangan *(Signature)*
Pemeriksa Luar: *(External Examiner)*	Prof. Dr. Safaai Deris	Tandatangan *(Signature)*
Pemeriksa Luar: *(External Examiner)*	Prof. Dr. Khairuddin Omar	Tandatangan *(Signature)*
Pemeriksa Dalam: *(Internal Examiner)*	Dr. Azizi Ab Aziz	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	Assoc. Prof. Dr. Azman Yasin	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	Prof. Dr. Horizon Walker Gitano-Briggs	Tandatangan *(Signature)*

Tarikh:
(Date) *February 21, 2016*

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Pembelajaran Mesin merupakan pembinaan model yang boleh mempelajari dan membuat ramalan berasaskan data. Pengekstrakan peraturan dari dunia data sebenar, kebiasaannya dicemari oleh bunyi, kekaburan dan ketidakpastian. Sistem Neuro-Kabur (NFS) yang diketahui dalam meramal prestasi, mempunyai kesukaran dalam menentukan bilangan peraturan yang sesuai dan bilangan fungsi keahlian bagi setiap peraturan. Penambahbaikan hibrid Algoritma Genetik Pengkelasan Bayesian Kabur (GA-FBC) dicadangkan untuk membantu NFS dalam mengekstrak peraturan. Pemilihan ciri dilakukan di dalam tahap peraturan bagi menyelesaikan masalah FBC yang bergantung pada kekerapam ciri yang terarah pada pengabaian corak kelas kecil. Dalam keadaan dunia sebenar masalah multi-objektif seperti ramalan nisbah Udara / Minyak (AFR) telah diguna pakai. GA-FBC menggunakan maklumat bersama entropi, yang mengambilkira perkaitan di antara sifat-sifat ciri dan sifat-sifat kelas. Fungsi kecergasan adalah dicadangkan untuk menangani masalah pelbagai objektif tanpa pemberat dengan menggunakan kaedah komposisi baru. Model ini telah dibuat perbandingan dengan algoritma pembelajaran yang lain seperti algoritma Pengelompokan Kabur C-Min, (FCM) dan algoritma Pecahan Grid. Ketepatan ramalan dan kerumitan dalam Sistem Kabur Berasaskan Peraturan (FRBS) termasuk bilangan peraturan dan bilangan syarat pada setiap peraturan telah diambilkira sebagai syarat penilaian. Perbandingkan juga dibuat dengan GA-FBC yang asal bergantung kepada kekerapan yang tiada dalam Maklumat Bersama (MI). Keputusan pengujian menggunakan set data AFR menunjukkan bahawa model baharu ini dapat membawa kepada penurunan bilangan atribut dalam peraturan dan boleh meningkatkan prestasi berbanding model lain. Kajian ini membolehkan berlakunya penjanaan sendiri FRBS daripada data sebenar. GA-FBC boleh digunakan sebagai satu arah baru dalam penyelidikan pembelajaran mesin. Kajian ini menyumbang dalam mengawal pelepasan asap kenderaan bagi membantu mengurangkan punca pencemaran untuk menghasilkan persekitaran yang lebih hijau.

Kata kunci: Algoritma Genetik, Pengkelasan Bayesian Kabur, Pemilihan ciri, Pengekstrakan peraturan, Maklumat Bersama Entropi

Abstract

Machine Learning is concerned in constructing models which can learn and make predictions based on data. Rule extraction from real world data that are usually tainted with noise, ambiguity, and uncertainty, automatically requires feature selection. Neuro-Fuzzy system (NFS) which is known with its prediction performance has the difficulty in determining the proper number of rules and the number of membership functions for each rule. An enhanced hybrid Genetic Algorithm based Fuzzy Bayesian classifier (GA-FBC) was proposed to help the NFS in the rule extraction. Feature selection was performed in the rule level overcoming the problems of the FBC which depends on the frequency of the features leading to ignore the patterns of small classes. As dealing with a real world problem such as the Air/Fuel Ratio (AFR) prediction, a multi-objective problem is adopted. The GA-FBC uses mutual information entropy, which considers the relevance between feature attributes and class attributes. A fitness function is proposed to deal with multi-objective problem without weight using a new composition method. The model was compared to other learning algorithms for NFS such as Fuzzy c-means (FCM) and grid partition algorithm. Predictive accuracy and the complexity of the Fuzzy Rule Base System (FRBS) including number of rules and number of terms in each rule were taken as terms of evaluation. It was also compared to the original GA-FBC depending on the frequency not on Mutual Information (MI). Experimental results using Air/Fuel Ratio (AFR) data sets show that the new model participates in decreasing the average number of attributes in the rule and sometimes in increasing the average performance compared to other models. This work facilitates in achieving a self-generating FRBS from real data. The GA-FBC can be used as a new direction in machine learning research. This research contributes in controlling automobile emissions in helping the reduction of one of the most causes of pollution to produce greener environment.

Keywords: Genetic Algorithms, Fuzzy Bayesian classifier, Rule extraction, Feature selection, Mutual Information Entropy.

.

Acknowledgement

First of all I have to express my thanks and gratitude to Allah who gives me the ability to achieve this imperfect work and without his blessing and support nothing can be done.

I would like to express my sincerest thanks and deepest gratitude to my supervisors Assoc. Prof. Dr. Azman Yasin and Prof. Dr. Horizon Gitano for their excellent guidance, caring, patience, encouragement and sharing of all their research experiences throughout these challenging years.

To my father, without his encouragement, advice, and sacrifice – none of this would have been possible. I hope he is proud of me. To my mother, who gave me life and prayed for me all the time, may Allah continuously bless her with good health. To my two brothers and sister, thanks for their love and support. To my husband Azher and my beautiful children Mustafa and Fatima, thank you for being there for me, without them my goal would not have been achieved. I dedicate this work to my family.

I thank all the workers in the UUM University and School of Computing to offer all support and facilities to complete this simple work.

Last but not least, thanks to all those who have been directly and indirectly involved in helping me complete this research.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AFR	Air/Fuel Ratio
ANFIS	Adaptive Network – based Fuzzy Inference System
BC	Bayesian Classifier
BN	Bayesian Network
CLT	Coolant Engine Temperature
DB	Data Base
DTGA	Decision Tree and Genetic Algorithm
ECU	Engine Control unit
FBC	Fuzzy Bayesian Classifier
FCM	Fuzzy c-means
FIS	Fuzzy Inference System
FRBS	Fuzzy Rule Base System
FS	Fuzzy Systems
GA	Genetic Algorithms
GA-FBC	Genetic Algorithm based Fuzzy Bayesian Classifier
GFS	Genetic fuzzy System
IRL	Iterative Rule Learning
KB	Knowledge Base
KNN	K-Nearest Neighbour
LDC	Linear Discriminant Classifier
MAP	Manifold Air Pressure
MAT	Manifold Air Temperature
MF	Membership Function
MIFS	Mutual Information Feature Selection
MOEA	Multi-objective Evolutionary Algorithm
MOEFS	Multi-objective Evolutionary Fuzzy System
MI	Mutual Information
MIM	Mutual Information Maximisation
MLP	Multi-Layer Perceptron
NFS	Neuro-Fuzzy System
NN	Neural Network
PW	Pulse Width – Injection Opening Time
QDC	Quadratic Discriminant Classifier
RB	Rule Base
RBFN	Radial Basis Function Network
RMSE	Root Mean Square Error
RPM	Revolution per Minute - Engine Speed
TPS	Throttle Position

CHAPTER ONE
INTRODUCTION

Developing systems or computer programs with self-learning capabilities is one of the most difficult feats in the field of computer science. Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data.

Machine learning is not only based on algorithms which enable programs or machines to analyse data and learn from it but is also influenced by the representation of knowledge obtained in the learning process. A learning system is usually capable of informing its user of the learning it has acquired. Therefore, a user, in addition to learning about the problem of interest, can also ascertain if the representation of knowledge within the learning system is accurate and credible.

Induction is a frequently used methodology for learning systems. This implies that a learning algorithm processes samples that can lead to an accurate output for any set of input data. Learning algorithm samples that are based on real life data are generally corrupted with noise, indistinctness, imprecision, uncertainty, incompleteness, or vagueness.

Many a time, it is necessary that the learning model is readable by humans. Rule sets are one of the most understandable kinds of models in this regard. In this thesis, the different learning models in the form of rule sets are discussed.

Linguistic rules or fuzzy rules are defined as “if-then” rules that use linguistic expressions (for example, “if x is small, then y is approximately zero”). Fuzzy systems (FS) are learning systems that are based on fuzzy rules or linguistic rules. Fuzzy controllers are FSs that are prevalently used in control engineering. The term “fuzzy logic” is extensively used to refer to the various types of implementation of FSs.

The term soft computing, coined by Zadeh, refers to the methods of human logic and reasoning, which purposely uses human acceptance of uncertainty and vagueness to achieve easy-to-handle economical solutions to problems. Besides FSs, soft computing also includes evolutionary computation, neural networks (NNs), and probabilistic reasoning, and their combinations. As these soft computing methods are all mathematical methods, these can also be incorporated under the subject of computational intelligence. At present, other than control applications, FSs are also used to solve data analysis problems such as data classification, time series prediction, or function estimation. The benefits of using FS include development of uncomplicated instinctive models for data interpretation and prediction. Fuzzy rules also enable easy integration of preceding knowledge. To use FSs in data analysis, there must be a provision for the FS to learn from examples.

It is crucial to create data models that have the capability to generalise data to explain a process with accurateness, in spite of the nonlinearity and intricacies in place. Different methods such as NNs, vector machines, and FS have been developed by the machine learning groups to support nonlinear function approximation.

Nevertheless, the research on FS has revealed that membership and probability are two entirely different concepts. In this study, the relation of FS to function approximation and probabilistic uncertainty in data is considered, within the context of probabilistic FS that explicitly and concurrently deal with two complementary uncertainties (fuzziness or linguistic uncertainty and probabilistic uncertainty) that depend on probability measures of fuzzy events. The two key innate properties of modelling real-world applications comprise uncertainty and impression, and it is important that these be integrated into the learning algorithms.

Characteristically, real-world models are built using training examples to enable dependable predictions. This approach of making predictions is very different from the one that uses a set of predefined rules. Consequently, machine learning enables a machine to learn from its own experience with its own capability without the intervention of any human expert, which is impossible to achieve for real world or large systems. These real world systems pose various insurmountable problems such as large dimensions, time-varying properties, huge volumes of data and noisy measurements, and the necessity for interpretation of the resulting model.

Minimising the magnitude of air pollution, caused by automobile engine exhausts and emissions, which leads to close to 50% of air pollution (Mahajan, 2013; Wargo, Wargo, & Alderman, 2006), is one of the most challenging real world problems that researchers are trying to mitigate. The emissions from automobiles contain carbon monoxide, hydrocarbons, NOx, etc., and are usually attributed to control problems in the automobile engines. The highly common automobile engine issue that leads to emissions is air/fuel ratio (Barghi & Safavi, 2012; Ebrahimi et al., 2012) prediction and lack of control in an electronic fuel injection engine. This is a nonlinear problem that has a huge search space.

1.1 Problem Statement

Decision rules are devised using a set of training data in a rule extraction or induction system. The functioning of a classifier that utilises the rules to categorise unseen objects is ascertained by a set of decision rules. Hence, it is of utmost importance that a rule abstraction system creates decision rules with high predictability.

Using FS is a simple and effective approach for domain experts to design systems for modelling. The subjective approach of designing a fuzzy system can be adopted to represent domain knowledge of experts. Nevertheless, domain experts do not organise their decision-making process in a formal way. Consequently, the process of transmitting expert knowledge to a knowledge base (KB) where it is rendered usable is laborious and nontrivial. Additionally, absolute reliability on human knowledge and experience may lead to certain serious problems, as even human knowledge is often inadequate and intermittent, instead of being systematic. (L. Wang & Fu, 2006; Zarandi, Mohammadhasan, & Bastani, 2012).

There is an improvement in the computational efficacy of the models due to the combined usage of FS and NNs. The Neuro-fuzzy system (NFS) unites the linguistic rule analysis of fuzzy inference systems (FIS) with the learning abilities of NNs. NNs integrate nonlinear permutations of data set elements into their results. The computational power of NNs is largely steered by the immensely parallel distributed structure of NNs and their capacity to learn and generalise. Even though NFSs have a great generative ability for tuning fuzzy rules, they are not capable to automatically produce fuzzy rules from data. As an application in this work, the air/fuel ratio problem, there were a number of researchers that participated in this area using various models as FSs and NFSs (Bose & Kumar, 2007; Cao, Du, Peng, & Yin, 2006; Ghaffari, Shamekhi, Saki, & Kamrani, 2008; S. H. Lee, Howlett, & Walters, 2004; Liu & Zhou, 2010; Wong, Wong, Vong, & Wong, 2016). They are share that the models are not self-learning and the rules are gathered from the domain experts.

There a number of algorithms for the self-learning of NFS as clustering and grid partition, but they usually have either lack of the interpretability of the fuzzy rules or curse of dimensionality (Jin, 2012; Lakhmi & Lim, 2010; Qin, Langari, & Gu, 2015; Rigatos, 2011).

The creation of a fuzzy rule base (RB) comprises the number of antecedents in the condition part of every rule, the number of fuzzy rules, and the rule class in the consequent part of each rule, which in turn can lead to an increase in complexity and inaccuracy of the end system. Conventionally, models with high syntactic complexity tend to be highly accurate, while models with low syntactic complexity tend to be low on accuracy. Striking a trade-off between the complexity of the fuzzy RB and the accurateness of the end system will lead to a multi-objective assessment problem, where most of the work in this field will be managed with an aggregate method known as weighted sum (Dehuri, Patnaik, Ghosh, & Mall, 2008; Gacto, Alcalá, & Herrera, 2010; Ishibuchi, Nakashima, & Murata, 2001; Ishibuchi & Nojima, 2007; Ishibuchi & Yamamoto, 2004). But there is one inevitable problem of this approach and that is the generation of a set of weights which correctly measures the objectives when nothing is known about the problem, which might be difficult if the number of objectives is high (Mandal & Pal, 2011; Xu & Zhou, 2011). These weights are generally identified using a trial and error method.

For real-world problems that allow multiple inputs, data fusion is implemented to merge the data and information from various sources including sensors. The methods of probabilistic data fusion are usually based on Bayes' rule for uniting prior and present observation information. Bayesian statistics offer a spontaneous manner to merge observations from various sensors (Khaleghi, Khamis, Karray, & Razavi, 2013; Koch, 2010). The consequent part of a rule is determined using Bayesian classifier. This leads to a high-speed, precise, and almost optimal method that can speed up rule generation

Bayesian classification is based on Bayes' theory that is governed by the prior probability P(H), and likelihood ratio P(e/H), to calculate the posterior probability (H/e), where H is the hypothesis and e is the event. In Fuzzy Bayesian classifier (FBC), the event is the fuzzy set (antecedent part) that uses the fuzzy membership grades and the hypothesis is the rule consequent (class). The conditional probabilities between the events and the hypothesis determine the functioning of the Bayesian classifier, and the rule class that has the highest posterior probability is identified as the predicted class of that rule.

The conditional probability that is computed from the frequency of events, without any information pertaining to the extent of uncertainty or dependency between the events and hypothesis, Bayesian classifier will have a tendency to put all patterns of the smaller class to error, which represents that no information is derived from classifiers. This information might result in the improvement of mapping between the fuzzy sets and the rule class in the rule extraction problem, which may lead to an increase in accuracy (Hu, 2014, 2015; X. Zhang & Hu, 2014).

Mutual information (MI), as a concept in information and probability theories, explores the quantitative relation between arbitrary variables, i.e. what information one contains about another and how it is dependent on it respectively. The mutual information also relates to the attribute significance by its success or deterioration in estimating data advancement. Namely the attribute in question can be regarded as the fuzzy sets antecedent when it comes to the fuzzy rule classification area.

Information theoretic concepts can provide explanation for multiple Bayesian quantities through the application of evidential entropy analogy. The Bayesian classifier (BC) can be enhanced through the mutual information as a likelihood ratio. In addition to that, the complicatedness of each rule can be simplified by reducing the antecedent choices for the rule by increasing the informational value of the variable relations and the data load they carry, thus providing additional informative attributes. Mutual and Bayesian information classifiers can in fact operate together towards problem solving in the field of classification and machine learning, and should be regarded as interconnected methods instead of opposing ones (Hu, 2014, 2015).

1.2 Research Questions

- How to collect the data.
- How to initialize the FRBS parameters.
- How to enhance the feature selection and rule extraction approach.
- How to improve the evaluation function for the optimization of fuzzy rules.
- How to evaluate the NFS based the generated fuzzy RB.

1.3 Research Objectives

One of the main objectives of this research is to develop an enhanced NFS to achieve a high accuracy without increasing the complexity of the system. The other objective is constructing the FRBS for the NFS automatically without the need of expert. The specific objectives of the research are:

- To collect the data using a test engine.
- To initialize the FRBS parameters using the FCM.
- To enhance the feature selection and rule extraction approach by applying them simultaneously using FBC based MI.
- To improve the evaluation function by reducing its complexity through discarding the weights via Composition method.
- To evaluate the NFS based generated fuzzy RB in terms of accuracy and complexity.

1.4 Significance of the Research

The design of the information concerning prediction problems in the vast majority of the real world is the product of either sensor measurement or human calculations. The human aspect, however, presents more of a difficulty since nobody has so far succeeded in establishing a relevant mathematical model for such an operation, which causes the necessity of a fuzzy rule introduction to an intricate but wholesome generation system.

The research aims at involving the fuzzy rule-based system in the operation of more advanced systems (ex. automobile engines) through the application of soft calculation approaches, which may result in several major improvements in the field, such as:

- Building a fuzzy rule based system for the purpose of predicting air fuel ratio;
- Environmental protection through more efficient control of automobile gas emissions, enhancement of the fuel economy through fuel efficiency stimulation, and boosting engine capabilities towards engines' top performance – all accomplished with the help of precise fuzzy forecasts.
- No expertise requirement, where a machine learning approach is applied.
- Simplicity during building fuzzy inference system and reducing the computational complexity.

1.5 Research Scope and Limitations

The effectiveness of the proposed approach is demonstrated in nonlinear system prediction and complex system with uncertainty.

In this thesis, the proposed approach works only on the rule base generation of fuzzy rules.

Only MISO (multiple inputs, single output) can be classified with the current algorithm system. The time for forecasts and training is not taken into consideration in the application of the algorithm defended in the current thesis.

1.6 Thesis Structure

The current research is divided into a total of six chapters. The first chapter presents the study's introduction. In it, the authors defend their reasoning behind the choice of precisely the NFS for their operational method for data modelling and forecast.

Chapter Two is further broken down into four sub-chapters: 1) The machine learning process – rule formation and learning plus choice of relevant features; 2) Algorithms for soft calculations and their application in the current methodology; 3) Research introduction; and 4) Relevant literature pertaining to the present study.

In Chapter Three, the technique for reaching the study's end goal is discussed in detail and therefore the chapter is also divided into several parts. The thesis behaviour and the research's general framework are discussed in the first part, the ways to get hold of relevant data sets – in the second, and the final assessment of the proposed model – in the third.

Chapter Four focuses on the data gathering stage, including details on the characteristics of the data and preprocessing.

Chapter Five focuses on how exactly the proposed model can be executed, namely the whole process of applying Genetic algorithm FBC for Adaptive Network – based Fuzzy Inference System (ANFIS) modelling and the derivation of fuzzy rules. It also includes the methodology of how to extract the algorithms in question and to select the proper features for their application.

The results of the study are presented in Chapter Six.

The research concludes with Chapter Seven, which acts as a summary of the whole study, demonstrating in short its objectives, methods, results, limitations and possibilities for additional examination of the subject.

CHAPTER TWO
LITERATURE REVIEW

This chapter is made up of four sections. The first section talks about the fundamental ideas of machine learning, such as rule learning and feature selection. The next section depicts concepts pertaining to soft computing algorithms. The third section initiates a discussion on the research applications. The following section sheds light on prior studies pertaining to the area of research in this thesis.

2.1 Machine Learning

The arena of machine learning pertains to the means which can learn from experience on their own without the need for a human intervention or support (Grosan & Abraham, 2011). The experience is typically rendered as learning examples based on which the machine learning methods can learn a model on their own. From time to time, it is quite vital that humans are able to read the learned model. One such most comprehensible kind of models is rule sets.

Machine learning evolves from the processes and conceptions from domains, such as statistics, artificial intelligence, and information theory. These algorithms function by constructing a model derived from example inputs and deploying it for making decisions or estimations. Machine learning is quite related to and frequently overlaps computational statistics, which is also an area specialising in making predictions.

Among the tasks of machine learning, an important one is classification task. This task involves estimating the value of a nominal variable with more than two probable values. Models tackling such tasks could take various forms, including logic programs and linear equations. Two most common models deployed are rules (Flach, 2012) and decision trees (Quinlan, 2014). These models are comprehensible and humans can read and interpret them effortlessly. Rules and decision trees split the space of examples into subspaces. For every subspace, they offer a straightforward estimation or a predictive sub-model. This thesis focuses on the means for learning models through rule sets.

A conventional domain of machine learning is supervised learning or predictive modelling (Hastie et al., 2009). The task associated with this area is as follows. A set of inputs is presented, which is presumed to exert a certain impact on an output. Besides, a set of training or learning examples is provided, wherein every example associates a particular instantiation of the inputs to the respective output values. The objective here is to capitalise on the stated examples of learning to learn a predictive model which estimates the output based on the inputs.

The above mentioned task holds practical importance for several fields, such as the social sciences, medicines as well as different spheres of engineering. The task can typically be construed in two manners. The first construal is common in the system's theory and is alike what is mentioned above. The inputs are deemed as stimuli of the environs to the system and an account of the state of the system. The outputs signify the system response. Learning examples are deemed as input and output measures under various circumstances such as the environment's statuses, influences, and system responses. The task is to build a system model which estimates the system response, on the basis of environmental impacts and the system's internal position. In case of the second construal, each example of learning is deemed as a description of an object. The inputs represent a description of an object, while the outputs are the properties of certain object in which we are interested. The model thus built has to be in a position to estimate the properties of the object based on its description. Both construals are indeed alike, and the choice depends on the problem area as well as individual taste. Yet, the second construal is more commonplace, and hence is deployed in this thesis.

2.1.1 Feature Selection

In the world of data mining and machine learning, feature selection has been extensively scrutinised. In this regard, an aspect known as variable or attribute signifies a property of a system or practice which has been quantified or built based on the actual input variables. Feature selection aims to choose the smallest feature subset in view of a particular generalisation error, or otherwise determining the best feature subset with k features, which generate the least generalisation error. Other aims of feature selection are: (i) offer a better generalisation and a faster response with unseen data; (ii) attain an improved and effortless understanding of the process which produces the data (Guyon & Elisseeff, 2003; Karegowda, Jayaram, & Manjunath, 2010); (iii) enhance the generalisation performance in terms of the model constructed using the entire set of features (Mansoori, Zolghadri, & Katebi, 2007).

Under feature selection, we endeavour to avoid choosing excess or minimal features than required. In case inadequate features are chosen, the information content to maintain the data concept is debased. In case excess features are chosen, which also include irrelevant or redundent features, the classification accurateness might be lower owing to the intrusion of irrelevent information. In applications of data mining, the elimination of irrelevant features is vital for revealing concealed relationships between features, as well as between features and class targets of patterns. Inappropriate diminution of data dimensionality causes information loss and might debase the quality of data mining tasks, including the classification task.

Feature selection is deployed as a pre-processing step or along with a learning machine for the purpose of regression or classification. Methods of feature selection methods (Guyon, 2006) are typically categorised into three key groups:

1. Filter methods: Filters choose features purely on the basis of statistics only. These are basically pre-processing methods, which try to evaluate the virtues of features from the data, disregarding the impacts of the chosen feature subset on the learning algorithm's performance. These methods can be divided into two: ones which assess all features in one pass and ones which assess manifold proposed feature-subsets along with an heuristic search.
2. Wrapper methods: As the name suggests, wrappers "wrap around" the induction algorithm which is deployed for the final classifier. The method involves an heuristic search wherein the induction algorithm is the heuristic. Needless to say, this method can be quite costly. It typically comprises K-fold cross-validation on the training data. However, there are certain benefits as well. The method assesses the variable set chosen and not merely one at a time. Thus, redundent variables can be eliminated, and variables which are

strictly useful can be determined. The method also determines variables which may be especially helpful with the recommended induction algorithm.

Figure 2.1 depicts a general framework for the wrapper method of classification.

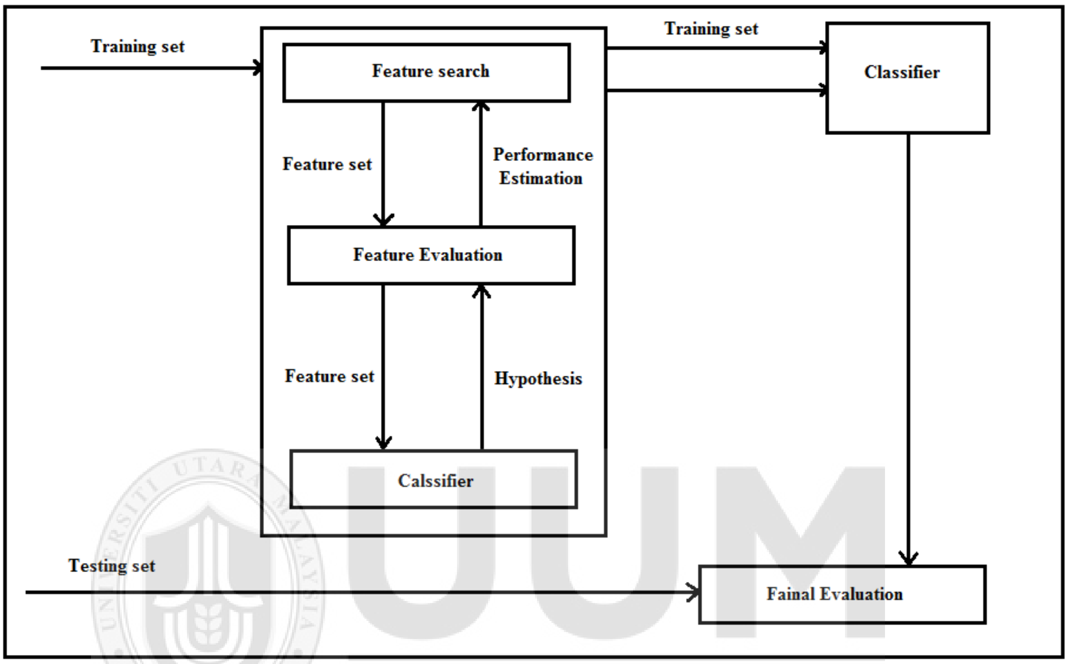


Figure 2.1. General framework for the wrapper method

3. Embedded methods: These methods integrate knowledge pertaining to the particular setup structure of the class of functions utilised by a specific learning machine. The methods carry out feature selection as an element of the learning process and are typically peculiar to the given learning machines.

Even though embedded methods are typically less computationally costly than wrappers, they are much slower when compared to the filter approaches. Furthermore, the features chosen depend on the learning machine. Filter methods presume total independence between the data and the learning machine, and hence utilise a metric independent of the induction learning algorithm so as to evaluate feature subsets. Moreover, these methods are comparatively stronger against over fitting; however, they might fail to choose the best feature subset for the purpose of regression or classification.

2.1.1.1 Objectives of Feature Selection

Various feature selection algorithms might have different objectives to accomplish. Here is a list of the frequent objectives cited by researchers (Kumar & Minz, 2014; Navot, 2006):

1. Determine the minimally sized feature subset which is essential and enough for the target concept.
2. Choose a subset of *N* features from a set of *M* features, $N < M$, such that the criterion function's value is optimised over all subsets of size *N*.
3. Select a features subset for better precision in prediction or reducing the structure size without decreasing prediction accuracy of the classifier constructed using purely the chosen features.
4. Choose a small subset so that the resultant class distribution, in view of only the values for the chosen features, is as close as possible to the actual class distribution, taking into consideration all feature values.

Mining on the reduced features set aids in making the patterns simpler to comprehend, besides decreasing the time taken for computation. This work proposes the wrapper approach for feature selection. As part of the feature selection process, a wrapper approach with genetic algorithm was deployed as a random search technique for subset generation, wrapped with induction algorithm, i.e. the Bayesian inference mechanism as a subset assessing mechanism for choosing relevant features automatically.

2.1.2 Rule Extraction

The rule extraction (RE) procedure could be deemed as a transformation of continuous, symbolic, or distinct knowledge from a dataset into beneficial, representative propositional logical rules which elaborate the knowledge embedded in datasets with adequate precision. The rules, or knowledge, mined during the course of knowledge discovery has to follow the criteria mentioned below:

- Rules have to be brief, uncomplicated, clear, and crisp
- Rules have to be understandable
- Duplication of rules must be avoided
- Rules have to be precise in outlining the data from which they were mined
- Knowledge compressed within the rules must be beneficial. Data mining is a process wherein valid as well as beneficial rules (i.e. knowledge) are mined from big databases through a classification method

In several classification problems, there are no explicit rules; however, examples can be acquired. A classifier, which is a function that outputs a class label for every input object, cannot be built from identified rules. Hence, in machine learning, one endeavours to deduce a classifier from a narrow set of training examples. The usage of examples raises the need to clearly mention the classification rules. The objective is to attain models and learning rules to learn from the examples and estimate the labels of prospective objects.

For the majority of real-world control issues, the information pertaining to the design arrives from sensor measurements or an experienced human controller. In case the environs facing a human controller are so complex that there is no relevant mathematical model, then the task is to devise a fuzzy rule generation system as a substitute for the human controller.

A characteristic of fuzzy logic is that it permits nonlinear input/output relationships to be articulated through a suite of qualitative “if-then” rules. Nonlinear control/decision surfaces, pattern classifiers, and process models might be articulated as fuzzy rules. A human expert hand-crafts the majority of FS so as to capture certain intended input/output relationships which the expert is aiming for. However, sometimes the expert is not able to express his knowledge in an explicit manner. In fact, for several applications, such an expert might not even be existent. Therefore, a significant interest exists in being able to mine fuzzy rules from experimental input/output data automatically.

Model structure as FRBSs comprises an extension to conventional rule-based systems, as they deal with ‘‘IF-THEN’’ rules. However, its antecedents and consequents comprise fuzzy logic statements rather than classical ones. Furthermore, with regards to fuzzy sets with linguistic labels, the output system demonstrates a higher level of interpretability for the expert to comprehend the former’s working procedure, and the inner particulars of the problem characteristics (Fernández, López, del Jesus, & Herrera, 2015; Gacto, Alcalá, & Herrera, 2011).

A fuzzy RB includes several rules stated as:

R: IF $\boldsymbol{x}$ is A_i , THEN $\boldsymbol{y}$ is Bi

where $Ai = \{ A_{i1} , A_{i2} , \ldots , A_{in} \}$, $B_i = B_{i1} ,B_{i2} , \ldots , B_{im}$, and A_{ij} and B_{ik} are, respectively, fuzzy sets which outline an input and output space partitioning, x is the input, and y is the output. The condition, also known as premise, comprises several antecedents which are combined by various operators like AND or OR computed with t-norms.

2.1.2.1 Assessment Criteria for Rule Extraction

There are three measurements that are typically employed to assess rule extraction (Huysmans, Baesens, & Vanthienen, 2006; Huysmans, Dejaeger, Mues, Vanthienen, & Baesens, 2011). These measurements are as follows:

1. *Predictive accuracy*. The induction of a rule possessing a considerably high degree of predictive accuracy is a primary consideration in rule inductive learning. For this reason, the capability of the extracted or refined rule sets derived by using the relevant techniques to make accurate predictions on new and unique cases should be assessed.
2. *Comprehensibility*. Another important aspect that should be taken into consideration is the comprehensibility of learned rule sets. To determine comprehensibility, it should be identified whether the extracted (or refined) rule sets could be inspected and elucidated by domain experts. The measurement of the comprehensibility of learned rule sets presents challenges.

An underlying assumption of the performed experiments is that *syntactic complexity* is directly and positively related to comprehensibility. In this thesis, syntactic complexity is evaluated by using measures that assess the comprehensibility of rule sets: the number of rules in the rule set and the average number of antecedents for each rule. Attempting to achieve high levels of both predictive accuracy and comprehensibility is contradictory because one tends to dominate the other. That is, significantly accurate models tend to exhibit high syntactic complexity, whereas those having low syntactic complexity exhibit low accuracy.

3. *Consistency.* The assessment of consistency is likewise related to the rules generated from a learning model. An extracted rule set is found consistent if rule sets that generate the same classification of unseen examples are derived from a learning model, even in different training sessions.

Rule extraction also has the objective of deriving a subset of interesting rules from all those that were identified. Numerous data mining algorithms were designed to identify accurate, comprehensible, and consistent rules, but most of them fail to identify interesting rules, which is a more difficult and challenging task. Approaches to the identification of interesting rules can generally be classified into subjective and objective methods. The former can be described as driven by users and dependent on domains. For example, the user could identify rule templates and indicate the combination of attributes that should occur for a rule to be found interesting. Another subjective method is one in which a user can provide the system with a general description of previous knowledge regarding the domain, such that the system can identify only the discovered rules that point to knowledge previously unknown to the user. On the other hand, objective methods are driven by data and independent of the domain. A number of these approaches compare a discovered rule with other rules instead of user beliefs. Thus, the underlying principle is that a rule’s level of being interesting relies on both its quality and like comparability to other rules.

This thesis employed objective approaches to identify interesting rules as the measure for assessing discovered rules.

2.2 Soft Computing-based Techniques

The implementation of soft-computing practices has evolved as a viable option in numerous situations to solve extremely intricate, non-linear, and stochastic problems that cannot be solved by conventional methods (K. Gopalakrishnan, Ceylan, & Attoh-Okine, 2009). Soft computing involves intelligent and efficient processing methods that are very different from the complicated and long processing techniques of conventional hard computing.

There are three categories of soft computing: evolutionary computation, FS, and artificial neural computing, with the latter including machine learning (ML), probabilistic reasoning (PR), expert system (ES), etc. as depicted in Figure 2.2.

While each one has its own set of strengths and capabilities, these computational techniques when used in combination (hybrid), along with their balancing attributes, become a robust alternative to solve complicated problems that cannot be solved using traditional mathematical methods.

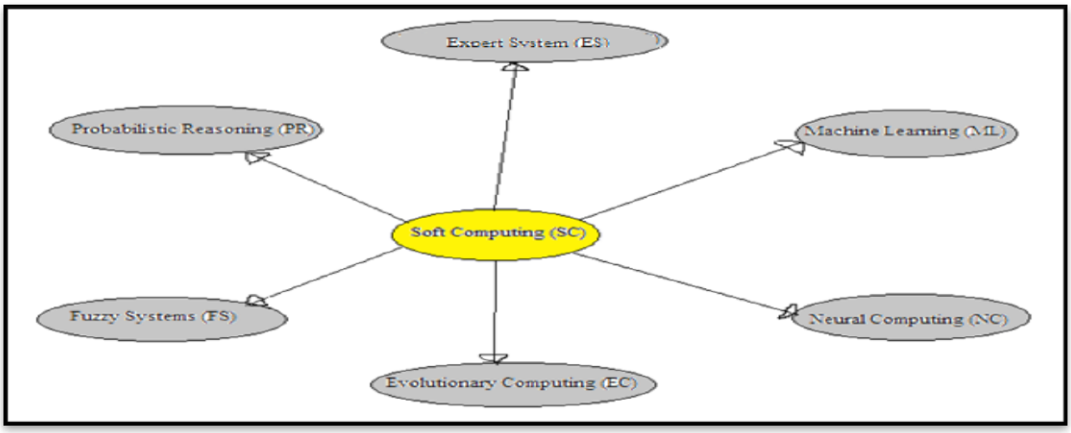


Figure 2.2. Soft Computing Techniques

Table 2.1

Advantages and Limitations of Principle Constituents of SC

Constituents of SC	Advantages	Limitations
GA	Natural progression and optimization	Inability to handle imprecision
FL	Approximation reasoning, imprecision	Inability to learn
NN	Learning and implicit knowledge representation	Inability to optimize
PR	Indecision	Inability to learn

In this chapter, the different soft computing techniques are thoroughly reviewed as the focus of this study is to implement soft computing techniques for modeling a real world application that can predict future values, while making use of the advantages of the techniques and avoiding the limitations as exhibited in Table 2.1. The hybrid model of using soft computing techniques that lets individual techniques overcome their limitations and creates a fused system to achieve greater efficiency and effectiveness through collaboration will also be discussed.

2.2.1 FS

Lotfi Zadeh designed fuzzy logic in the 1960s. Fuzzy logic is a soft computing technique that simulates the learning capability of the human mind and makes logical decisions in an uncertain and imprecise situation. It can be implemented to solve very complex problems that a mathematical model cannot solve because of the non-linear, time varying behaviour and imprecise characteristics of the problem.

FS can be beneficial in two situations: when a very complicated problem whose behaviour cannot be comprehended needs to be solved, and secondly when a fast fairly accurate solution is required (Ross, 2013). FS may be very useful to understand complex systems with no clear models because it can effectually handle indefinable and uncertain information.

The prime objective in using FS is to keep the human aspect under consideration. It is helpful in involving a human expert in the modeling or validation process to ensure that the fuzzy learning is justified. It is extremely convenient for human experts to express their knowledge through linguistic rules than through mathematical forms. Fuzzy learning enables the merging of specialised human knowledge with experimental information through measurement points. Fuzzy logic can be used to integrate the knowledge of human experts with the information from databanks to recreate a more specialised KB.

2.2.1.1 Linguistic Variable

The quantification and reasoning applied to ambiguous or fuzzy terms contained within our natural language are referred to as fuzzy logic. Vague or fuzzy terms are called linguistic or fuzzy variables.

2.2.1.2 Fuzzy Set and Membership

Uncertainty is rooted in inprecision measurements attributable to the unreliability of tools or other factors. It can also be attributed to vagueness language. Linguistic variables are frequently employed to describe and characterise physical objects and situations.

Therefore, rather than defining crisp sets, where elements are either in or out of the set with absolute certainty, Zadeh posited the idea of a MF instead of defining crisp sets with elements that are either absolutely in or out of such sets.

Figure 2.3 shows how fuzzy sets could represent the idea of vehicle speed. The figure also illustrates how we can differentiate between the concepts of high and very high in order to establish a model that is more reliable than the classical set theory.

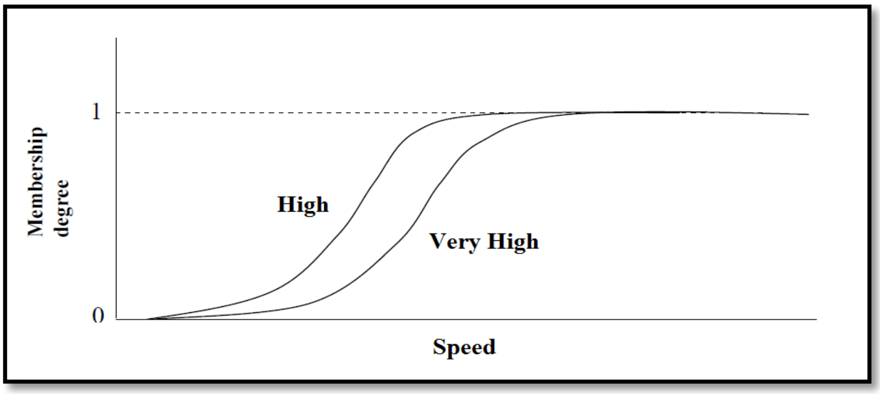


Figure 2.3. Fuzzy set representation

2.2.1.3 Types of MFs

Actual data must be converted into fuzzy data according to specific MFs to be able to handle the fuzzy data. MFs are selected depending on the application. Such functions assume varying shapes, including triangular, trapezoidal, Gaussian, and bell-shaped.

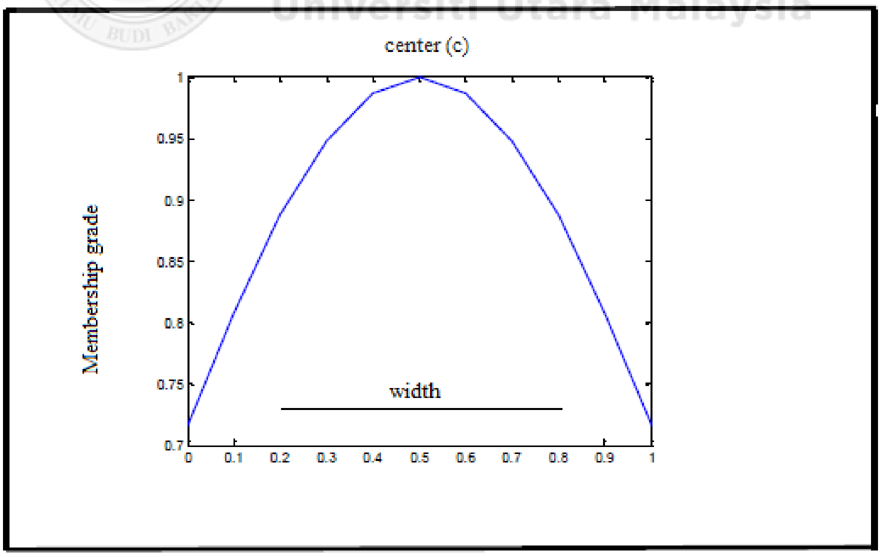


Figure 2.4. Gaussian MF

Figure 2.4 shows the symmetric Gaussian MF used in this work (Ishibuchi, Nakashima, & Morisawa). This MF is expressed as

$$f(x,\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \qquad (2.1)$$

where parameter c determines the distance from the origin, whereas σ stands for the width of the curve. Gaussian MFs are characterised as smooth and non-zero at all points.

2.2.1.4 Fuzzy Logic System Methodology

Figure 2.5 reveals the three phases comprising the fuzzy logic system methodology (Negnevitsky, 2005; Ross, 2013):

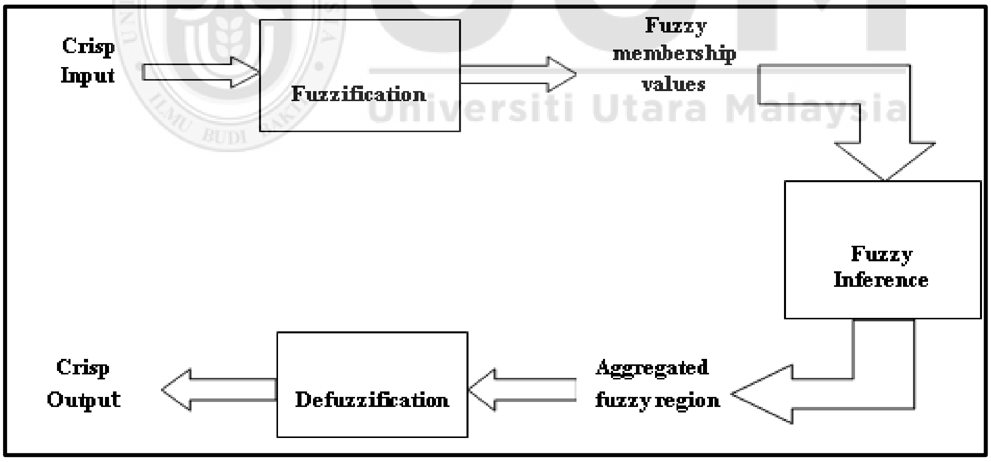


Figure 2.5. Fuzzy logic methodology

1. Fuzzification is a process that refers to the change in input variables from being crisp to being linguistic.

The following are the steps in the fuzzification process:

- Identify a universe of discourse,
- Determine and characterise the linguistic variables,
- Identify the MFs for each linguistic variable within the universe of discourse, and
- Graphically illustrate the MFs graphically using relevant MFs.

2. Fuzzy inference maps use linguistic variables as input for output linguistic variables based on a system of fuzzy rules of the type “IF A THEN B.” Given that input linguistic variables are weighted, output linguistic variables are also derived as weighted. In this phase, the rule outputs are aggregated, which refers to the unification of all rule outputs.
3. In the defuzzification phase, the weighted values of the output linguistic variables obtained through fuzzy inference are changed to become crisp variables on the basis of on MFs. Defuzzification can be achieved using the following main methods:

- First is the maximum defuzzification method in which process an output value is identified using the linguistic variable with the highest weight;
- Second is the centroid calculation defuzzification method that identifies an output value based on the weighted influence of all active output membership functions.

2.2.1.5 Fuzzy System Variants

Chiefly, FS are of three varieties:

- Mamdani fuzzy system: This is also called the linguistic fuzzy system.
- Singleton fuzzy system: The intricacy of defuzzification of a linguistic fuzzy system can be made simpler by limiting the output to a singleton MF. With no assimilation being carried out in a numerical manner, there is a decrease in the computational demand for the learning and assessment of the fuzzy system. Hence, this system is the most commonly used in the industry.
- Takagi-Sugeno fuzzy system: This can be said as an extension of the above mentioned system, wherein the function *f* is not considered as a fuzzy set. However, the basic premise of this system is linguistically explainable. Models based on the Takagi-Sugeno system present a good interpretation for a dynamic process modelling. A singleton fuzzy system could be retrieved from a Takagi-Sugeno fuzzy system if the function *f* is taken as a constant. As the constant could also be deemed as a zeroth order, the Taylor series expansion of the function *f* is also known as the zeroth order Takagi-Sugeno fuzzy system.

2.2.2 NNs

Drawing inspiration from the biological nervous system, neural networks function according to the principle of mostly interrelated simple elements functioning as a network function. In this process, there is no assumption of prior knowledge; however, records data, observations, and measurements are taken into consideration.

NN research is based on the fact of learning from data so as to mimic the biological ability of linear as well as nonlinear problem solving. Typically, the NNs can be devised and trained for addressing those issues that are tough to resolve for humans or the regular computational algorithms.

The training's computational meaning amounts to the adjustments of specific weights that are the main aspects of the NNs.

A typical NN is made from different simple components known as neurons. Every neuron is associated with few of the other components. The transmission of information takes place by means of these connections or links. A neuron comprises input as well as output parameters, besides an activation function. Every connection is characterised by a specific weight. The input parameters are multiplied with the weight of the connection; the parameters are then summed. This outcome is then subjected to the activation function that provides a specific output. The activation function has to signify the nonlinear operation as performed in the brain, wherein it transmutes a unit's (neuron) activation level into an output signal (Ukil, 2007; Zilouchian & Jamshidi, 2000). Figure 2.6 depicts an example of a neuron.

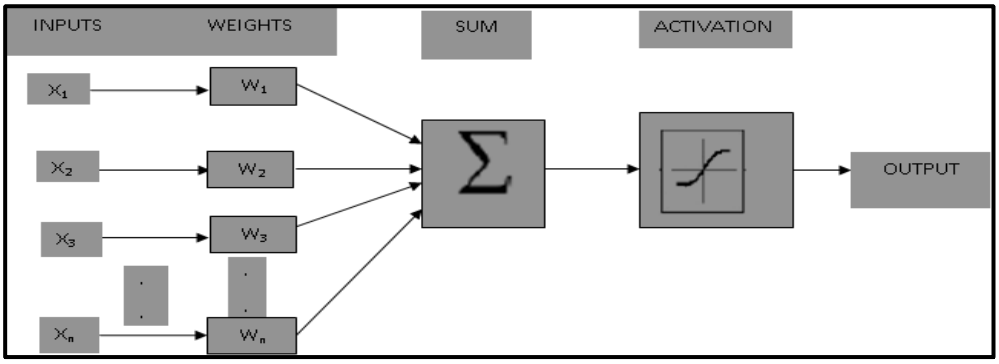


Figure 2.6. Example of a Neuron

Based on Figure 2.6, the sum and output can be computed as:

$$S_j = \sum_{i=1}^{n} X_i W_i \qquad (2.2)$$

$$O_j = T_j(S_j) \qquad (2.3)$$

Where,

j: Neuron j,

i: Index of the inputs,

n: Number of the inputs,

X_i: Input i,

W_i: Weight of the input Xi,

S_j: Sum of the weighted inputs for neuron j,

$T_j(S)$: Transfer function, and

O_j: Output of neuron j

2.2.2.1 Basic NNs Structure

NNs stem from the interconnections of multiple unit neurons or nodes. Typically, as a rule, the neuron layers mentioned below are utilised in the artificial networks (Ukil, 2007; Vancoillie, 2003).

- Input layer: The number of neurons corresponds with the number of inputs to the neuronal network. This layer comprises passive nodes—the ones that do not take part in the actual process of signal modification—and simply transfers the signal to the next layer.

- Hidden layer: Here there are a random number of layers along with random number of neurons. The nodes take part in the process of signal modification, and therefore are deemed active.
- Output layer: In this case, the number of neurons corresponds with the NN's output values, and the nodes are active.

Figure 2.7 depicts the basic structure of the artificial NN.

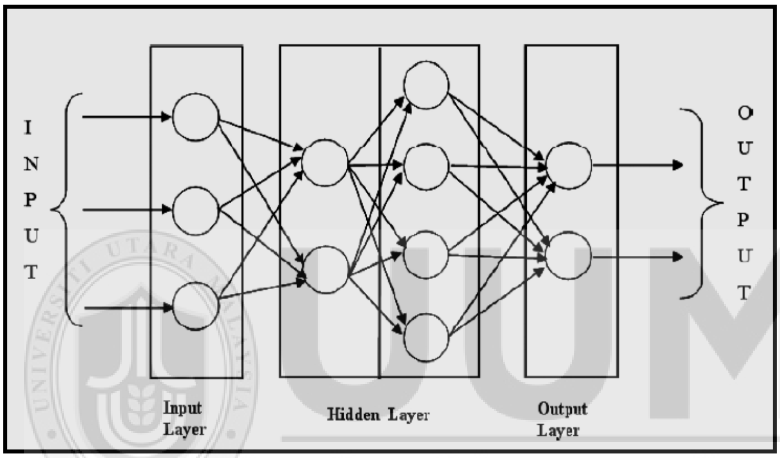


Figure 2.7. Basic structure of the Artificial NN

2.2.2.2 Learning Algorithm

Training an NN to obtain the targeted result entails choosing a model which reduces a criterion cost, learning from its environs and bettering performance (Amakali, 2008; Zilouchian & Jamshidi, 2000). Presented below are few of the learning algorithms:

- **Supervised Learning**

Here, the NNs absorb knowledge from examples. A set of inputs is entered into the neural network and the outcome is compared with the intended output (i.e. the outputs conforming to the inputs). The network parameters are adjusted one by one based on the error signal acquired from the difference between the outputs and the intended output.

- **Unsupervised Learning**

In this case, a measure that is 'task-independent' is utilised to distinguish, which is an intended or unnecessary network representation. The parameters are adjusted in accordance with this measure. In the absence of an external teacher, the system should systematise itself by the internal norms and local information that is designed into the network. Unsupervised learning is also called as self-organising learning; in other words, learning to categorise without being trained. Here, just the input samples are available and the network categorises the input patterns into various groups.

- **Backpropagation Algorithm**

This algorithm, created by David Rumelhart, Geoff Hinton, and R. J. Williams in 1986 (Rumelhart, Hinton, & Williams, 1988), is among the renowned algorithms for computing the weights of the NN.

O_j stands for the output values of a given input pattern X_i. These values do not always correspond with the predetermined values R_j. E_j is the error resulting from the difference of R_j and O_j. This error is minimised by the changes in weight. This is achieved by propagating the error originating at the output backwards towards the hidden layers. In other words, the error is considered and back-calculated to determine the correct solution, which is the principle behind the backpropagation algorithm. That is, error is propagated in a backward direction.

The error of a neuron j in the output layer is given below when the energy analogy is considered:

$$E_j = \frac{1}{2}(\mathrm{Rj} \; - \; \mathrm{Oj}\,)^2 \tag{2.4}$$

E is the total error of the output layer and is given by

$$E = \sum_j E_j = \frac{1}{2}(\mathrm{Rj} \; - \; \mathrm{Oj}\,)^2 \tag{2.5}$$

The error E has to be minimised while accounting for the changes in weight (Rezaee, Goedhart, Lelieveldt, & Reiber). Through the delta rule, the learning rate α is incorporated, and the weight change method is selected based on the gradient descent algorithm (Alsmadi, Omar, & Noah, 2009).

2.2.3 NFS s

A NFS combines techniques from FS and NNs. However, this does not mean that both are used in any actual combination. Rather, a fuzzy system is augmented by a NN to improve few of its characteristics such as speed, flexibility, and adaptableness. A NFS is devised to accomplish the process of fuzzy reasoning, wherein the connection weights of the network embody the parameters of fuzzy reasoning (Abraham, 2005; DD Nauck & Nurnberger, 2005)

2.2.3.1 Comparison between FSs and NNs

There are noteworthy similaritiess between NNs and FSs. From the mathematical perspective, both are dynamic, parallel processing systems which estimate input-output functions. Furthermore, both are both universal approximators which can assess a function devoid of any mathematical model and draw learnings from experience with sample data. From a practical perspective, both attempt to model expert behaviour while addressing complex issues.

Table 2.2

Comparison between FSs and NNs

	Skills	FSs	NNs
Knowledge acquisition	Inputs	Human experts	Sample sets
	Tools	Interaction	Algorithms
Uncertainty	Information	Quantitative/Qualitative	Quantitative
	Cognition	Decision making	Perception
Reasoning	Mechanism	Heuristic search	Parallel computation
	Speed	Low	High
Adaptation	Fault-tolerant	Low	Very high
	Learning	Induction	Adjusting weights
Neural Language	Implementation	Explicit	Implicit
	Flexibility	High	Low

Yet, the two tactics show some extreme differences as well. Table 2.2 shows in brief these differences from various angles (Abraham, 2004; Grosan & Abraham, 2011).

NNs and fuzzy systems can be combined to capitalise on their benefits and overpower their respective drawbacks. Neural fuzzy systems are typically utilised in function approximation and control issues, particularly in the engineering arena where classical approaches do not supply straightforward and precise solutions.

2.2.3.2 Adaptive Network based Fuzzy Inference System

ANFIS is a simple data learning technique which utilises a FIS model to transform a crisp input into a target output by creating MFs for every input.

The fundamental notion behind such Neuro-adaptive learning methods is to offer a technique for the fuzzy modeling procedure to study information about a data set, so as to automatically calculate the MF parameters which best enable the associated FIS to trace the given input/output data.

This method enjoys an advantage over the pure fuzzy paradigm, as the need for a human operator to adjust the system by adjusting the MFs' bounds is no longer there.

This this chose ANFIS as a control tactic due to its uncomplicated structure and the benefits it offers of fuzzy logic as well as adaptive NNs. Adaptive NNs enjoy the benefit of having the ability to learn and adjust to the system.

The ANFIS is able to sharpen up the fuzzy 'if-then' rules and MFs so as to outline the input/output behaviour of an intricate system. In practical uses, the Takagi-Sugeno type of FISs (Gorrostieta & Pedraza, 2006) are deemed more apt for building fuzzy models owing to their more compacted and computationally-effective representation of data compared to the Mamdani fuzzy systems. A usual zero-order Takagi-Sugeno fuzzy system has the following form:

If x is A and y is B then z = c

where A and B are fuzzy sets and z is a crisply defined function. To cater to the requirements of a particular problem, a singleton spike is most of the time totally adequate . A zero-order Takagi-Sugeno FIS is utilised in this study.

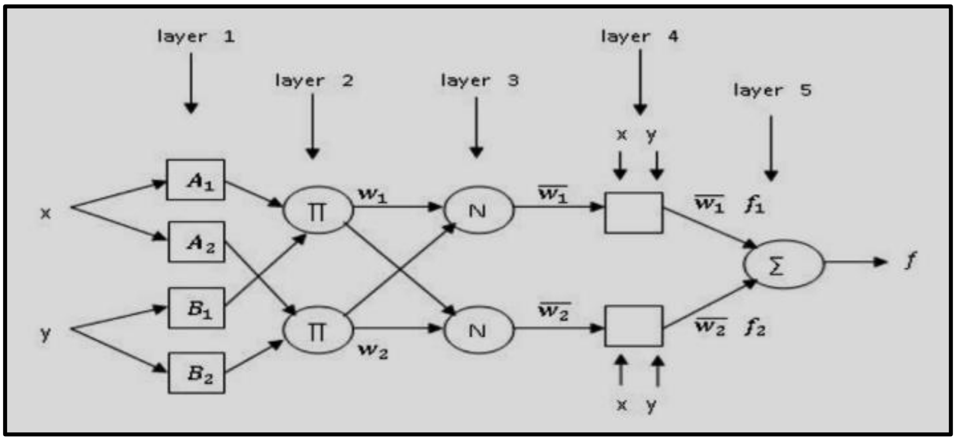


Figure 2.8. ANFIS Takagi-Sugeno fuzzy system

As can be seen in Figure 2.8, ANFIS is made up of five layers. The first concealed layer is accountable for the mapping of the input variable fairly to every MF. The operator T-norm is used in the second concealed layer to compute the antecedents of the rules. The third concealed layer normalises the rules’ strengths, which is followed by the next concealed layer in which the rules’ consequents are ascertained. The output layer computes the global output as the aggregate of all signals which arrive to this layer.

2.2.4 Genetic Algorithms

These are based on the notions of the evolution theory (Holland, 1975), with the key principle being: only the fittest objects prevail (Reid, 2000; Sivanandam & Deepa, 2007). A genetic algorithm can be split into many sub-parts which are utilised in this algorithm: fitness function evaluation, representation, selection, initialisation, recombination (crossover and mutation), and termination. Figure 2.9 outlines the entire process of a genetic algorithm.

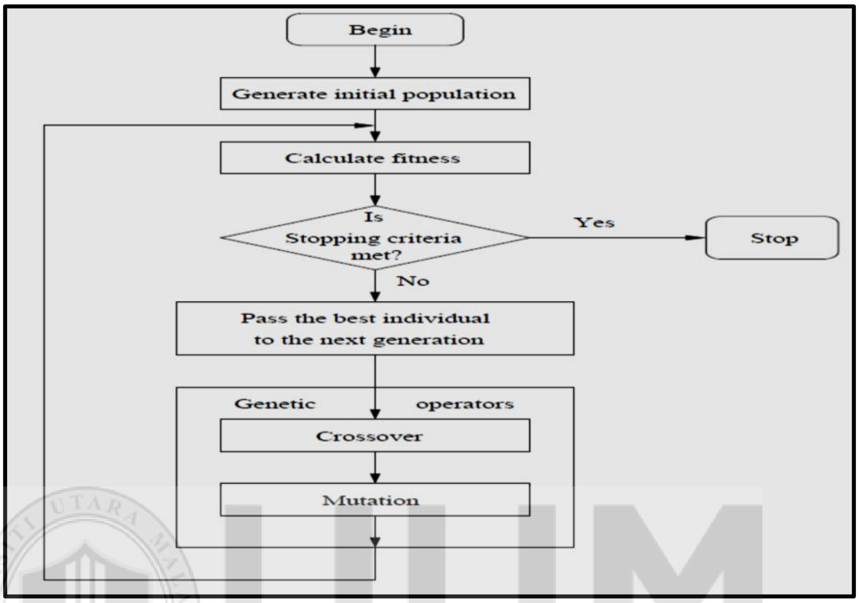

Begin
Generate initial population
Calculate fitness
Is Stopping criteria met?
Yes
Stop
No
Pass the best individual to the next generation
Genetic operators
Crossover
Mutation

Figure 2.9. GA algorithm flowchart

2.2.4.1 Representation

In this study, real coded strings, which employ a real number are used because they are suitable for optimisation problems in the continuous domain. In real-coded GAs, the chromosome is a vector of floating point numbers, $C_i = (x^i_1,..., x^i_n)$, the size *n* of which corresponds to the dimension of the continuous search space, such that each gene x^i_j corresponds to a specific problem parameter.

2.2.4.2 Population and Initialisation

In case of initialisation, the initial chromosomes set, also known as the initial population, is formed. The initial population's size is crucial for the overall genetic algorithm. A small size could help find only a local optimum, whereas a larger size offers a greater prospect that the global optimum will be located; notably, the computation time rises (Reid, 2000).

2.2.4.3 Evaluation and Selection for Reproduction

The selection operator is utilised to determine chromosomes that would be used in reproduction and would live on until the next generation. Various methods can be deployed in selection operators. Typically though, a natural selection process is simulated, wherein individuals who are the "strongest" are used in reproduction. T*ournament selection* is used in this study.

2.2.4.4 Recombination

Recombination operators are a key component of the genetic algorithm. The crossover operator mimics the reproduction between two individuals, wherein the offsprings produced inherit certain traits from parent individuals. There are several mutation and crossover operators which function with a chromosome encoded as a literal line of numbers or symbols.

Here is a list of some common crossovers:

- Ordered crossover,
- Cycle crossover,
- Single-point crossover,
- Uniform crossover,
- Two-point crossover

The above mentioned crossovers generate an encoded chromosome or chromosomes as a consequence, which have to be decoded for the purpose of assessment.

Binary string: 0 1 1 0 1 1 0 0

1st parent: 1 2 3 4 5 6 7 8

2nd parent: 3 5 1 8 4 7 2 6

Intermediate offsprings

1st offspring: – 5 1 – 4 7 – –

2nd offspring: 1 – – 4 – – 7 8

Generated offsprings:

1st offspring 2 5 1 3 4 7 6 8

2nd offspring 1 3 5 4 2 6 7 8

Figure 2.10. Uniform order crossover

The uniform crossover operator is perhaps the strongest of its kind as it enables the offspring chromosomes to look for all prospects of re-combining those various genes in parents. Uniform crossover has emerged as the most commonly utilised crossover operator (Picek & Golub, 2010).

Figure 2.10 demonstrates a uniform order crossover.

A mutation operator is utilised so as to avoid getting trapped in the local optimum and raise the possibility of locating the global optimum (Hong, Wang, Lin, & Lee, 2002). In case of the mutation operator, a new offspring is generated from a single solution. This is accomplished by altering few intrinsic characteristics. For a genetic algorithm, crossover and mutation operators are deployed by a predefined probability.

2.2.4.5 Termination

GAs are stochastic techniques which can go on endlessly if there is no termination criterion. Maximum calculation time, maximum iteration number, or iterations which are counted from the last successful enhancement of the best individual constitute the simple stopping criteria (Hong et al., 2002; Reid, 2000). The prospect to improve the best individual reduces proportionately to the calculation time. Thus, the number of iterations without improvement is directly proportional to the prospect of improvement. A bigger value will raise the computation time and a better solution might be discovered. Likewise, a low value will cause an early stop and a substandard solution.

2.2.4.6 Genetic Algorithm for Rule Extraction

A particular kind of optimisation problem is the knowledge discovery process from databases, also called as rule extraction (RE). Rule mining is deemed as a usable mining method technique of mining so as to gain valuable knowledge from data stored in database systems.

The process aims to mine a minimal set of simple rules which classifies (covers) with the best probable examples of accuracy in the particular database (dataset). There are many global search algorithms in the evolutionary algorithms (EA) paradigm that effectually categorise huge datasets without carrying out an exhaustive search. Few examples of application of EA for classification and optimisation problems are as follows: geometric programing (Boyd, Kim, Vandenberghe, & Hassibi, 2007), GA (Holland, 1975; Knowles & Corne, 2000), evolution strategy (ES) (Coello, 2005), and evolutionary programming (EP) (Fogel & Fogel, 1996). This paper emphasises on genetic algorithm for feature selection as well as rule extraction in data mining.

The key drive behind using GA in the detection of high-level prediction rules is to carry out a global search and deal with attribute interaction in a better manner.

This study principally aims that the function of rule extraction is carried out automatically. It is evident that the automated knowledge discovery process enjoys certain benefits over the conventional kind of analysis carried out by humans. These are as follows:

- More adaptable and dynamic to changes in parameter and attribute
- Reduced time to solution (classification) period
- Cost-effective and more steady performers compared to human classifiers
- Better precision in solutions (classifications)
- Multi-purpose
- Hardly any proficiency required to carry out analysis

In the domain of evolutionary algorithms, there are two key approaches for developing these kind of rule systems: the Pittsburgh approach and the Michigan approach (Michalewicz, 2013). The iterative rule learning (IRL) approach is a method that has been recommended recently, especially for fuzzy modeling (Triantaphyllou & Felici, 2006).

The above mentioned approaches are described below (Peña-Reyes, 2004).

- **The Pittsburgh approach.** The evolutionary algorithm keeps a populace of candidate fuzzy systems, with every individual standing for an entire fuzzy system. Genetic and selection operators are utilised to generate new generations of fuzzy systems.

  The problem of credit assignment is avoided as evaluation is applicable to the total system. This approach allows for inclusion of additional optimisation criteria within the fitness function, and hence multi-objective optimisation can be implemented. The main shortcoming of this approach is its computational cost, since a population of full-fledged fuzzy systems has to be evaluated each generation.
- **The Michigan approach:** Every individual is a representative of a single rule. The FIS is a representative of the total populace. As many rules partake in the inference process, these rules are continually competing for the best action to be recommended, and liaise to create an effective fuzzy system. The method's cooperative-competitive nature makes it tough to determine which rules are eventually responsible for good system conduct. This calls for an effectual credit-assignment policy to attribute individual rules with fitness values.

- **The IRL approach:** One of the main disadvantages of the above two approaches is the using up of considerable computer memory for scouting various fuzzy rules. To tackle this issue, the IRL approach was developed. Here as well, every individual encodes a single rule. An evolutionary algorithm is deployed to determine a single rule, therefore offering a partial solution. Then the evolutionary algorithm is deployed iteratively for discovering new rules till a proper RB is constructed. A penalisation scheme is used every time there is an addition of a new rule. This is to stop the process from determining rules that are redundent – in other words, rules involving the same input subspace. This approach gels the Pittsburgh approach's simplicity of fitness evaluation with the pace of the Michigan approach. This paper has taken up this learning approach.

The IRL offers several benefits:

1. IRL curtails the search space. This is because in every sequence of iterations, the learning method simply looks for a single best rule rather than the entire RB.
2. The Michigan approach offers online learning for non-inductive learning concerns, while IRL aids in offline inductive learning issues.

2.2.4.7 Multi-objective Genetic Algorithms

Multi-objective designs are realistic models for several intricate engineering optimisation issues. In several real-life issues, objectives under consideration clash against each other. Optimising a specific solution with regards to a single objective could lead to undesirable outcomes in terms of other objectives. A rational resolution to a multi-objective issue is to study a set of solutions, with each of it fulfilling the goals at a satisfactory level without being dominated by any other resolution. GA are a common meta-heuristic especially compatible with these kind of problems. Traditional GA are tailored to accommodate multi-objective issues by utilising specialised fitness functions and launching methods to endorse solution diversity.

2.2.5 GFS

FSs can be automatically designed, and this approach can be considered as an optimisation or search process on the potential solution space. GAs are the more popular and widely used global search methods for their capability to explore and exploit a given operating space with the use of available performance measures. A priori knowledge, which include linguistic variables, fuzzy rules, fuzzy MF parameters, and number of rules, could be easily integrated into the genetic design process. GAs have a generic code structure and independent performance features that are suitable for a priori knowledge incorporation. These features have expanded the use of GAs in developing of a wide range of methods for fuzzy system design.

Genetic fuzzy inference systems are the most common kind of GFSs. In this case, GA are utilised for optimising the various constituents of the FIS.

2.2.5.1 Taxonomy of GFSs

The first thing to do while devising a GFS is to make a decision as to which parts of the fuzzy systems need to be optimised. The next step is coding them into chromosomes which need be optimised by genetic algorithm.

GFS approaches are split into two groups: tuning and learning. The first crucial factor for making a choice between them is whether an initial KB exists or not. By taking into consideration database and RB, the GFS framework is outlined in brief as follows:

- *Genetic tuning:* In case there is a KB, genetic tuning would be deployed for system upgrade such that the RB stays the same; however, the FRBS parameters would be adjusted.
- *Genetic learning:* This is one more complex method of optimising only the RB or the entire KB. It does not require predefined rules. Furthermore, the RB would be developed during genetic learning (Cordón, Herrera, Gomide, Hoffman, & Magdalena, 2004).

Given below Figure 2.11, is the taxonomy as per the stated points (Herrera, 2008).

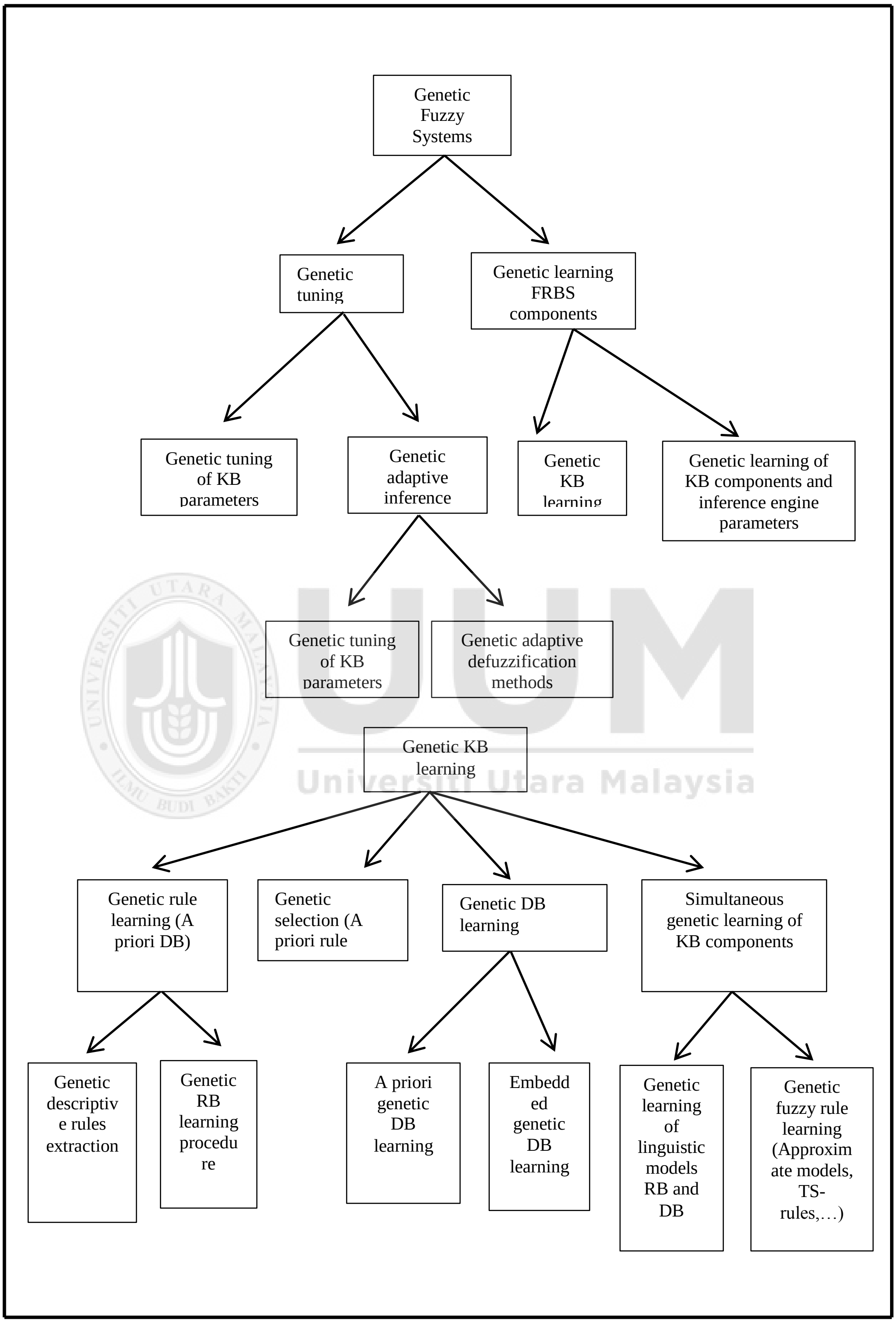

Genetic Fuzzy Systems
Genetic tuning
Genetic learning FRBS components
Genetic tuning of KB parameters
Genetic adaptive inference
Genetic KB learning
Genetic learning of KB components and inference engine parameters
Genetic tuning of KB parameters
Genetic adaptive defuzzification methods
Genetic KB learning
Genetic rule learning (A priori DB)
Genetic selection (A priori rule
Genetic DB learning
Simultaneous genetic learning of KB components
Genetic descriptive rules extraction
Genetic RB learning procedure
A priori genetic DB learning
Embedded genetic DB learning
Genetic learning of linguistic models RB and DB
Genetic fuzzy rule learning (Approximate models, TS-rules,…)

Figure 2.11. GFS taxonomy

2.2.5.2 Genetic Tuning

After the design of the RB, a number of approaches can be used for the effective implementation of FRBS. To achieve this, the initial database definition or the inference engine parameters are redefined.

Tuning can be conducted in three ways on the basis of sub-tree under genetic tuning:

1. *Genetic tuning of* KB *parameters.* This approach is employed to modify MF parameters. The learning process merely modifies the shape of the MFs, but not the length of chromosomes (Casillas, Cordón, Del Jesus, & Herrera, 2005).

2. *Genetic adaptive inference systems.* This approach applies parameterised expressions to inference systems called Adaptive Inference Systems to enhance more cooperation in fuzzy rules and to design more accurate fuzzy models (Alcalá-Fdez, Herrera, Márquez, & Peregrín, 2007).

3. *Genetic adaptive defuzzification.* Defuzzification interfaces are important components of a fuzzy interference system. This approach uses a GA for optimising the defuzzification unit of fuzzy inference RB systems.

2.2.5.3 Genetic KB Learning

Another field in GFSs is genetic learning. It is much more complex in comparison to genetic tuning. The method can comprise learning of only the RB or learning of the entire KB (Cordón, Herrera, Hoffmann, & Magdalena, 2001; Mumford, 2009). Here are the four approaches outlined for genetic learning:

1. *Genetic rule learning:* The majority of the methodologies that have recommended to automatically learn about the KB from numerical infomation

emphasise on RB learning by deploying a predefined database (Del Jesus et al., 2007; Herrera, 2008). Figure 2.12 depicts this in a clear manner.

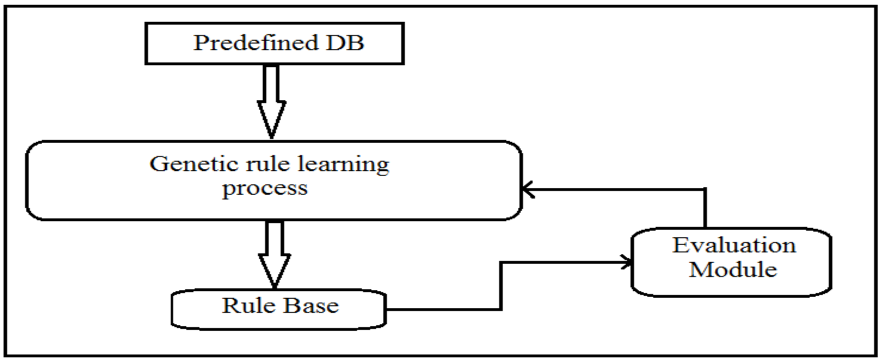


Figure 2.12. Genetic rule learning process

2. *Genetic rule selection:* Data mining methods, on some occasions, create a large number of rules, making it tough to comprehend the conduct of FISs. Few of the rules are redundant, unrelated, erroneous and conflictive. To prevent such a scenario, one can deploy a genetic rule selection process so as to secure an optimised abstract of rules (Alcalá et al., 2007; Casillas et al., 2005). Figure 2.13 below explains this.

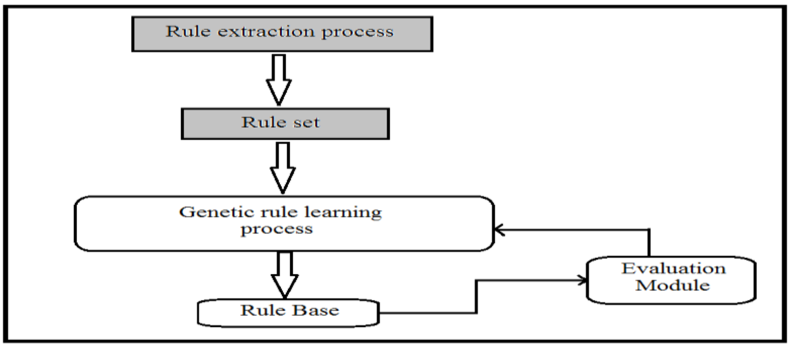


Figure 2.13. Genetic rule selection process

Furthermore, rule selection can be blended with tuning methods, which aids in obtaining a robust rule set supplemented with a tuned suite of parameters (Herrera, 2008).

3. *Genetic database learning*: This approach optimises the entire KB in two steps, such that every time a database is mined by the process of database definition, the rules will be extracted by deploying the RB generation process (Cordon et al., 2001; Herrera, 2008). Then the entire KB is substantiated. These conceptions are depicted in Figure 2.14.

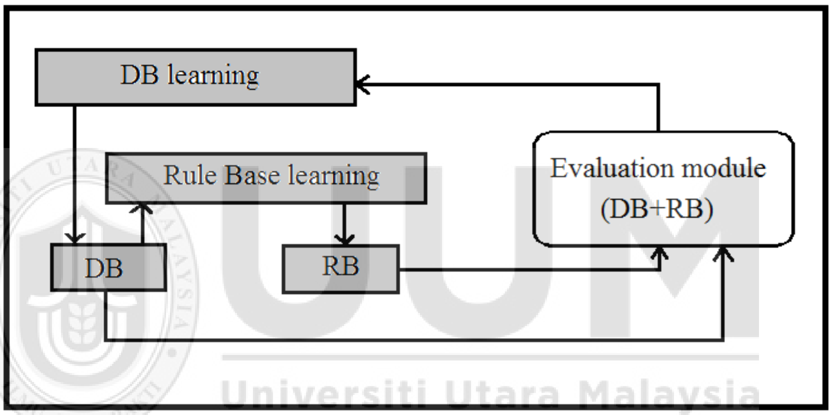


Figure 2.14. Genetic database learning

4. *Simultaneous genetic learning of* KB *components:* As the name suggests, this approach attempts to learn the two constituents of KB simultaneously. The benefit is that better definitions are generated, while the drawback is that it entails tackling a larger search space, which might make the learning progression sluggish and complex (Fazzolari, Alcala, Nojima, Ishibuchi, & Herrera, 2013; Herrera, 2008).This is shown in Figure 2.15.

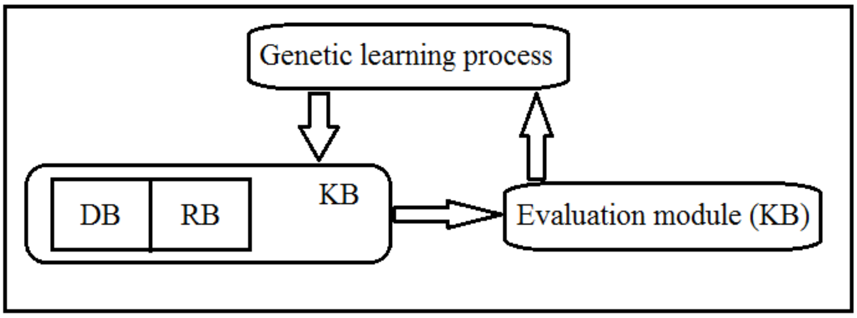


Figure 2.15. Genetic KB Learning

2.2.5.4 Multi-Objective Evolutionary Fuzzy Systems (MOEFS)

An instinctive approach to FRBS design is to extract the knowledge from a human expert and to organise this knowledge in the DB (Ishida, Pozo, Goldbarg, & Goldbarg, 2009) and rule base (RB) and of the FRBS. In few of the application areas, this knowledge could be limited. This could be because of that domain complexity, hence rendering this natural approach unfeasible. To get past this issue, there have been certain methods recommended in prior studies so as to create the DB and RB by mining this knowledge from available information, such as experimental samples. Initially, thus type of generation was usually carried out with the distinct goal of boosting the precision; however, soon the academics came to know that these precision-driven approaches usually generate FRBSs characterised by a great number of rules and by linguistic fuzzy partitions having a low comprehensibility level, hence squandering that feature that has made FRBSs the preferred choice compared to other methodologies in real applications, i.e. interpretability. To tackle this issue, during the last ten years, new approaches have been recommended to generate FRBSs by considering precision as well as interpretability of rule and databases and other criteria of assessment. Therefore, the creation of FRBSs entails resolving a multi-objective optimisation problem.

In this framework, multi-objective evolutionary algorithms (MOEAs) have been utilised. MOEFSs has been created to determine the hybridisation of FRBSs with MOEAs, which are used to obtain FRBSs with different trade-offs between different evaluation criteria. This is achieved by learning the overall RB or by choosing subsets of rules from heuristically determined initial RBs, as well as by learning the overall DB or by tuning a DB put forward by the experts. This study uses MOEAs to choose a subset of rules according to different evaluation criteria.

2.2.6 Probabilistic Reasoning

Taking a theoretical perspective, the classical probability and Fuzzy set theories are complemented to develop a hybrid formalism to model the complex uncertainty that characterise real-world systems.

Conventionally, uncertainty in the real life is addressed mainly by the probability theory. However, in real-world applications, randomness is just one of the several sources of uncertainty. Ambiguity is another key source. Ambiguity directly arises from the concept of set membership. In this instance, membership is represented by a continuous function that can take the form of any real number on the closed interval [0,1] rather than 0 or 1 alone. Fuzzy set theory uses this highly generalizable idea of membership to model the ambiguity of real-world systems. The new idea is that set membership is crucial to decision making in the presence of uncertainty.

2.2.6.1 Bayesian Statistics

Bayesian inference is a statistical inference approach that uses Bayes' rule to update a probability estimate for a hypothesis as additional evidence is acquired. Bayesian inference is applied to a wide scope of fields, including science, medicine and engineering. This approach calculates the probabilities according to Bayes' rule:

$$P(x|z) = \frac{P(z|x).P(x)}{P(z)} \tag{2.6}$$

Where,

P(x\z): called posterior.

P(x): called prior, belief before seeing the evidence.

P(z\x): likelihood of seeing evidence if the hypotheses is correct.

P(z): hypotheses of evidence under a considering status.

Bayesian theorem proposes a simple process to identify the hypothesis with the maximum probability without relying on search methods.

Prior probabilities for categories and attributed values based on categories are estimated derived from frequency counts calculated from the training data.

2.2.6.2 Mutual Information

MI is defined as a measure of statistical dependence characterised by two main properties. The first property is the capability to measure any type of relationship, including nonlinear relationships, between random variables (Cover & Thomas, 2012). The second is that MI remains invariant despite invertible and differentiable feature space transformations, such as rotations, translations or any other transformation that preserves the order of the feature vectors' original elements.

The statistical dependence between two variables can be detected by using information theoretic techniques, which are thus intuitively applied in the design of fuzzy partitions. The replacement of a continuous variable with a linguistic variable results in information loss. In this context, the best discretisation will be one that maximises the MI between input and output.

MI is commonly employed to measure the relevance between features and decisions. However, the relevance between continuous or fuzzy features is difficult to calculate using MI. To calculate the relevance between numerical or fuzzy features and decisions, fuzzy information entropy and fuzzy MI are adopted.

MI can be alternatively defined in terms of entropies and conditional entropies as follows:

Let X be a random variable having a finite set of M_X possible states X_i with i ε $\{1,\ldots,M_X\}$ and with a probability distribution P_X , and Shannon entropy of X. $H(X)$ or $H(P_X)$ is expressed as follows:

$$H(X) = -\sum_{i=0}^{Mx} p(Xi)\log(p(Xi)) \tag{2.7}$$

where $p(X_i)$ refers to the probability of the state X_i, whereas the Shannon entropy measures how evenly the states of X are distributed.

The conditional entropy $H(X|Y)$ stands for the uncertainty of X given Y and is expressed as follows:

$$H(X|Yj) \quad = - \sum_{i=1}^{Mx} p(Xi|Yj) \log\big(p(Xi|Yj)\big) \tag{2.8}$$

$$H(X|Y) \quad = - \sum_{j=1}^{Mx} p(Yj) H(X|Yj) \tag{2.9}$$

Thus, the MI *I(X;Y)* can be given by:

$$I(X;Y) = H(X) - H(X|Y), \tag{2.10}$$

or

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{2.11}$$

$$I(X;Y) = -\sum\sum p(X,Y) \log \frac{p(X,Y)}{p(X)p(Y)} \tag{2.12}$$

2.3 Air/Fuel Ratio

The new trend is to develop control strategies in the engine that operate within highly accurate values. This situation will facilitate a decrease in fuel composition and emissions, thereby maintaining optimum engine performance.

In electronic fuel injection engines, when the fuel enters the vehicle engine (internal combustion engine) it will be burned, so there is a need for air in the combustion chamber. This procedure happens in the fuel injection system such that mixes fuel with air in an internal combustion engine with a certain amount. There is a ratio must be adhered to between the air-fuel mixture called the perfect or stoichiometric ratio (Kamat, Javaherian, Diwanji, Smith, & Madhavan, 2006).

In this regard, such factors as engine speed, spark timing, fuel injection timing, engine torque, air intake and AFR must be controlled (Ebrahimi et al., 2012). Among these factors, the AFR has the most important influence on fuel effectiveness and emission reduction.

To compose an ideal engine control system, it is necessary to achieve an accurate prediction of the AFR. The AFR must be controlled in both the transient and steady engine operation states. This can easily be achieved in steady-state operation because fuel dynamics and air-induction are eliminated at the steady state. By contrast, the value is hard to control in transient state when the engine is in the throttle position because engine speed and intake air pressure instantaneously change (Garcia, 2013).

Accurate control of the AFR requires predetermination of the physical effects that can result in AFR transitions. Following the determination, such effects can be mitigated by constructing a control strategy (Ghaffari A. & others, 2008).

The AFR ratio is plagued with various engine-related factors that may influence its value. Some of these factors include spark timing, throttle position (TPS), engine torque, fuel injection timing, air intake, engine speed (RPM), manifold air pressure (MAP) and manifold air temperature (MAT).

The AFR can also be expressed with lambda (λ), which for a certain mixture may imply the ratio between the actual AFR and the stoichiometric AFR (Howlett, Zoysa, Walters, 2000). For a stoichiometry mixture, the value of λ is 1.0. Similarly, for rich mixtures, the value is less than 1.0 and for lean mixtures, it is greater than 1.0. The computation of λ involves dividing the AFR by the stoichiometric AFR for a specific fuel (different fuels have different stoichiometric AFR), which can be expressed as follows:

$$\lambda = \frac{AFR}{AFR_{stoich}} \qquad (2.13)$$

14.7:1 is considered the perfect ratio (based on the amount of hydrogen and carbon present in a given quantity of fuel, as perfect ratios vary with different fuels).

A rich mixture is obtained when the quantity of air is less than the perfect ratio, while a lean mixture is obtained when the quantity of air is more than the perfect ratio.

A rich mixture contains fuel that is left over after combustion, which results in:

- Increase in the car’s power.
- Diminution in the fuel economy.
- Increment in CO and hydrocarbon emissions in the air.
- Overheating of the catalytic converter, a device that reduces the toxic emissions from internal combustion engines. Overheating may sometimes result in melting of the catalyst, which may result in complete or partial blockage inside it. This could result in a considerable decline in the performance on highways or hindering the motion due to the back pressure created in the exhaust system.

A lean mixture contains redundant oxygen that may result in the following:

- Increase in the fuel efficiency;
- Increment in the emissions of nitrogen oxide;
- Engine damage;
- Decrease in the performance.

The air–fuel mixture is regulated by the oxygen sensor (O2 sensor) by sending the feedback to an electronic control system (ECU) to control the air–fuel ratio.

Three main parts are included in the electronic control system (Thomas & Sharma, 2007; Fleming, 2008):

(a) Sensors that can measure and control the main characteristics. These sensors allow converting physical characteristics into electric signals.

b) The measured data is processed by an electronic control unit (ECU) that employs a microprocessor. Integrated computer programs backed by a database can be used to compute several major characteristics. This database contains crucial information for operating the engine.

c) The output electric signals are translated into mechanical characteristics by actuators.

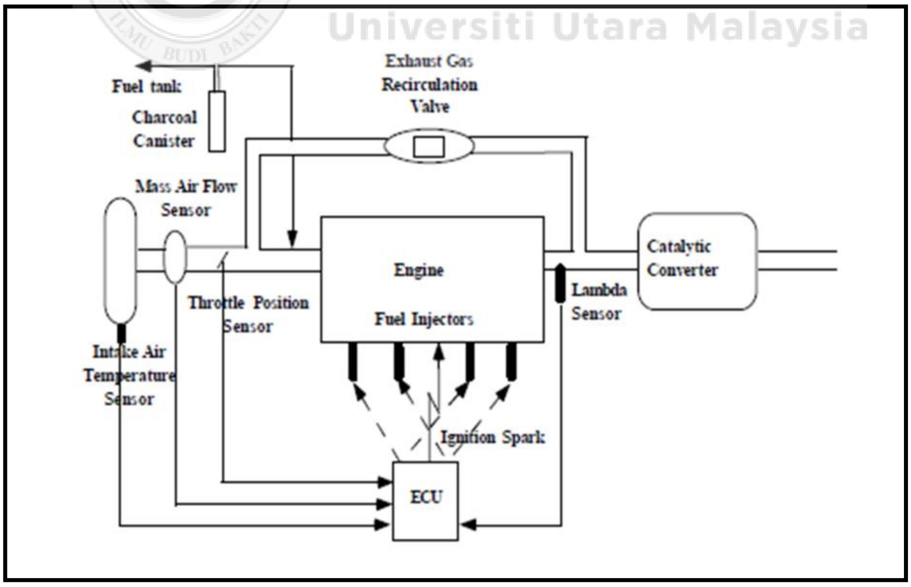


Figure 2.16. Engine Emission Control System

As presented in Figure 2.16, control strategies are employed by the ECU to gain high output power and optimum efficiency based on requirements, while simultaneously maintaining low emission levels. In a gasoline or spark-ignition engine, the engine must be operated by the ECU in a favourable region that allows functioning of a three-way catalytic converter. This helps in reducing the harmful content of the exhaust further.

The ECU employs three-dimensional mappings (3-D maps) in the form of look-up tables to signify non-linear conduct of the engine in real time. A major disadvantage of this method is the need to achieve optimal engine operation in the required time for determining the values. This process is also referred as the ECU calibration (Lee, Howlett & Walters, 2004). Optimised functions and data tables stored in the ECU help in calibrating the engine. The calibration process aims at determining the best settings suitable for an engine as well as ensuring that the process can be executed as quickly as possible with the minimum cost. However, the process is iterative and time consuming as it involves multiple cycles of engine measurements. For both tuning and calibration, the process of building the table is difficult. Techniques that can reduce both the effort and time of the whole calibration process are the need of the hour.

In the electronic control system, the O2 sensor forms a part of the many sensors. Figure 2.17 shows a zirconium ceramic bulb, which acts as a sensing element that is coated with a thin layer of platinum on both sides. The bulb’s outer casing is exposed to hot exhaust gases while the inside portion is connected to the ambient air or passed through the sensor.

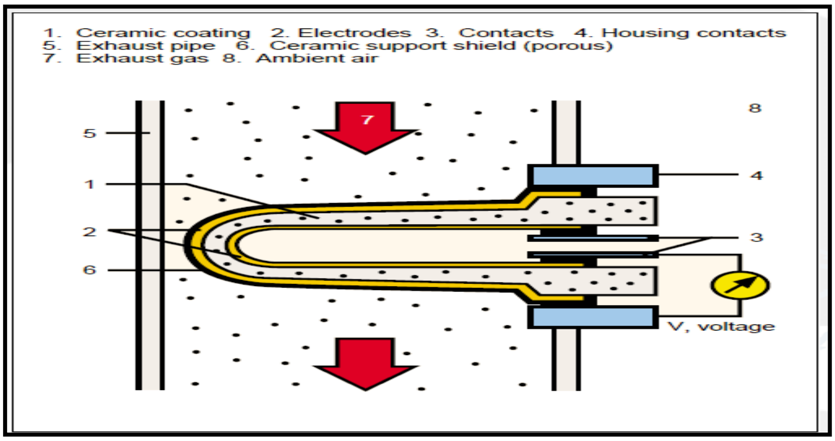

1. Ceramic coating 2. Electrodes 3. Contacts 4. Housing contacts
5. Exhaust pipe 6. Ceramic support shield (porous)
7. Exhaust gas 8. Ambient air
5
1
2
6
7
8
4
3
V, voltage

Figure 2.17. *Oxygen sensor*

As presented in Figure 2.18, a voltage is produced by the O2 sensor on detecting a difference in the oxygen levels. Typically, the voltage range lies from 0.8 to 0.9 volts when the air–fuel is rich and the exhaust contains little oxygen. However, when the air–fuel mixture is lean and the exhaust contains large quantities of oxygen, the output voltage of the sensor drops to 0.1–0.3 volts. For a perfectly balanced and regulated air–fuel mixture, the output voltage of the sensor is around 0.45 volts (Cesario, Lavorgna & Pirozzi, 2005), as presented in Figure 2.19.

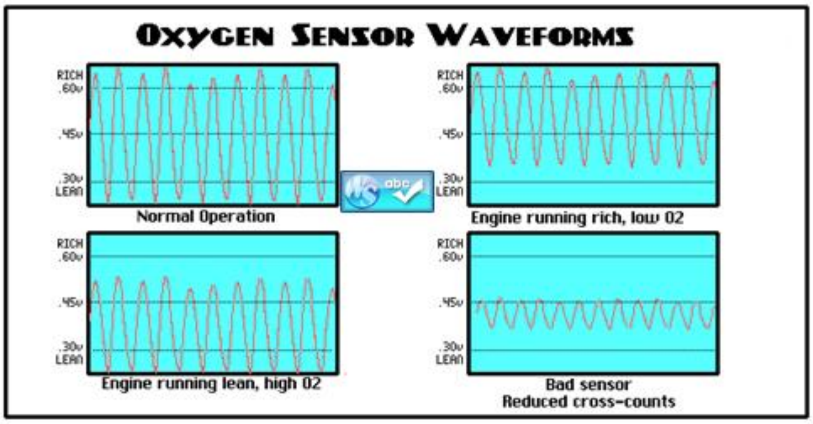


Figure 2.18. Oxygen sensor waveforms

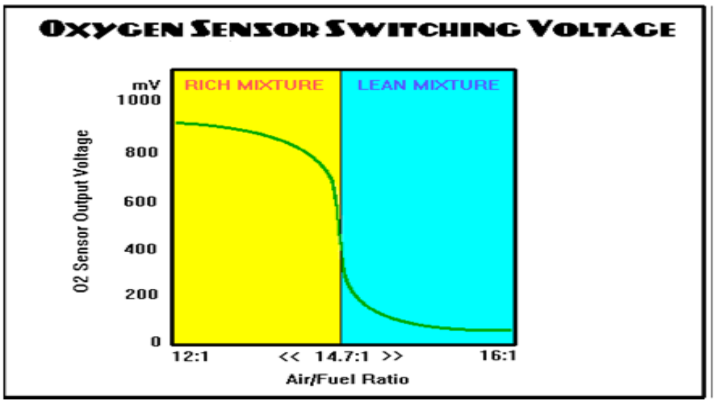


Figure 2.19. Oxygen sensor switching voltage

As illustrated in Figure 2.20, the O2 sensor is a part of the emissions control system and is responsible for feeding the data in the form of a voltage to the ECU. As the O2 sensor and the ECU work in a consentient manner, it is also known as a closed loop. The O2 sensor's goal is to help engine to run efficiently in the best possible manner by emitting minimum emissions (Howlett, Zoysa, & Walters, 2000). The sensor is mounted on the exhaust pipe to identify rich and lean mixture. The sensor operates in a mechanism where a chemical reaction produces a voltage to adjust the fuel quantity entering the engine. This, however, is based on the lean or rich characteristics of the mixture.

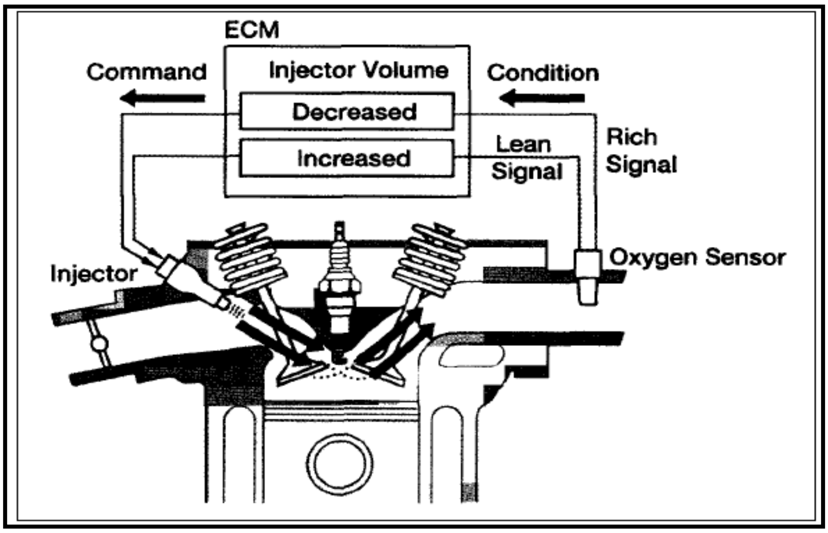


Figure 2.20. Fuel Injection System

The engine needs an O2 sensor as its capacity to pull in oxygen, which is the relative leanness or richness of the fuel mixture, relies on a lot of factors, including engine coolant temperature, air temperature, throttle position, barometric pressure, engine load, airflow, etc. Also, another major reason would be the inability of the ECU to measure and compensate for every possible factor.

Apart from this, the O2 sensor itself has its own problems. One such problem is the longer time taken to transport exhaust gases to the O2 sensor, which leads to delay in transporting the voltage to the ECU as well. Rapid fluctuation in the air–fuel ratio is caused by this extended time delay between the measurement of the O2 sensor and the controlling on the ECU’s action. This also has an impact on the air–fuel ratio control loop’s performance, on top of the time processing of the ECU, as shown in Figure 2.21.

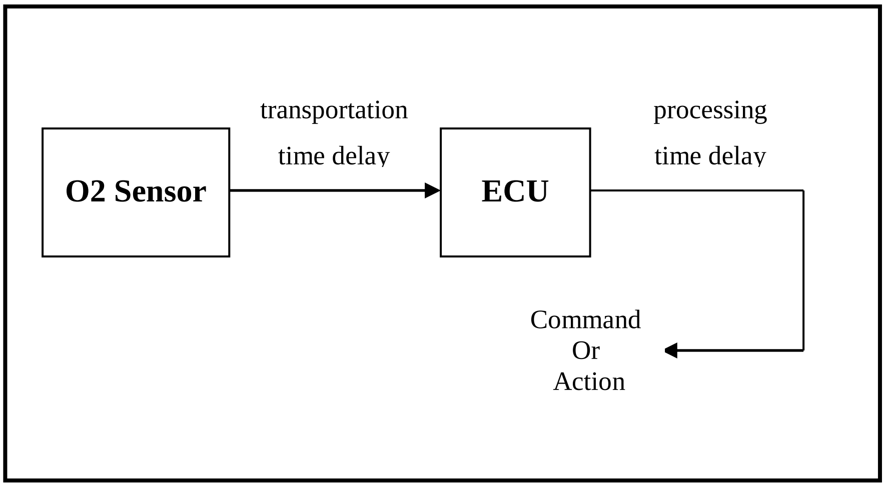


Figure 2.21. Time Delays

Another issue would be the O2 sensor becoming less active because of the pollutants from oil ash as well as normal combustion. Some of the other problems that may arise are:

1. The formation of deposits on the element sensing portion of the sensor, which leads to decrease in the ability of the sensor to function and response quickly to changes in the fuel or air mixture. Thus, gradually, the sensor’s functions start to diminish and behave as a laggard.
2. If the output voltage of the sensor is not high enough as was previously, the system may perceive that the air–fuel mixture is leaner than it is actually. This results in the computer giving the signal to send in more fuel to the engine, which can lead to a richer air–fuel mixture. Should the O2 sensors fail, the computer keeps guessing and may result in utilisation of more fuel than the required amount, which will affect the fuel economy. In such a condition, there is no data transfer to the ECU from the O2 sensor and it ends in guessing the air–fuel ratio, which is known as the open loop.

The O2 sensor issue can be dealt with by developing a virtual sensor. The virtual sensor data can be utilised for monitoring performance and making smart decisions pertaining to engine control. The key benefit of utilising a virtual sensor is that it offers dependable estimated measurements that were not hitherto available. Furthermore, virtual sensors do not consume much space or payload in comparison to the hardware which would be required to offer equivalent measurements. Moreover, their maintenance is less arduous compared to that for hardware. The virtual sensors can be constructed using artificial intelligence techniques for developing intelligent engine control prototypes, capitalising on the used methods to overpower the sensor issues and aiding the ECU in its processing for executing control actions.

2.3.1 Literature Review for AFR

Numerous studies have used classical and modern methods to assess AFR control in both the steady and transient engine operation states.

Richter, Oliveira, and da Silva (2010) studied the use artificial NNs to predict oxygen sensor values. They employed the multi-layer perceptron (MLP) neural network that used 42 factors as training data for the network. These factors include torque, air flow rate, engine speed, and oxygen sensor. The whole universe of data and their subsets were used to train over 2,300 topologies of MLP neural networks. Through this process, almost 350 out of 792 topologies achieved a mean relative error of less than 1%. With subset training, over 600 topologies achieved a mean relative error of less than 0.5%. The same datasets were used to train all topologies, and the same data were used for all tests.

Ghaffari et al. (2008) presented a control method using adaptive fuzzy control for the AFR of a vehicle spark-ignition engine. A proportional integral-derivative (PID) tuning method that uses an adaptive fuzzy system designed to determine the correlation between controller gains and target output response was also discussed. The fuzzy-PID controller employed a Mamdani-type FIS.

Cao et al. (2006) proposed a gas heating furnace intelligent control algorithm to achieve the optimum control of the A/F ratio. This algorithm was derived from the fuzzy NN. The project reveals that the control algorithm has high accuracy and reliability, rapid inference, satisfactory tracking, strong disturbance, feasibility, and usability, as well as suitability for project application compared with classical algorithms such as the PID.

Liu and Zhou (2010) likewise proposed a fuzzy NN to control the AFR. This network employs the backpropagation learning algorithm and defines the target function as the sum of square errors between the stoichiometric and actual AFRs, which were simulated under transient conditions. Simulation results showed that when the throttle degree was considerably changed without the presence of a controller, the AFR errors were greater than 10.9%. Meanwhile, with the use of the fuzzy NN controller, the errors were controlled to a narrow range and had values greater than 1.8%. The system also has a shorter adjustment time.

NNs are highly suitable for real-time systems because of their immediate response and rapid computational times attributable to their parallel architecture. However, despite these advantages, a number of disadvantages can be noted, particularly in terms of their dependence on data-intensive training algorithms and the difficulty in integrating available discrete knowledge. Owing to this condition, local culmination easily occurs. An important disadvantage is the difficulty in explaining NN behaviour owing to the representation of distributed knowledge. The knowledge can only be represented by revealing the responses to a certain input. In addition, MLP networks can be considered as a type of black-box model that may be difficult to debug.

Previous researchers failed to extract the rules from the NFS data. Such system exhibited advantages in dealing with real-world problems, particularly nonlinear problems. A drawback is its dependence on experts.

2.4 Learning Algorithms for NFSs

A fuzzy system designer specifies the distinguished points of an assumed underlying function by encoding such points as simple fuzzy rules. Except for these characteristic points, the function is unknown. The fuzzy sets employed to describe such points linguistically reveal the degree of indistinguishability of the points that are in close proximity to one another. Fuzzy rules can be conveniently derived from the data using different learning algorithms for the components of the function with unknown characteristic points but with available training data.

A fuzzy system has a structure that is based on its rules and data space granularity, which refers to the number of fuzzy sets used to partition each variable. Fuzzy system parameters refer to the MFs' shapes and locations. FS also have the benefit of enabling the use of expert knowledge to create a RB. However, expert knowledge is only partially available or unavailable in most applications. In such cases, a RB that performs well on a given data set must be created from scratch.

2.4.1 Cluster-Oriented Fuzzy Rule Learning

Cluster-oriented methods attempt to group training data into clusters that are used for rule creation. Soft (fuzzy) clustering is a clustering algorithm and a powerful unsupervised method for data analysis and model construction. FCM algorithm is one of the most widely used approached. Joe Dunn first developed FCM in 1973, and Jim Bezdek further improved it in 1981 (Bezdek, 1981; Dunn, 1973). This clustering method uses fuzzy partitioning to enable a data point to belong to all groups at different membership grades ranging from 0 to 1. The pseudocode of FCM is as in Figure 2.22.

```
Algorithm 2.1: Fuzzy c-means
    1. Initialise U={$u_g$} matrix, $U^{(0)}$
    2. At k-step: calculate the centres of vectors $C^{(k)} = [c_j]$
       with $U^{(k)}$
    3. Update $U^{(k)}$ , $U^{(k+1)}$
    4.
       $d_{ij} = \sqrt{\sum_{i=1}^{n}(xi - cj)}$   …….. (2.4)

       $u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{dij}{dkj}\right)^{2/(m-1)}}$   ……….(2.5)

    5. If || $U^{(k + 1)}$ – U(k) || < ε then STOP; otherwise return
       to stop 2.

    end-algorithm
```

Figure 2.22. Pseudocode for FCM

where *m* is any real number greater than 1, *uij* refers to the degree of membership of *xi* in the cluster *j*, *xi* is the *i*th of d-dimensional measured data, *cj* is the d-dimension centre of the cluster, and d_{ij} is the objective function.

This algorithm assigns membership to each data point that corresponds to each cluster centre according to the distance between the cluster centre and the data point. When the data is nearer the cluster centre, its membership degree is higher towards this particular cluster centre. Obviously, the summation of membership of each data point should be equal to one. Membership and cluster centres must be updated according to the formula following each iteration.

The subtractive clustering algorithm is another fuzzy cluster algorithm that identifies cluster centres through subtractive clustering (Chiu, 1994).

FCM can not identify the number of clusters or fuzzy sets for each data point. Subtractive clustering helps to identify the number of clusters.

2.4.2 Structure-Oriented Fuzzy Rule Learning

Structure-oriented approaches can be considered as special cases of hyperbox approaches that do not search for clusters in the data space but rather choose hyperboxes from a grid structure. The data space is structured by overlapping hyperboxes by providing (initial) fuzzy sets for each variable. This method of learning fuzzy rules was suggested by Wang and Mendel (L.-X. Wang & Mendel, 1992). Multi-dimensional fuzzy sets with hyperbox as support are overlapped to partition the feature space. Rules are created by selecting those hyperboxes that contain data.

In the process of partitioning the universe of discourse towards becoming grid-like sections, the primary disadvantage is related to the number of fuzzy rules. Juang, Lin, and Lin (2000) used GA and noted that all chromosome length grows exponentially as the fuzzy variables' dimension increases.

In 1996, (Nozaki, Ishibuchi, & Tanaka) came up with a proposal of using a simple grid-based fuzzy system of classification. This approach makes automatic adjustment to the fuzzy rules' grades of certainty; however, the dimensionality cruise cannot be ignored.

Riza, Bergmeir, Herrera, and Benítez Sánchez (2015) used a grid-type fuzzy partition for determining the antecedents parts of the fuzzy rules.

2.4.3 Evolutionary Fuzzy Rule Learning

A number of studies propose the use of automatic methods for fuzzy rule learning with different algorithms, such as GA, along with different classifiers. GAs are mainly used in discovering high-level prediction rules to perform a global search and better address attribute interaction than greedy rule induction algorithms frequently used for data mining. This section provides a brief overview on a few representative works that contributed to the application of GA or other algorithms to data mining and machine learning fuzzy classification rule creation.

Feature selection and rule extraction approaches are required by a rule-based fuzzy generation system. Feature selection is developed by some models as a pre-processing approach, whereas other models apply both approaches simultaneously.

2.4.3.1 Feature Selection

The problem of feature selection can easily be positioned as an optimisation problem where the objective is to choose a subset of features based on some optimised feature-subset assessment criterion. Consequently, genetic and other evolutionary algorithms have been extensively utilised to solve the problem of feature selection. The evolutionary algorithms for feature selection methods usually follow a wrapper approach wherein a subset of features is encrypted in a chromosome and a feature assessment criterion is employed as the fitness function. The feature subsets are analysed based on the efficiency with which the selected features classify (for the supervised case) or cluster (for the unsupervised case) the dataset. However, assessment of chosen features based on a single criterion is not equally applicable for all datasets. Thus, the requirement of simultaneously optimising multiple such criteria came into being. The robustness of the feature selection techniques is also enhanced with the use of multi-objective feature selection.

Included in the fuzzy researchers' concern is the NNs approach, with MLP being its most popular method. Features are frequently chosen through the evaluation of the link's weights between the features and the hidden layer. If the weight's absolute values for a number of features are near zero, it can then be taken away from the feature set. Mitra and Hayashi (2000) provides an example for this method. A good number of these methods involve certain kinds of regularisation. Oftentimes, a three-layer NN is employed for feature selection: when a penalty term is added to the error function, relevant connections are made distinguishable from redundant ones with small weights.

Since proposals have been made in a number of researches, particularly by Ichihashi, Shirai, Nagasaka, and Miyoshi (1996), Rudas and Kaynak (1998), and H.-M. Lee, Chen, Chen, and Jou (2001), the concept of utilising entropy measures in guiding rule extraction and feature selection is not really new. In 2001, Lee et al. attempted to simplify fuzzy rules using an explicit feature elimination step. They employed a measure of fuzzy entropy along with the backward elimination method in order to select useful features. Stopping the feature elimination process is dependent on the classifier error rate. Because this is an incremental process, possible nonlinear interaction, which may be present between features, may not be accounted for.

Here, the process begins with a fuzzy rule-based system. The authors then introduce a measure of information within a fuzzy model, using it towards the quantification of the available information in the fuzzy model. This is done both before and after a specific input feature is utilised for classification. The information gain related to this input variable is signified by the difference between these two information measures. Along the associated input feature, this is considered as a discriminability measure between output classes. Silipo & Berthold's approach is towards the simplification of a fuzzy rule-based system; this is only applicable after the identification of a fuzzy system.

MI for fuzzy random variables was defined in a study by Sánchez, Suárez, Villar, and Couso (2008). Here, the entropy is employed in estimating the MI. Conventional feature selection approaches that use MI focus on the dependence between variables, which is computed before they are fuzzified. These authors maintain, however, that the shape of the MFs defined on them may influence the dependence. Thus, they provided a definition to MI for fuzzy random variables, which they utilised for fuzzy rules feature selection.

For Battiti (1994), the feature reduction problem is defined as the process of choosing, from an initial set of *n* features, the most relevant ones. He proposed a greedy selection method as a solution. The problem can be solved, ideally, by maximising $I(C;S)$, which is the joint MI between the class variable C and S which is the subset of selected features. Computing Shannon's MI between high-dimensional vectors, however, has been proven to be impractical since the required number of samples, as well as the central processing unit (CPU) time is prohibitive. In order to prevail over these limitations, Battiti adopted a heuristic criterion to approximate the ideal solution. In the computation, he only used $I(C;f_i)$ and $I(f_i;f_j)$ where f_i and f_j are individual features, instead of calculating the joint MI between the selected feature set and the class variable. His mutual information feature selector (MIFS) chooses the feature which maximises the information about the class, correcting this through the subtraction of a quantity that is proportional to the average MI with the features that were previously selected.

A unifying framework for methods of information-theoretic feature selection was proposed by Brown, Pocock, Zhao, and Luján (2012). Posing the problem of feature selection as a conditional likelihood of the class labels, the authors provided several features. With the filter assumption, conditional likelihood is considered as an equivalent to conditional MI (CMI). Formulating the feature selection task as a conditional likelihood problem, it is referred to as a feature scoring criterion: 'MIM', meaning *Mutual Information Maximisation*. The model likelihood is utilised as the loss function, which therefore, seeks to maximise the likelihood. The likelihood is iteratively maximised through the selection (or removal) of features greedily. The feature that maximises or minimises the conditional MI between that feature and the class label can then be chosen. This is conditioned upon the chosen feature set along with the prior term for the current feature set. In the iterative updates, it must be noted that the prior term is a ratio; thus, if an uninformative prior is employed, the prior term gets cancelled. This brings about a maximum likelihood update rule based entirely on the conditional MI. Examples of each type of feature selection methods are the Relief and the SSGA Integer k-nearest neighbour classifier (KNN) method.

The Relief method is a filter method which employs the KNN algorithm and the information gain to choose the feature subset, according to Kira and Rendell (1992). Notwithstanding the target concept to be learned, this method requires linear time in terms of the number of given features and training instances. It is not necessary that the algorithm should find the smallest subset of features since the size has the tendency to be small due to the fact that only statistically relevant features are chosen. The Relief method only applies to the problem of two-class classification; it does not deal with problems on continuous value prediction. Insufficient training instances pose a problem for this method.

The SSGA Integer knn method, according to Casillas, Cordón, Del Jesus, and Herrera (2001), is a wrapper method that employs a filter feature selection method, followed by a wrapper feature selection method in order to obtain a fuzzy rule based classifier. This wrapper utilises a genetic algorithm in order to generate a feature subset that is then evaluated by means of a knn classifier. A wrapper can likewise be utilised as a filter, in any case, as demonstrated by Ravi and Zimmermann (2000). Their study provides a predefined number of features. An optimisation algorithm is employed in searching for the combination of features which gives the best classification error rate. A distance measure, which gives a certainty assessment with which an object is assigned to a class, sorts two subsets of features with the same classification error rate.

C. K. Zhang and Hu (2005) put forward the proposal of a new scheme of feature selection for ANNs via genetic algorithm utilising MI. Here, MI between input and output is used in mutation in GA so as to guide the evolutionary search direction purposively based on certain criteria that can make the search process faster and get better performance. In its examination of the forecasting at the Australian Bureau of Meteorology, the study noted that the simulation of three different feature selection methods demonstrates that the proposed method can lead to the reduction of the dimensionality of inputs, a speeding up of the network training and the achievement of a better performance.

The feature selection can be classified to: the class-independent and the class-dependent techniques.

In class-independent feature selection, chosen features are common to all classes, while in class-dependent feature selection, different feature sets are chosen for different classes. In class-independent feature selection, there is an assumption that all features chosen equally discriminate each class from the others. This hides the following possibility: different groups of features might possess different capabilities in discriminating classes.

This thesis is based on the selection of a subset of features that can distinguish one class from another. This aspect of features that enables class discrimination forms the basis of class dependent feature selection techniques.

Bailey (2001) defined class-dependent features as *features whose differentiation ability varies considerably based on the classes that are required to be discriminated.*

A method for class-dependent features selection and construction of a novel RBF classifier was described by L. Wang and Fu (2006) based on class-dependent features. Diverse feature subsets are chosen as inputs for diverse groups of obscure units corresponding to diverse classes in RBFN. For all classes, the ideal feature masks are searched using GAs. When selecting class-dependent features only a single RBF network is needed for a multi-class problem, instead of multiple MLPs.

Although there are different methods discussed in the study in relation to feature selection, it is important to note that feature selection results greatly vary depending on the classification algorithm that is adopted as the wrapper. In addition to this, the final selection of feature subset is significantly impacted by the type and number of objective functions.

2.4.3.2 Rule Extraction

In relation to fuzzy rule extraction, there are many articles available on dimensionality reduction or feature exclusion (that is the objective of feature selection). There are multiple manners to extract useful rules from a training dataset (X, Y). There are two key methods to fuzzy rule extraction. A static partition of the input space is used to create fuzzy rules in one family of methods, while clustering is used in the other family of methods to arrive at useful rules.

The partition-based approach is used by a number of papers (Alcala-Fdez, Alcala, & Herrera, 2011; Alcalá, Gacto, & Herrera, 2011; Gacto et al., 2010; Ishibuchi & Nojima, 2007; Ishibuchi & Yamamoto, 2004; Ishibuchi, Yamamoto, & Nakashima, 2005; Mansoori, Zolghadri, & Katebi, 2008), where a search algorithm is generally utilised to derive the necessary rule set. In these methods, there is an exponential growth of the search space with respect to the dimensionality of the dataset and evolutionary algorithms are usually employed for selection of rules. A number of these methods intrinsically accomplish feature selection at the rule level. To state it differently, a specific feature may not be used by a rule, and hence a feature that is not used by any rules, can be dropped. The application of fixed fuzzy partition has made it simple to interpret the rules.

On the other hand, clustering-based methods cluster the dataset and then render each cluster into a fuzzy rule (Lughofer & Kindermann, 2010; Pal & Saha, 2008).

From the observational data, Setnes and Roubos (2000) assessed the GA-fuzzy modeling. In this model, fuzzy clustering is employed to derive a primary condensed rule-based model. This model is then augmented by a real-coded GA, considering the constraints that support the semantic properties of the rules. A great number of work based on fuzzy clustering algorithms has been attempted to initialise the fuzzy model, where the objective was to decrease the intricacy of the fuzzy model and attain a compact and interpretable fuzzy classification technique.

Pomares, Rojas, Ortega, Gonzalez, and Prieto (2000) a plan for the self-generation of a fuzzy RB was proposed. Initially, in this scheme, the authors begin by using a simple rule-based system. Afterwards, through an analysis of the approximation error, the system's structure automatically undergoes modification. Along with the rule consequents, the location and the number of MFs on each input variable are made optimal by this method. However, the problem of feature selection is not addressed by this approach.

Sarkar, Sana, and Chaudhuri (2012) proposed a GA-based rule extraction system to enhance prediction accuracy for any classification problem regardless of domain, size, dimensionality, and class distribution. An accuracy-based learning system called decision tree and genetic algorithm (DTGA), which improves prediction accuracy, is introduced. In particular, the proposed system has two rule-inducing phases. In the first phase, a base classifier called C4.5 (a decision tree-based rule inducer) is employed to generate rules from the training data set, whereas the GA in the next phase refines these rules to provide more accurate and high-performance prediction rules.

Detlef Nauck and Kruse (1997) put forward a learning method for fuzzy classification rules based on NEFCLASS or Neuro-Fuzzy Classification. NEFCLASS is employed towards deriving fuzzy classification rules and learning the MFs' shape based on a data set that can be segregated in different crisp classes. Gliwa and Byrski (2011), however, note that the NEFCLASS model's orientation is towards interpretability, not accuracy. As NFSs for building rule-based FS is adversely affected by conflicting rules, these have to be eliminated through a little post-processing after the initial system's extraction. Another primary disadvantage with such a system applies even in the case of moderately high-dimensional data: the very rapid growth of the network size renders such a Neuro fuzzy system less useful when applied to many practical problems. Thus, Neuro-fuzzy models, as it relates to fuzzy rules, are only effective for tuning or choosing rules, not at extracting or constructing them.

It was previously stated that creating a rule-based fuzzy system is a classification problem that needs approaches to select the consequent class in the fuzzy rule which best fits the rule's antecedent. Probabilistic classification, a common subclass of algorithm classification, utilises statistical inference in finding the best class for a given instance. Other algorithms simply output a "best" class, while probabilistic algorithms generate all the probabilities of an instance to be a member of each of the possible classes. The one with the highest probability is then chosen as the best class. While such an algorithm possesses an edge over non-probabilistic classifiers, the latter can be more effectual when it is incorporated into larger machine learning tasks, partially or entirely avoiding the problem of error propagation.

V. Gopalakrishnan, Lustgarten, Visweswaran, and Cooper (2010) used the Bayesian network (BN) approach to rule learning—evaluating the rules using Bayesian scores and translating them into rules. A rule model is defined as a set of rules that, taken together, make up a classifier which is applicable to new data in order to predict the target. This model's primary contribution lies in its ability to quantify uncertainty about the validity of a rule model with the use of a Bayesian score for model selection. The data enables this algorithm to learn BN models, which then gets translated into a set of rules complete with associated statistics. Since these rules are mutually exclusive and exhaustive when it comes to the values of the predictor variables, therefore, inference that uses these sets of rules becomes trivial. The rule that matches the values for the predictor variables, in the case of a new test case, is employed in inferring the target variable’s value. The research of V. Gopalakrishnan, et al have demonstrated that, in generating rule models, the use of a BN approach not only produces a more parsimonious model when it comes to the number of variables; it also produces outputs that are, statistically, significantly superior to the usual rule learning methods. Probabilistic rules which are optimised on the rule model level are also created in lieu of the current evaluation per rule method. As a generating model, BN presents a coherent method that incorporates different types of prior information and updates the rule model.

Tang and Sun (2004) provided a gradient-based method for the identification of a new fuzzy classification system from observation data. In such a fuzzy classifier, which can be described as a mixture model of classical fuzzy classifiers, each fuzzy rule stands for a probability distribution of class labels (i.e. a special belief structure). This fuzzy classifier is made up of rules with each one representing more than one class consisting of different conditional probabilities.

Lv, Zhu, and Tang (2007) likewise presented a similar gradient-based method with practically the same objectives as well as rule characteristics, constitution, and representation. Identifying the proposed fuzzy classifier involve working out a gradient-based method through the optimisation of an objective function that is based on the posterior probabilities of classes with certain predictive features. An alternating optimisation process is undertaken in the proposed identification method. The training data and parameters of the fuzzy sets in the antecedents are used to calculate the class conditional probabilities that are associated with each fuzzy rule. Then, from the class conditional probabilities and the training data, a computation of the gradient of the fuzzy set's parameters in the fuzzy model is made.

A model using Bayesian reliability estimation under fuzzy environments was proposed by Wu (2004). Invoking Resolution Identity in the fuzzy set theory, it created the fuzzy Bayes point estimator of reliability. It is hard to get a fitting prior probability in many practical situations. Because of this, it was assumed by most researches that prior probability requires a specific kind of distribution, using complicated testing and advanced statistical techniques in estimating the selected distribution's parameters. Wu's paper considered non-repairable multi-state element with this assumption: the failure rates are fuzzy triangular fuzzy function. Towards developing posterior fuzzy reliability with fuzzy probabilities, the traditional Bayesian formula is used in Bayesian inference with fuzzy probabilities. Lastly, the evaluation of the posterior fuzzy reliability is done with the use of Genetic Algorithm (GA), an optimisation technique. The illustrative example section primarily shows the results from experiments where the likelihood probability is set as constant; inappropriate posterior probability may result from this.

A presentation involving a Fuzzy Bayesian Classifier (FBC) over LR-type fuzzy numbers with an unknown conditional probability density function was presented by Sadoghi Yazdi and Vahedian Mazloum (2009). A new version of the KNN method was employed in estimating the likelihood density function.

Saxena and Pratap (2012) presented a Genetic Algorithm Based Bayesian Classification Algorithm. Depending on the conditional probability counted from the frequency of the events. The proposed GA based classifier on the basis of learning data has predicted the grade of given test data. The algorithm attempted to minimize the response time and improve the classification accuracy.

In the context of fuzzy rule extraction, feature selection is sometimes done in a separate phase. On the other hand, some methods factor in the learning machines, but usually removing, stepwise, one feature (or a set of features) at a time. There is a possibility, therefore, that the chosen features might not be the best set for the current problem since, one, a set of features can have interactions between themselves, and two, a feature can have an interaction with the tool which utilised to address the problem. Moreover, methods of feature selection frequently ignore the subtle nonlinear interaction which can occur between the features and the learning system. On top of the additional time for learning, the search for relevant features' subset introduces another layer of complexity in the task of modeling.

2.4.3.3 Feature Selection and Rule Extraction Together

Towards addressing this problem, an integrated method is utilised in finding the features simultaneously during its search for the rules in the data for FS. The nonlinear interactions possibly present between the features and the fuzzy rule-based systems can be considered in this integrated learning mechanism. Therefore, a small set of useful features can be picked up by this mechanism; it could also lead towards generating useful rules for the current problem. Since it is not iterative in nature, unlike the forward or backward selection approaches, this integrated approach is computationally very attractive. Described below are some researches that applied the rule extraction for FS and the feature selection simultaneously.

Gonzalez and Perez (2001) used a MI criterion as well as a GA for feature selection and fuzzy rule induction. SLAVE, their learning algorithm, chooses features for each particular rule. Consequently, each rule may obtain a different subset of variables in identifying the class. A genetic learning algorithm, SLAVE utilises the iterative approach to generate an FRB. Chromosomes normally represent individual rules in this iterative approach; a single rule is chosen at each GA iteration. The model's RB is formed by the set of selected rules. With SLAVE's embedded feature selection process, the pre-selection of attributes minimises the problems brought on by large search spaces (i.e. excessive execution time), while it improves the generated models' interpretability. Reducing the original problem of getting an entire set of rules into a simpler problem consisting of obtaining only one rule at a time: this is the primary concept behind SLAVE. A single rule is represented by each chromosome of the population in this approach. However, in each iteration, only the best individual is considered, while the remaining chromosomes are being discarded. The feature selection process that is adopted by SLAVE explores the set of possible variables dynamically so as to find the rule that is most useful, as well as to locate the most relevant variables for this rule. Therefore, the implementation of this feature selection process is for each single rule and not for the complete set of rules. The process' basic schema consists of the modification of the rule representation in the search mechanism of SLAVE. This is to allow the learning algorithm to look for, not only for the best rule, but also for the best set of variables for every rule. The definition of the best rule here is the one which has the highest degrees of consistency and completeness simultaneously, while combining both factors with the use of a product operator. SLAVE likewise generates rules with different weights.

Jin (2000) initially identified an initial RB with the use of a training data set that is consequently simplified based on a fuzzy similarity measure. The fuzzy system's structure and parameters are then optimised using GAs and gradient-based methods. This process results in a more interpretable fuzzy system, in which different rules involve different numbers of input variables. This approach may fail to eliminate any input variable unless such variable is not involved in any rule.

Another proposal involves an approach to extract rough set rule from a decision system with the use of conditional information entropy presented by Zhao and Sun (2012). This paper presents a new perspective on conditional information entropy. It employs a heuristic approach for rule extraction that is based on conditional information entropy in mining classification rules. With a feature selection approach, removing redundant features without any classification information loss is first undertaken to reduce a decision system. After this, with the use of the rule extraction approach, classification rules are mined from the reduced decision system. Zhao and Sun's paper initially proposed an efficient feature significance measure of information entropy towards improving the classification accuracy of rule extraction algorithm from the decision system. Here the feature selection and rule extraction are at two levels.

An integrated mechanism to simultaneously extract fuzzy rules and to select useful features was presented by Chen, Pal, and Chung (2012). The method employs a single-pass gradient-based search technique as well as a clustering approach.

The research work of Rawat and Burse (2013) involves the design of a framework incorporating genetic algorithm with Neuro-fuzzy for the purpose of feature selection and classification on the training dataset. This research's objective is towards the simultaneous optimisation of the parameters and feature subset without degrading the accuracy of the ANFIS classification**.** The feature selection serves as a pre-processing for the ANFIS classifier, which was evaluated in terms of the training performance and classification accuracy. Accuracy is also the reason why the wrapper approach in featuring subset selection is utilised in this paper. The evaluation of the overall fitness function is done by getting the weighted sum of the match score of all of the correct recognitions and the incorrect recognitions, and then subtracting the latter from the former. The experts provided the fuzzy rules.

Chakraborty and Pal (2004) made a proposal about a new scheme to design fuzzy rule based classifier in a Neuro-fuzzy framework. The system is considered novel because the network can choose good features alongside relevant rules in an integrated manner. The initial step consists of the network learning about the relevant features and the classification rules. Later, the network is pruned in order to obtain the best architecture representing an optimal set of rules. This is done by removing redundant nodes, as well as incompatible and not-used rules. The final step involves tuning the final rules' MFs for improved performance. This therefore results in a smaller architecture of the final network; this will now have a lower running time compared to the initial network.

Another integrated system was proposed in 2008, this time by N. R. Pal & Saha. This system, which can choose the useful features as it designs the RB for approximating function in the TS framework, uses a feature modulator that is a monotonic differentiable function with the range [0,1]. Eliminating poor/bad features which can be unpredictable and problematic during system identification is the main objective in proposing this system. Towards its realisation, membership values are modulated: setting it to a high negative value if the feature is bad; to a high positive value, if the feature is considered important for the discrimination task.

In 2003, (Y. Wang, Chang, Wang, & Li) presented a method for input selection that is based on fuzzy clustering, as well as for rule generation that is based on genetic algorithm (GA). This was set in an ANFIS that is used to model protein secondary structure prediction. The model utilised a two-phase process. During the first phase, choosing the number and position of the fuzzy sets of the initial input variables is done by using a fuzzy clustering algorithm, During the second phase, an iterative GA updating algorithm is utilised to achieve the ANFIS rule-base’s more accurate structural identification and optimal parameters.

2.4.3.4 Multi-objective evolutionary algorithm for Fuzzy Rule Learning

The aim of the previous mentioned works is to decrease in the number of fuzzy rules and/or the number of features but this mostly creates a negative effect on the exactness of the RB, affecting the interpretability of the system. A typical rule creation process where two objectives are contradictory in nature, involves a multi-objective optimisation approach where there is maximisation of accuracy and minimisation of complexity with reduction in the number of rules and/or reduction in the total length of the RB. Classification accuracy is based on how good the rules are. Thus, many attempts have been made to produce rules that best fit the data. For instance, Abadeh, Habibi, and Lucas (2007) count the number of patterns that fall within the coverage area of the rules and use this measure as a fitness function. In (2002), (Carvalho & Freitas) introduced a hybrid system to identify and address rules that cover a small number of training examples by using a decision-tree/algorithm approach. It was highlighted by the authors that if the extraction algorithm rule negates the sets of small number of training examples, the classification accuracy is significantly degraded.

According to (Gliwa & Byrski) in 2011, decrease in the number of features sometimes improves the accurateness of classification, since a vast number of features can override the learning data and escape the simplification. Also Antonelli, Ducange, and Marcelloni (2012) outlined accuracy and the total number of attributes (conditions) as the two objectives.

There are numerous approaches to sort out the issue of accuracy-complexity/interpretability in the evolutionary algorithm domain.

Ishibuchi et al. (2001) proposed a plan for coherent rule creation through a three-objective optimization approach. In this approach, it was attempted to increase the classifier performance, and decrease the number of fuzzy rules and number of antecedent clauses in the antecedent of the fuzzy rules. A modified multi-objective genetic algorithm (MOGA) and a hybrid fuzzy genetic-based machine learning (GBML) was used to create non-dominated rule sets.

The interpretability-accuracy trade-off for fuzzy rule-based classifiers was explored by Ishibuchi and Nojima (2007) using three-objective optimisation problems based on their GBML algorithm. Three different formulations for each multi-objective and single-objective optimisation problem were adopted based on several different considerations. A clear trade-off structure was reportedly visualised for each considered dataset. On the other hand, a clear trade-off structure cannot always be obtained for test patterns due to the possibility of overfitting within the duration of the training.

The generalisation ability of such systems has been the focus of a number of researchers. Mansoori et al. (2008) have put forward the proposal of extracting a compact set of good fuzzy rules from numerical data through a steady-state genetic algorithm, which begins with a particular fuzzy partition of the feature space. Enabling the search to go faster, this approach uses fitness function measure, which tends to discard rules with longer antecedent parts. Characteristically more specific than shorter rules, these ones have the tendency of over fitting the classifier. This is done by locating lonely instances that fall into covering area of generated rules.

For supervised cases, it is presumed that there exists a training set of objects with known class labels. Hence, in these cases, some classification algorithms are generally employed to determine the effectiveness of the selected feature subset. The efficiency of the feature subset is ascertained based on how well it can classify the training objects with a certain classifier. The functioning of the feature subset is assessed through some classification implementation metrics. One of the founding studies in this regard is Emmanouilidis, Hunter, and MacIntyre (2000), which analysed how two objective functions correspond to the rate of misclassification and the number of features. Both the objective functions make use of a minimum set of features with the intent to reduce the rate of misclassification as much as possible.

For interestings rules, little explored in this field for its’ difficulity. But they are adopted in this thesis for its importance especially in real world applications. Some of the work related to this subject:

Noda, Freitas, and Lopes (1999) promoted a GA approach intended to identify interesting rules in line with the situation that most data mining jobs focus on predictive comprehensibility and accuracy. The fitness function is made of two parts. The first part measures the level of interestingness of the rule, whereas the other part measures predictive accuracy. The calculation of the degree of interestingness is based on the following indications: the bigger the relative frequency (in the training set) of the amount being projected by the consequent, the less interesting it becomes. Therefore, when the value of the goal attribute is less common, the rule prediction becomes more interesting.

An other work related to our work was proposed by Noda, Freitas, and Yamakami (2003). A GA which searches for rules that are interesting and accurate at the same time, depending on a certain rule-interestingness measure. The work is applied by following an objective approach for discovering interesting rules.

Dehuri et al. (2008) employed a Michigan approach with the objective functions being predictive accuracy, comprehensibility, and interestingness. The function of the number of attributes in that rule that has to be minimised is called comprehensibility, whereas the information gained to quantify how interesting the rule is called interestingness. Predictive accuracy is calculated with the use of the objects covered by such rule.

Most of other work, Ishibuchi et al. (2001) apply a subjective approach depending on the user or expert to evaluate the interesting rules rather thn discovering them from the experimental data. In real world applications such as AFR prediction, is very difficult for the expert to discover interesting rules.

Most of the work in the field of multi-objective evaluation problem will be managed by an aggregate method calle the weighed sum.

In Dehuri et al. (2008), Gacto et al. (2010), Ishibuchi et al. (2001), Ishibuchi and Nojima (2007), and Ishibuchi and Yamamoto (2004) the weighted sum was used to solve the multi-objective evaluation. But there is one inevitable problem of this approach and that is the generation of a set of weights which correctly measures the objectives when nothing is known about the problem, which might be difficult if the number of objectives is high (Mandal & Pal, 2011; Xu & Zhou, 2011). These weights are generally identified using a trial and error method.

In this thesis a new Composition method will be used to solve this problem.

2.5 Summary

This chapter provides a presentation of the literature review. The discussion consists of basic concepts of soft computing methods, rule extraction approaches, feature selection, as well as past works about the learning of NFSs including several approaches. As a conclusion for this chapter, the following points can be detected:

- No machine learning model in previous work for AFR.
- For reducing the time of learning and complexity of modeling, it is important for the feature selection and rule extraction to be in the same level. Inorder to address the nonlinear interactions between the features.
- Addressing both the probabilistic (randomness) and possibilistic (imprecise) uncertainty when dealing with real world systems, they complete each other.
- Tradeoff of accuracy and interpretability is very important. Adding the interestingness is benefit and so important in real world applications to discover hidden rules and information.

CHAPTER THREE
RESEARCH FRAMEWORK AND METHODOLOGY

3.1 Introduction

This chapter presents the framework and methodology of this research. It starts with Section 3.2 that depicts the research framework and the methods used to achieve the objectives of the research. Based on the research objectives, the proposed methods are presented in Section 3.3, which briefly explains the roles of each method and the experimental design and the selected benchmarks for evaluating the whole approach. Finally, this chapter is summarized in Section 3.4.

3.2 The Research Framework

The high-level focus of this research is to elaborate on the methodology for data set classification to be later applied to ANFIS. Figure 3.1 presents the various research phases step by step, including data collection, construction of the rule classification algorithm, and evaluation of the constructed model with the derived approach.

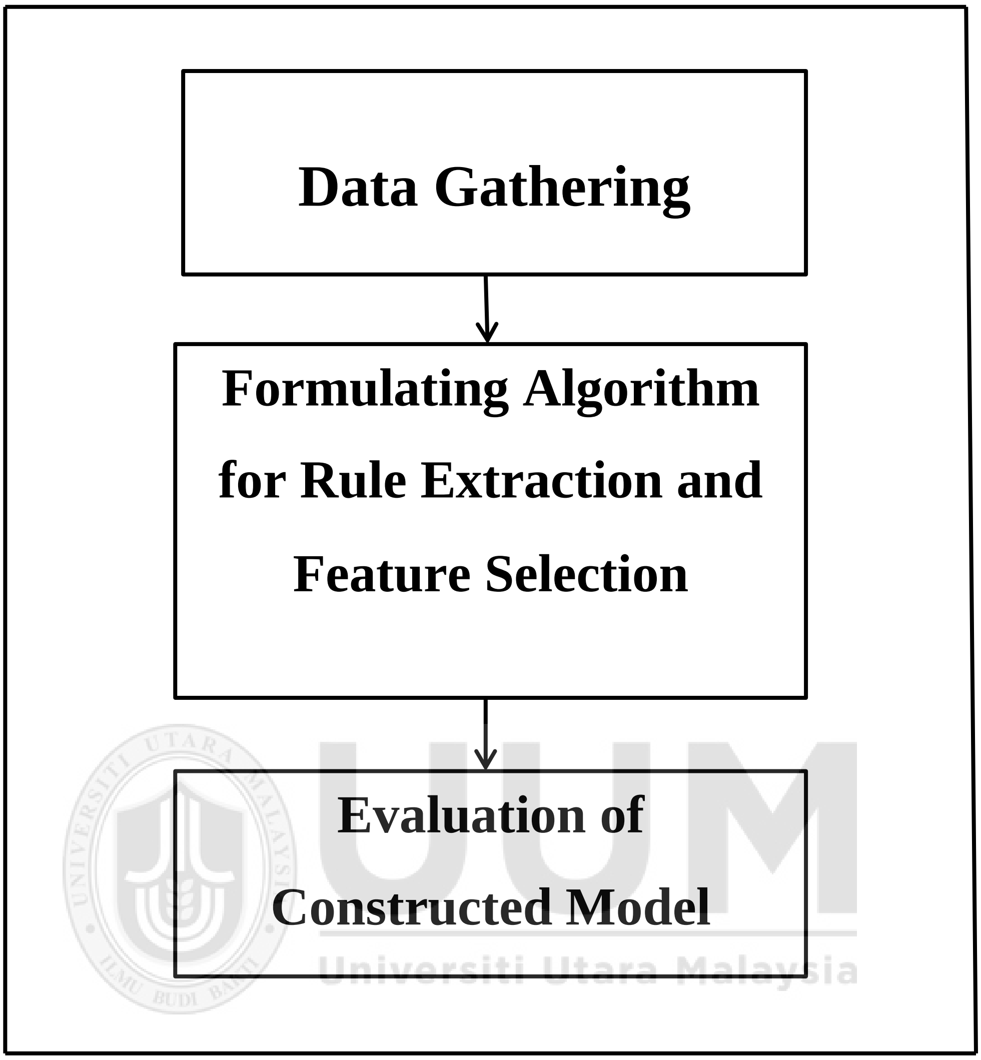

Data Gathering
Formulating Algorithm for Rule Extraction and Feature Selection
Evaluation of Constructed Model

Figure 3.1. High-level Research Framework

The first step is applying data gathering. The basis of the data collection for the research is the data fusion technology – a reliable method for precise problem evaluation and in-depth problem analysis for the purpose of further threat identification, based on validating knowledge from different sensors (Durrant-Whyte & Henderson, 2008; Ma, 2001; Mitchell, 2007). The technology is relatively innovative and focuses on improved data collection efficiency due to the application of data and information derived by multiple instead of single sensors, which intensifies the inference and results in more accurate outcomes. This improve efficiency is to a large extent due to the fact that when a singular sensor is concerned, its precision may be affected by external factors like thermal and system noise, which in turn may cause functionality deterioration.

The present sensor technology specifications obstruct the automation systems' synergy with their environment. This is one of the reasons a single sensor cannot provide the accurate results needed for any automation function due to its limited dependability and efficiency, which is why the sensor combination approach is necessary to counterbalance the negative acting forces and to increase data validation accuracy. In this way, should a singular sensor be affected in a detrimental way by an external factor, the rest of the sensor can compensate for the disintegration of knowledge. In the case of deriving information from multiple source types, the end result becomes even richer in meaning and precision. Sensor combination also presents a solution for the occasional inefficient use of information, as the sensor should in general convey a fast, brief decision. The more information is available on a topic, the more time it will take the system to analyse it and present a viable solution to a problem, while in the case of multiple sensors, even though the data load is higher, the intelligence is more systemised and follows easier interpretation. Multiple sensors provide measurements from various environmental aspects which enhances the overall structure of the mechanism. The proper application of sensor combination can diminish the disadvantages of singular sensors and grow into one big reliable source which can be trusted to act accurately and in a timely manner.

The second step is the formation of the rule extraction and feature selection algorithm. A hybrid algorithm was formulated in the current research, aiming at creating a forecast about the AFR ratio using directly the provided data. The GA-ANFIS algorithm (Rawat & Burse, 2013; Y. Wang et al., 2003), which is in essence the already mentioned hybrid algorithm, is constructed in two phases: 1) FIS extraction from the data with no expert intervention, and 2) ANFIS modelling.

The third step is the model evaluation. The layered tenfold cross-validation method is applied in the current research for the purpose of classifier performance assessment as a step to achieve the last objective of this study.

The ultimate goal of diversity strategies in this research is to improve algorithm performance. Figure 3.2 extends the aforementioned steps in detail and gives a conceptual view about the proposed approach.

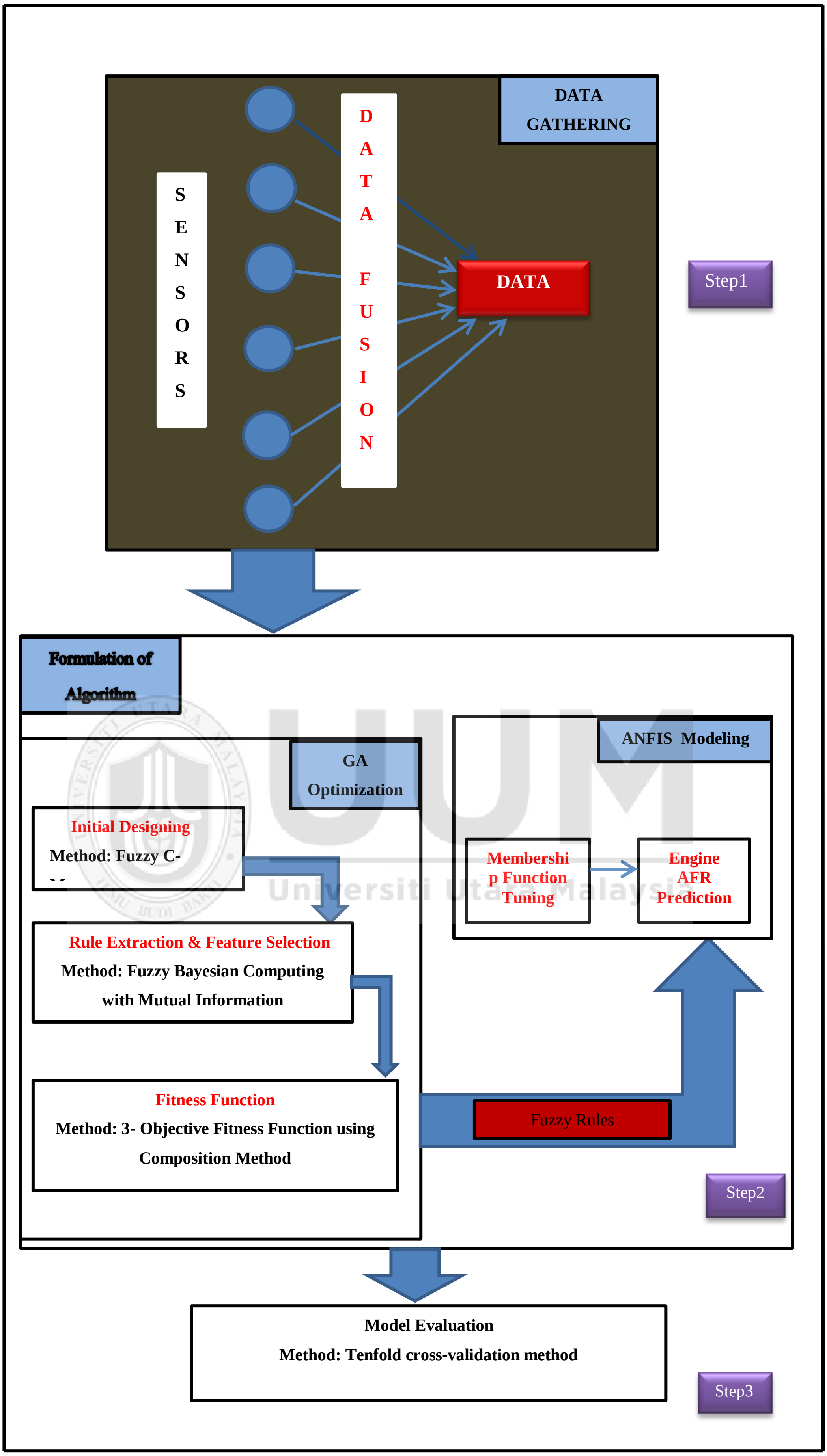

DATA GATHERING
SENSORS
DATA FUSION
DATA
Step1
Formulation of Algorithm
GA Optimization
Initial Designing
Method: Fuzzy C-
- -
Rule Extraction & Feature Selection
Method: Fuzzy Bayesian Computing with Mutual Information
Fitness Function
Method: 3- Objective Fitness Function using Composition Method
ANFIS Modeling
Membership Function Tuning
Engine AFR Prediction
Fuzzy Rules
Step2
Model Evaluation
Method: Tenfold cross-validation method
Step3

Figure 3.2. Low-level Research Framework

3.3 Research Methods

This section presents the proposed methods and draws the roadmap for understanding the proposed approach. The data gathering method is presented in Subsection 3.3.1. The formulation of algorithm and the model evaluation methods are highlighted in Subsection 3.3.2 and Subsection 3.3.3 respectively.

3.3.1 Data Gathering

For the purposes of this research, a test engine was used to obtain a huge engine information load. Out of the multiple factors singled out for the collection, measurement, and analysis of air fuel ratio data, six were selected as the ones having the greatest impact on the end result –TPS, Pulse Width-Injection Opening Time (PW), MAP, Coolant Engine Temperature (CLT), MAT), and RPM. The data gathering concerns the first objective in this research. More information about the data will be discussed in chapter four.

3.3.2 Formation of the Algorithm

The way this research has constructed the algorithm is presented in Figure 3.2. The main objective is creating a rule-based FIS, distinguished by functional simplicity and solid level of intelligibility from all points of attribute and rule numbers, as well as by exceptional overall system precision, involves the design of appropriate rules and a relevant factor choice in order to be able to account for any possible non-linear synergies between the fuzzy rule bases systems and the features. The current research has applied a method which can extract both the rules and the features from the provided data at the same time (Chen et al., 2012; Pal & Saha, 2008). The method can, therefore, be relied on for the determining of an efficient feature set which can create appropriate rules for the given issue.

Two possible titles can be assigned to the mentioned algorithm concering the feature selection and RB extraction.

3.3.2.1 Optimisation based Genetic Algorithm

A steady state genetic algorithm is selected for the determination of the best feature set since the feature choice itself is in essence a search issue involving a decision towards the most reliable input subset (Oh, Lee, & Moon, 2004; C. K. Zhang & Hu, 2005).

The stages of designing the GA are as following:

1. **Initial Designing**

Standing for the second objective, the initial construction is performed with the use of FCM clustering because of its prior utilisation in a wide range of operations like pattern recognition, classification, modelling, and identification.

2. **Rule Extraction & Feature Selection**

As for achieving the second and third objective, on the basis of the first constructed RB, a calculation of the rule output values was performed, involving the Bayes inference rule for all relevant training data points.

From here on, a remodelling of the intial RB should be executed through the application of any of the evolutionary algorithms. For the purposes of the current research, a genetic algorithm of real value has been applied, namely the Iterative

Approach, in which each rule is represented by a chromosome. A single rule is derived by the use of the genetic algorithm with iterative.

As previously described by (Durrant-Whyte & Henderson, 2008), processes and examinations of probabilistic characteristics have been used for the enhancement of the current research's sensor fusion, in addition to Bayes' rule for data fusion which again is inspirational for the probabilistic information combination methods. Bayes' rule is proven to have an exceptionally efficient combinatory quality when it comes to sensor evaluation. This is why it is believed that the Bayesian concept is among the most widely used approaches on the second level of sensor data fusion (Dailey, Harn, & Lin, 1996; Mitchell, 2007). More than one sensor in data fusion projects can, however, have serious disadvantages besides the multiple benefits. Among them are conflicting data which can be detrimental for the whole project. Regardless of this threat, in most cases, the Bayesian theory application has turned out to produce highly successful results. The utilisation of probabilistic functions towards the unknown state design leads to an accurate system outcome, providing all the correct action sets. Despite the fact that a single sensor may not be reliable enough a source, a sensor fusion binds the sensor decisions together and analyses them against a threshold which has been established prior to the experiment. As stated by Hu (2014), Bayesian classifiers' major advantage lies in the practical application of the generic analysis approach, not in their theoretical problem-solving characteristics.

The objective of rule derivation and feature choice is presented as a problem of conditional likelihood. Consequently, the research has been able to accurately connect the dots between the previously approved statistical structure of probabilities and the factor choice made on the level of joined criteria for the combined sensors (Brown et al., 2012). MIM is the active norm selected for the study. The likelihood is ensured by the model likelihood applied as the loss function. At this stage, the study is yet to see the selection of the feature which will influence the data flow between that same feature and the class label most, which will become the base for the singling and leaving out of the available features.

3. Fitness Function

Where the fitness function is concerned, a 3-objective Genetic algorithm is applied in order to achieve top efficiency of the rule database. It does so by employing a composition method for designing the fitness function using the three appointed targets. The objectives themselves include interpretability level, interestingness level, and quality of the rule (Dehuri et al., 2008; Noda et al., 1999; Noda et al., 2003).

The GA discovers the rules, which were subject to evaluation according to three criteria, namely:

(a) Rule's Quality.

This represents the calculation of the rule quality

$$Q = \frac{TP}{TP+FN} + \frac{TN}{TN+FP} \qquad (3.1)$$

Where,

- TP stands for the number of examples that are covered by the rule and has the same class from the class that is predicted by the rule,

- FP stands for the number of examples that is covered by the rule and has a different class from the class that is predicted by the rule,

- FN is the number of examples that is not covered by the rule, but has the same class with the class that is predicted by the rule, and

- TN is the number of examples that is not covered by the rule, and has a different class with the class that is predicted by the rule.

(b) *Degree of Interpretability (Rule's Complexity)*

A value (K) stands for a number of rules and the number of these rules' conditions can be considered for its comprehensibility characteristic. (K) is an inversely proportional value that is relative to the number of conditions N(X) in X, the rule's antecedent. Whether a rule can have a minimum of M conditions, the comprehensibility can thus be defined as:

$K = 1 - (N(X)/M)$ (3.2)

(c) *Degree of interestingness (Surprisingness)*

In data mining, a key factor is concerned with the discovered knowledge and that it should somehow be interesting; here, the term "interestingness" is arguably related with "surprisingness" (unexpectedness), novelty, and usefulness. To date, there are a number of objective criteria proposed in the measurement of the interestingness of rule. Frietas (1999) proposed one of these criteria based on information theory.

Interestingness = $(1/(\sum_{i=1}^{k} InfoGAin(Ai))/k)$ (3.3)

Where, InfoGain(*Ai*) is the information gain (MI) of the *i*th attribute occurring in the rule antecedent and *k* the number of attributes occurring in the rule antecedent.

The purpose of the three objectives here is towards maximising the rule's quality, maximising the degree of interpretability, and maximising the degree of interestingness, as illustrated in Figure 3.3. A Composition Function method is used to aggregate the three objectives without requiring weights in order to generate a fitness function which best stands for the desirability of the chromosome (or FIS).

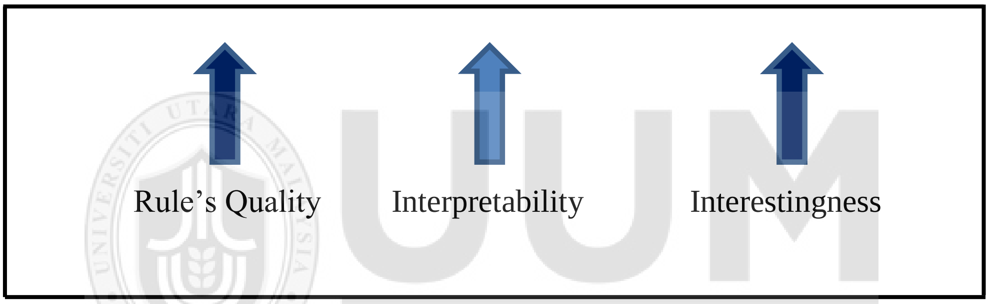


Figure 3.3. Composition Function of three objectives

A number of the composition functions are used in this work, as,

- min (max (Quality, Interpretability), Interestingness)
- probor (min (Quality, Interpretability), Interestingness)
- Probor (max (Quality, Interpretability), Interestingness)

Where min stands for the operator (AND) between two values as it takes always the minimum between them, max stands for the operator (OR) between two values as it takes always the maximum , Avg stands for the average, and Probor stands for probabilistic OR.

Figures 3.4, 3.5, and 3.6 illustrates the probor, max, and min function respectively. Where, *y1*, *y2* are the variables and *yy* stand for the function concerned.

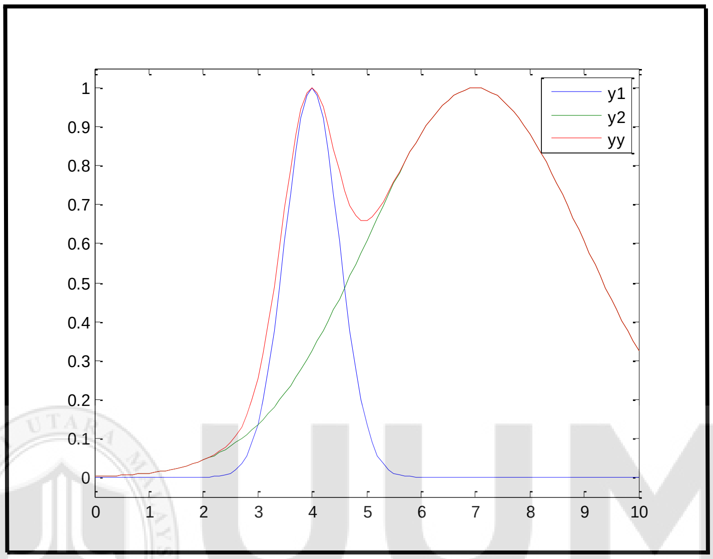


Figure 3.4. Probor Function

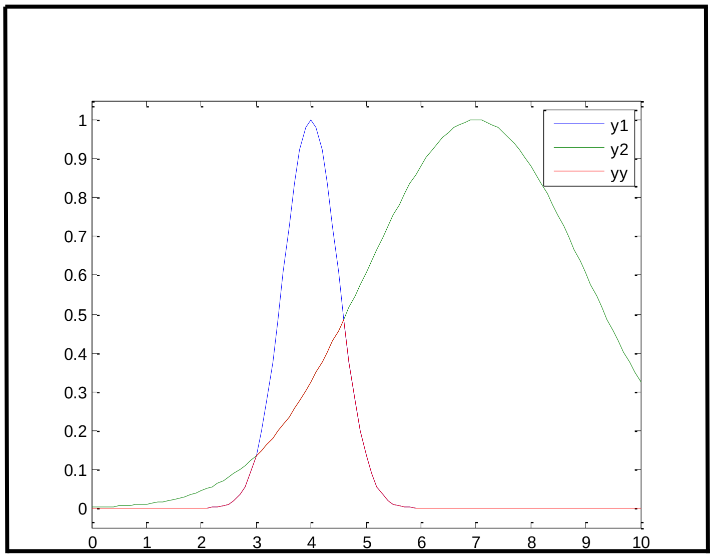


Figure 3.5. Max Function

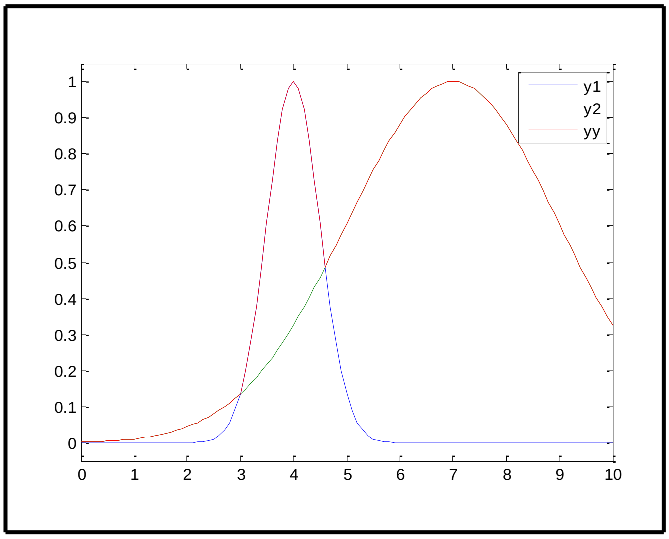


Figure 3.6. Min Function

3.3.2.2 ANFIS Modelling

The rules already devised and improved by the genetic algorithm will consequently be employed for the purpose of AFR ratio forecast by the ANFIS, which will in turn upgrade the MFs – the result of the clustering of the FCM.

3.3.3 Model Validation

Figure 3.7 depicts the layered tenfold cross-validation method applied in the current research for the purpose of classifier performance assessment as a step to achieve the last objective of this study. Kohavi (1995) describes cross validation as an approach for evaluating classifier performance in the presence of new data. It is based on informational set classifications of ten-fold sets. The *i*-th set should be regarded as the test set, while all other sets – as the training ones, in order for accurate classifier assessment to be performed.

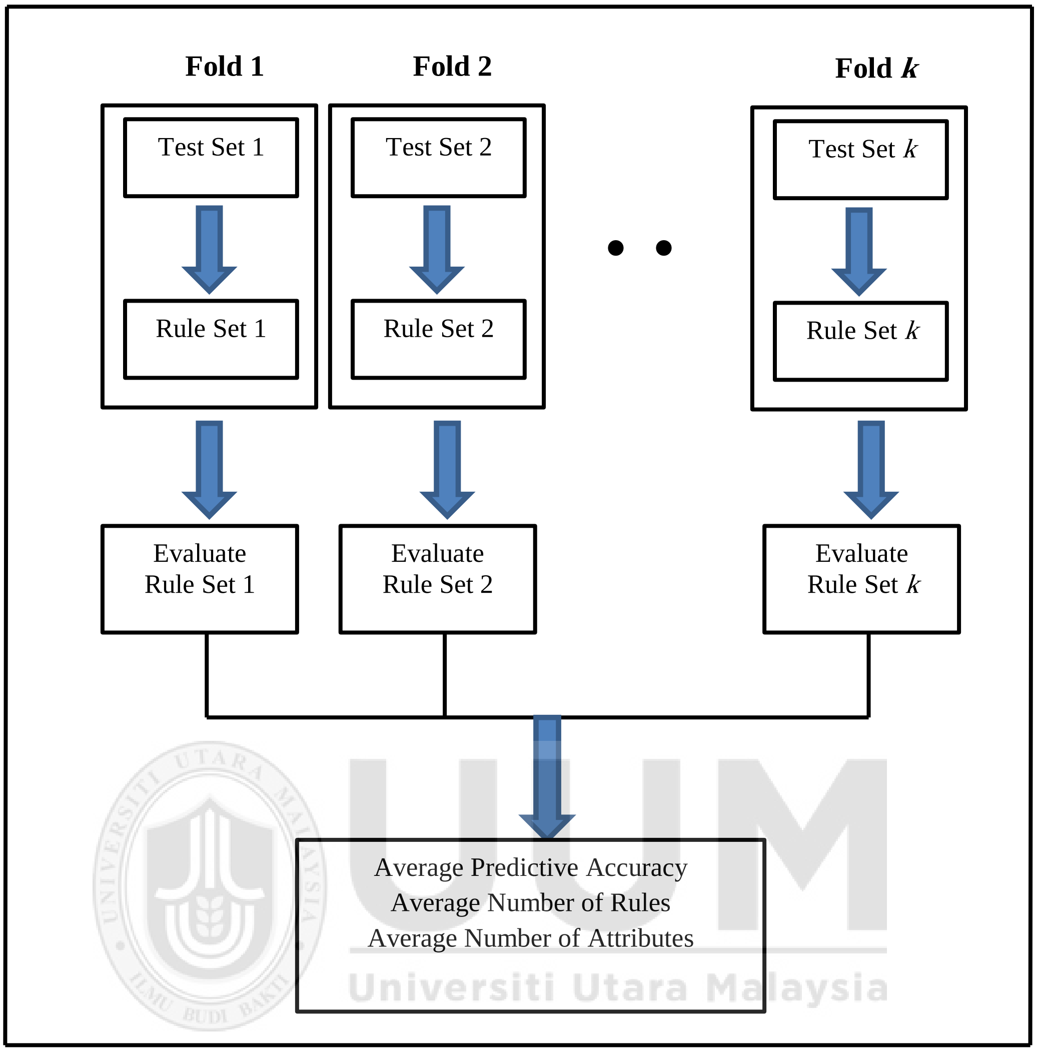


Figure 3.7. k-fold Cross Validation Procedure

The AFR proportion forecast was appraised through the use of Root Mean Square Error (RMSE) by correlating the test AFR from the experiments and the ANFIS model. As shown in the equation below, the average predictive accuracy is reached by dividing the sum of the predictive accuracies by the total set count. Average predictive accuracy (RMSE) = $\frac{1}{10}\left(\frac{\sum_{i=1}^{n}(y'-y)^2}{(n-1)}\right)^{\frac{1}{2}}$ (3.4)

In the equation above, *n* stands for the data number, *y* stands for the actual output, and *y'*– for the conjectured one.

The pseudocode for Creating Training Set and Test Set in K-Fold Cross Validation is showed in Figure 3.8.

```
Algorithm 3.1: K-Fold Cross Validation
1 SplitLists: List of k sets created by splitting the DataSet
2 FOREACH SplitSet Si in SplitLists
   2.1  SET TestSet = Si
   2.2 SET TrainingSet = Remaining sets merged
3 END
end-algorithm
```

Figure 3.8. Pseudocode for K-Fold Cross Validation

The evaluation of the presented model will additionally take into account the median rule count and the number of attributes assigned to the rules.

Therefore, there are numerous algorithms that can be used for understanding of the NFS. This study thus relates the performance of the proposed algorithm with our learning algorithms, which include FCM clustering and grid partition. In addition, the proposed algorithm performance is related to the GA-Bayesian classifier, the possibility of which depends on the frequency.

3.4 Summary

In this chapter, the methodology used for the study is explained further. The methodology comprises three phases: data collection, formulation of algorithm, and validation of rule. Data collection shows how to gather data. The formulation of algorithm has two stages, which are FIS building and ANFIS modelling. It is important to identify the best feature selection models for the purpose of dimensionality reduction, which is an approach used in optimising the rules built by the model.

CHAPTER FOUR
DATA COLLECTION AND EMPLOYED INSTRUMENTS

4.1 Introduction

This chapter presents a discussion about the data used in the research. It starts with Section 4.2 that depicts the data characteristics of the experiments. In Section 4.3 briefly explains the instrument used to collect the data. Section 4.4 discusses the data collection and the methods used. Finally, this chapter is summarized in Section 4.5.

4.2 Data characteristics

The data was collected from University Science Malaysia engines laboratory using a CB 125cc air cooled 4-stroke engine from Taiwan manufacturer 'Adly' equipped with a Euro-3 compliant Focus Applied Technologies gasoline EFI system and 3-way catalyst. The lab is illustrated in Figure 4.1.

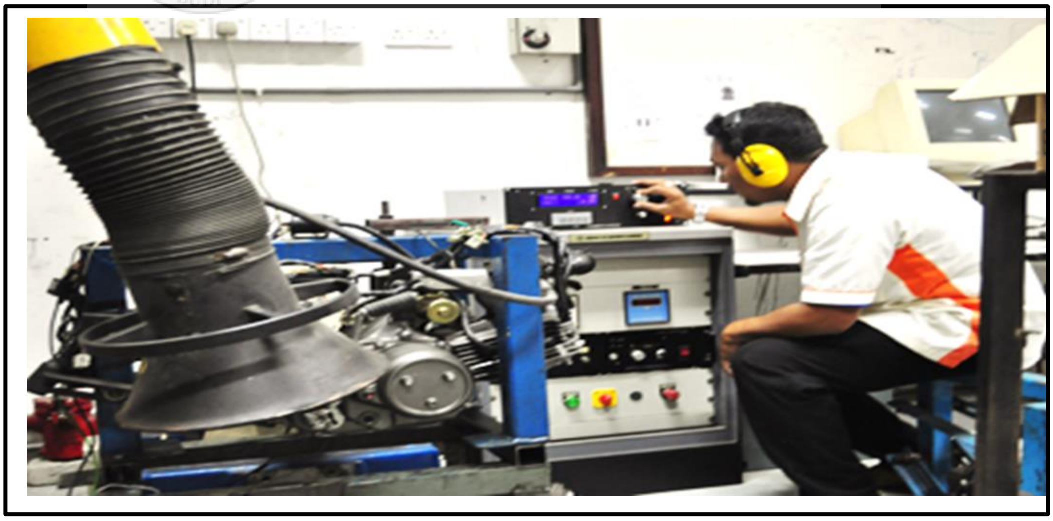


Figure 4.1. Test Engine in University Science Malaysia engines laboratory

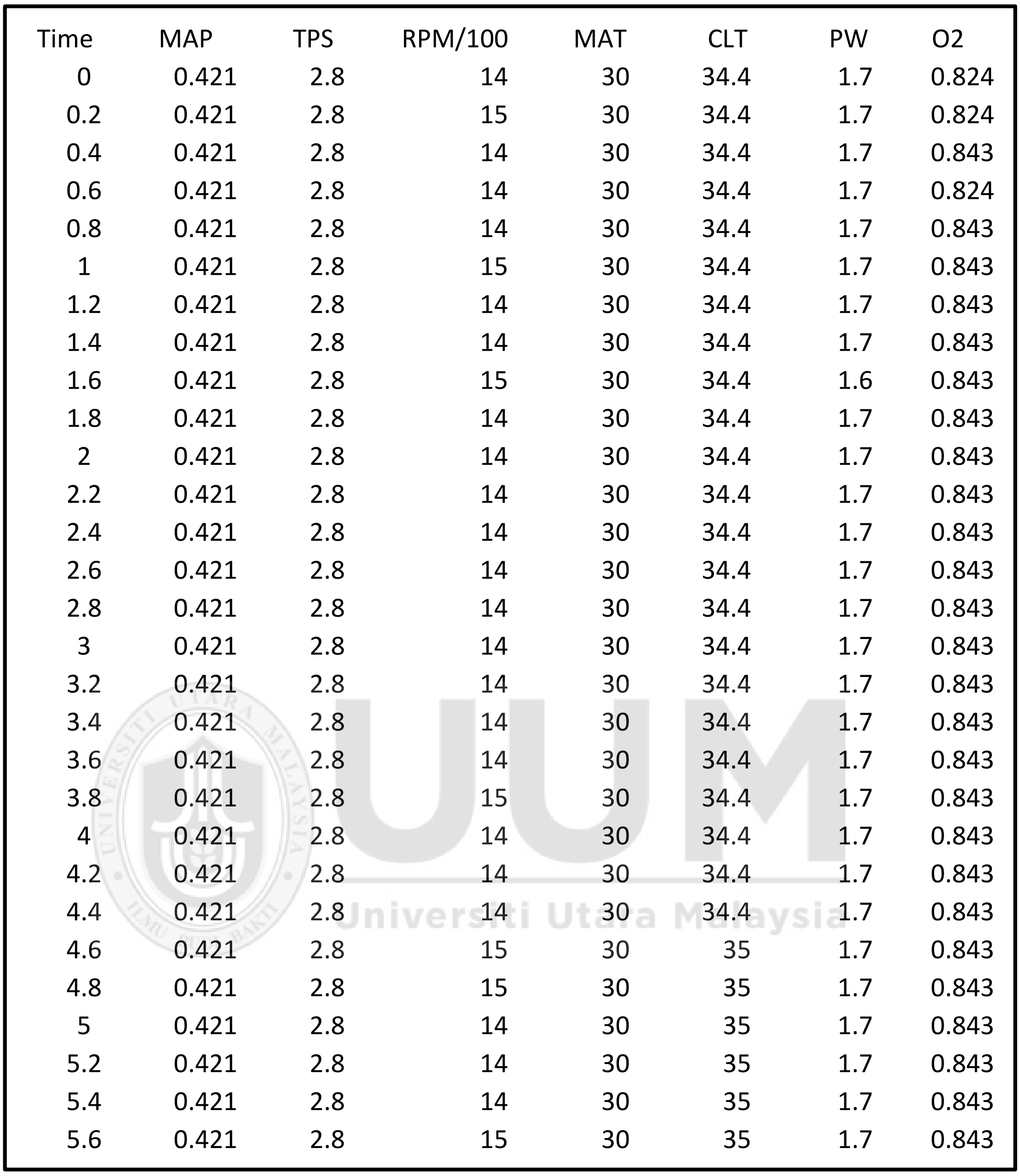
The AFR data sets are a total of three data sets which are used for the research. This number includes engine operation of 15,182 samples selected from the chosen dataset's testing and training information for the purpose of achieving firm temporary states.

The scope of the data factors is as follows: MAP [0 - 1.05] bar, TPS [0% - 194.49%], RPM [0 - 8900] revolution per minute, MAT [26.7C° - 34.4 C°], CLT [29 C° - 141 C°], and PW [0.6ms – 7.9ms]. Table 4.1 depicts he characteristics of the experimental data sets. Figure 4.2, 4.3, 4.4 illustrates samples from Data Set1, Data Set2, and Data Set3 respectively.

Table 4.1

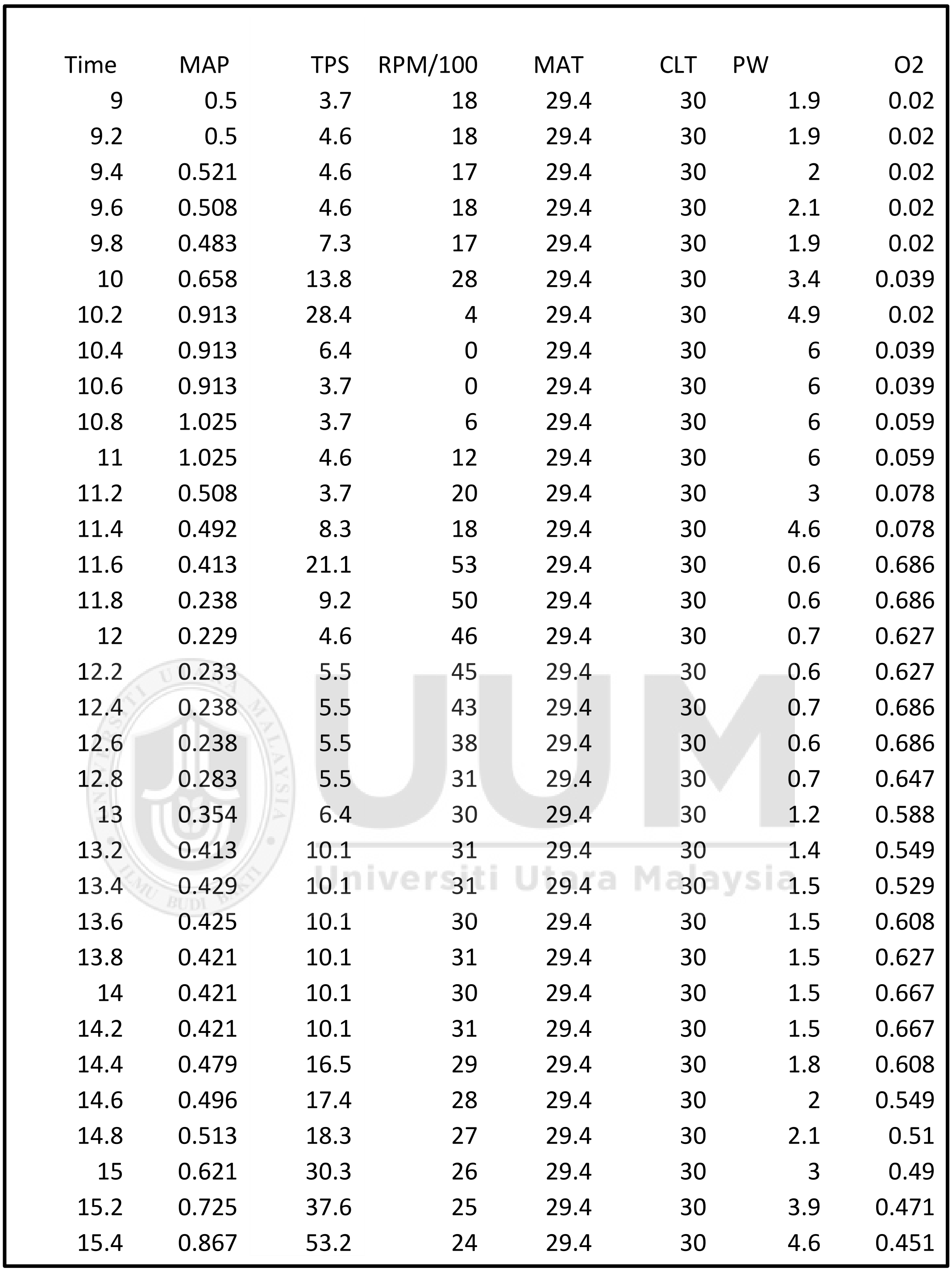
Characteristics of the Experimental Data Sets

Data Sets	**Number of Examples**	**Number of Factors**
Data Set1	3500	6
Data Set2	5776	6
Data Set3	5906	6



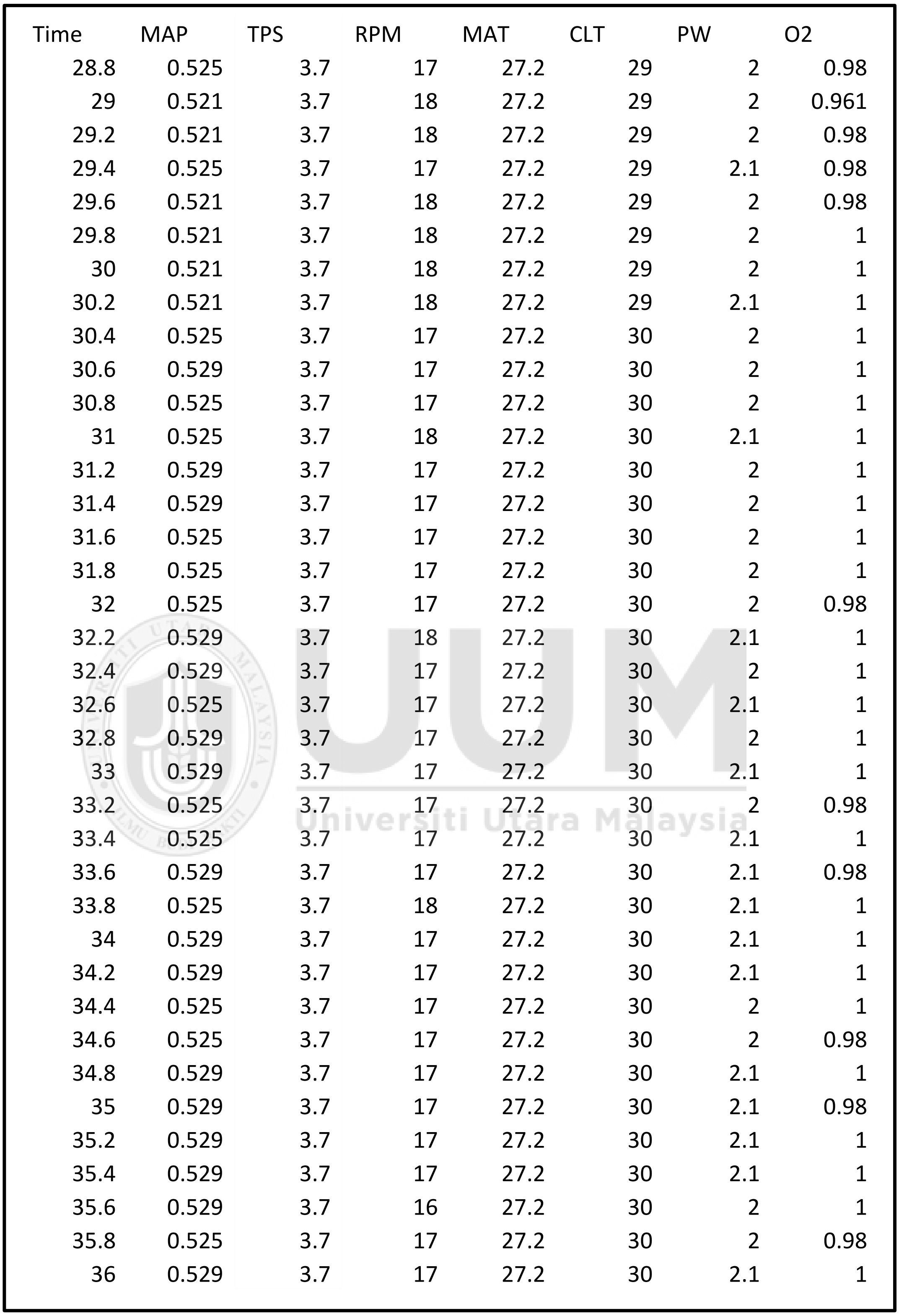
Time	MAP	TPS	RPM/100	MAT	CLT	PW	O2
0	0.421	2.8	14	30	34.4	1.7	0.824
0.2	0.421	2.8	15	30	34.4	1.7	0.824
0.4	0.421	2.8	14	30	34.4	1.7	0.843
0.6	0.421	2.8	14	30	34.4	1.7	0.824
0.8	0.421	2.8	14	30	34.4	1.7	0.843
1	0.421	2.8	15	30	34.4	1.7	0.843
1.2	0.421	2.8	14	30	34.4	1.7	0.843
1.4	0.421	2.8	14	30	34.4	1.7	0.843
1.6	0.421	2.8	15	30	34.4	1.6	0.843
1.8	0.421	2.8	14	30	34.4	1.7	0.843
2	0.421	2.8	14	30	34.4	1.7	0.843
2.2	0.421	2.8	14	30	34.4	1.7	0.843
2.4	0.421	2.8	14	30	34.4	1.7	0.843
2.6	0.421	2.8	14	30	34.4	1.7	0.843
2.8	0.421	2.8	14	30	34.4	1.7	0.843
3	0.421	2.8	14	30	34.4	1.7	0.843
3.2	0.421	2.8	14	30	34.4	1.7	0.843
3.4	0.421	2.8	14	30	34.4	1.7	0.843
3.6	0.421	2.8	14	30	34.4	1.7	0.843
3.8	0.421	2.8	15	30	34.4	1.7	0.843
4	0.421	2.8	14	30	34.4	1.7	0.843
4.2	0.421	2.8	14	30	34.4	1.7	0.843
4.4	0.421	2.8	14	30	34.4	1.7	0.843
4.6	0.421	2.8	15	30	35	1.7	0.843
4.8	0.421	2.8	15	30	35	1.7	0.843
5	0.421	2.8	14	30	35	1.7	0.843
5.2	0.421	2.8	14	30	35	1.7	0.843
5.4	0.421	2.8	14	30	35	1.7	0.843
5.6	0.421	2.8	15	30	35	1.7	0.843

Figure 4.2. Sample of Data Set1



Time	MAP	TPS	RPM/100	MAT	CLT	PW	O2
9	0.5	3.7	18	29.4	30	1.9	0.02
9.2	0.5	4.6	18	29.4	30	1.9	0.02
9.4	0.521	4.6	17	29.4	30	2	0.02
9.6	0.508	4.6	18	29.4	30	2.1	0.02
9.8	0.483	7.3	17	29.4	30	1.9	0.02
10	0.658	13.8	28	29.4	30	3.4	0.039
10.2	0.913	28.4	4	29.4	30	4.9	0.02
10.4	0.913	6.4	0	29.4	30	6	0.039
10.6	0.913	3.7	0	29.4	30	6	0.039
10.8	1.025	3.7	6	29.4	30	6	0.059
11	1.025	4.6	12	29.4	30	6	0.059
11.2	0.508	3.7	20	29.4	30	3	0.078
11.4	0.492	8.3	18	29.4	30	4.6	0.078
11.6	0.413	21.1	53	29.4	30	0.6	0.686
11.8	0.238	9.2	50	29.4	30	0.6	0.686
12	0.229	4.6	46	29.4	30	0.7	0.627
12.2	0.233	5.5	45	29.4	30	0.6	0.627
12.4	0.238	5.5	43	29.4	30	0.7	0.686
12.6	0.238	5.5	38	29.4	30	0.6	0.686
12.8	0.283	5.5	31	29.4	30	0.7	0.647
13	0.354	6.4	30	29.4	30	1.2	0.588
13.2	0.413	10.1	31	29.4	30	1.4	0.549
13.4	0.429	10.1	31	29.4	30	1.5	0.529
13.6	0.425	10.1	30	29.4	30	1.5	0.608
13.8	0.421	10.1	31	29.4	30	1.5	0.627
14	0.421	10.1	30	29.4	30	1.5	0.667
14.2	0.421	10.1	31	29.4	30	1.5	0.667
14.4	0.479	16.5	29	29.4	30	1.8	0.608
14.6	0.496	17.4	28	29.4	30	2	0.549
14.8	0.513	18.3	27	29.4	30	2.1	0.51
15	0.621	30.3	26	29.4	30	3	0.49
15.2	0.725	37.6	25	29.4	30	3.9	0.471
15.4	0.867	53.2	24	29.4	30	4.6	0.451

Figure 4.3. Sample of Data Set2



Time	MAP	TPS	RPM	MAT	CLT	PW	O2
28.8	0.525	3.7	17	27.2	29	2	0.98
29	0.521	3.7	18	27.2	29	2	0.961
29.2	0.521	3.7	18	27.2	29	2	0.98
29.4	0.525	3.7	17	27.2	29	2.1	0.98
29.6	0.521	3.7	18	27.2	29	2	0.98
29.8	0.521	3.7	18	27.2	29	2	1
30	0.521	3.7	18	27.2	29	2	1
30.2	0.521	3.7	18	27.2	29	2.1	1
30.4	0.525	3.7	17	27.2	30	2	1
30.6	0.529	3.7	17	27.2	30	2	1
30.8	0.525	3.7	17	27.2	30	2	1
31	0.525	3.7	18	27.2	30	2.1	1
31.2	0.529	3.7	17	27.2	30	2	1
31.4	0.529	3.7	17	27.2	30	2	1
31.6	0.525	3.7	17	27.2	30	2	1
31.8	0.525	3.7	17	27.2	30	2	1
32	0.525	3.7	17	27.2	30	2	0.98
32.2	0.529	3.7	18	27.2	30	2.1	1
32.4	0.529	3.7	17	27.2	30	2	1
32.6	0.525	3.7	17	27.2	30	2.1	1
32.8	0.529	3.7	17	27.2	30	2	1
33	0.529	3.7	17	27.2	30	2.1	1
33.2	0.525	3.7	17	27.2	30	2	0.98
33.4	0.525	3.7	17	27.2	30	2.1	1
33.6	0.529	3.7	17	27.2	30	2.1	0.98
33.8	0.525	3.7	18	27.2	30	2.1	1
34	0.529	3.7	17	27.2	30	2.1	1
34.2	0.529	3.7	17	27.2	30	2.1	1
34.4	0.525	3.7	17	27.2	30	2	1
34.6	0.525	3.7	17	27.2	30	2	0.98
34.8	0.529	3.7	17	27.2	30	2.1	1
35	0.529	3.7	17	27.2	30	2.1	0.98
35.2	0.529	3.7	17	27.2	30	2.1	1
35.4	0.529	3.7	17	27.2	30	2.1	1
35.6	0.529	3.7	16	27.2	30	2	1
35.8	0.525	3.7	17	27.2	30	2	0.98
36	0.529	3.7	17	27.2	30	2.1	1

Figure 4.4. Sample of Data Set3

4.3 Test Engine

The instrument used in collecting the data is the test engine as illustrated in Figure 4.5.

Figure 4.5. Test Engine

A test engine is a facility applied to characterize, develop and test engines, which permits engine operation to operate in various operating regimes and provides different physical variables measurement associated with the engine operation. Engine testing consists of collecting huge amounts of data from different types of tests. Applying a test engine in data collection decreases the cost and also provides situations were it is difficult to apply from vehicles on road such as very high speed or very high load or even very high tempreture. High load here is about applying heavy load such as climbing a mountain. Very high speed such as reaching 8900 revolution per minute is not applicable in real world and even though it can be reached it will be very harmful to the engine, the same thing for high tempreture.

4.4 Data collection

In order to benefit from using multi-inputs without discarding the nonlinear interactions, multi-sensor data fusion technology is used in this study. In Figure 4.1, the Bayes rule helped to indicate this condition.

The Bayes' rule necessitates that the relationship between $\boldsymbol{x}$ and $\boldsymbol{z}$ be encoded as a joint probability or joint probability distribution $P(\boldsymbol{x}, \boldsymbol{z})$ for continuous and discrete variables, respectively. The chain rule of conditional probabilities can be used to broaden the joint probability in two ways.

$$P(\boldsymbol{x}, \boldsymbol{y}) = P(x \mid y)P(\boldsymbol{y}) = P(y| \, x)P(\boldsymbol{x}) \qquad (4.1)$$

Through reorganisation in terms of one of the conditionals, the Bayes' rule is obtained as:

$$P(\boldsymbol{x} \mid \boldsymbol{y}) = \frac{P(y| \, x)P(x)}{P(y)} \qquad (4.2)$$

The result value depends on the interpretation of the probabilities $P(\boldsymbol{x} \mid \boldsymbol{y})$, $P(\boldsymbol{y} \mid \boldsymbol{x})$, and $P(\boldsymbol{x})$.

It is very important to conclude the likelihood of various values of an unidentified state $\boldsymbol{x}$**.** Prior beliefs regarding what values of $\boldsymbol{x}$ can be expected are encoded in the form of relative likelihoods in the prior probability $P(\boldsymbol{x})$. Observation $\boldsymbol{z}$ is made to acquire more information about the state $\boldsymbol{x}$. Such observations are modelled in the form of conditional probability $P(\boldsymbol{y} \mid \boldsymbol{x})$, which describes the probability that the observation $\boldsymbol{z}$ will be made for each fixed state $\boldsymbol{x}$, similar to the probability of $\boldsymbol{y}$ given $\boldsymbol{x}$, for example. New possibilities associated with the state $\boldsymbol{x}$ are calculated from the result of the original prior information and the information obtained via observation. This is programmed in the posterior probability $P(\boldsymbol{x} \mid \boldsymbol{y})$, which defines the probability associated with $\boldsymbol{x}$ given the observation $\boldsymbol{y}$. The marginal probability is used to normalise the posterior and is generally not calculated in this fusion process. The marginal $P(\boldsymbol{y})$ serves an important function in model validation or data association because it provides a measurement of how successful the observation is foreseen by the prior because $P(\boldsymbol{y}) = \int P(\boldsymbol{y} \mid \boldsymbol{x})P(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$.

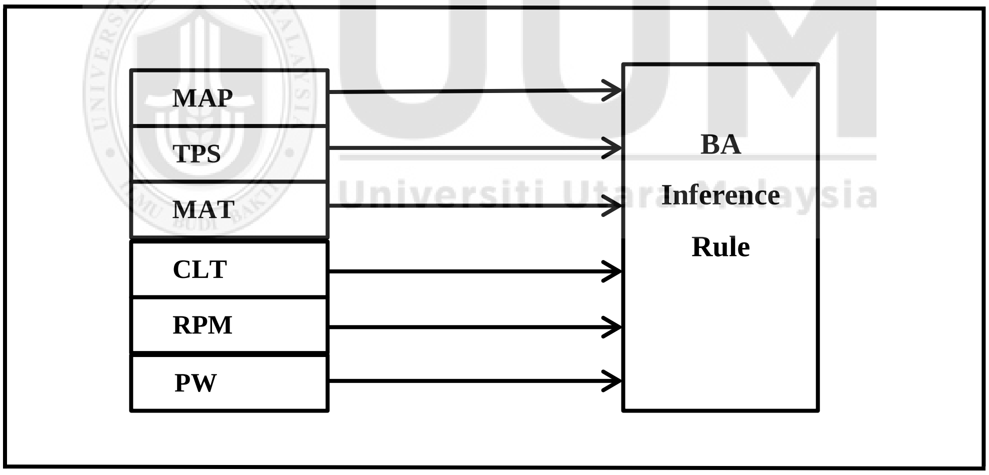


Figure 4.6. Sensor data fusion

A total of six factors (features) have been selected as the most influential. These are TPS, MAP, CLT, MAT, RPM, and PW.

Figures 4.7, 4.8, and 4.9 reveal the behaviour of datasets 1, 2, and 3, respectively.

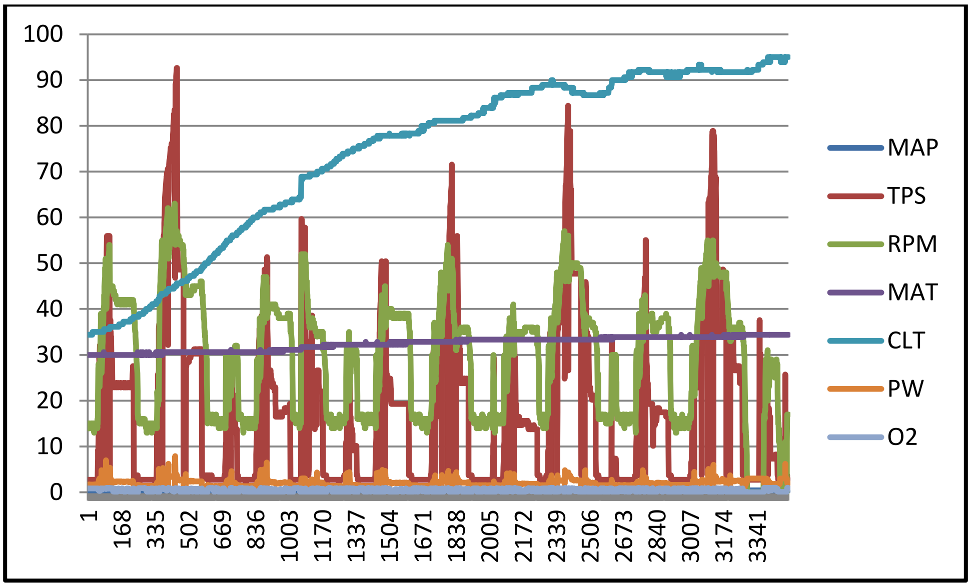


Figure 4.7. Data Set1

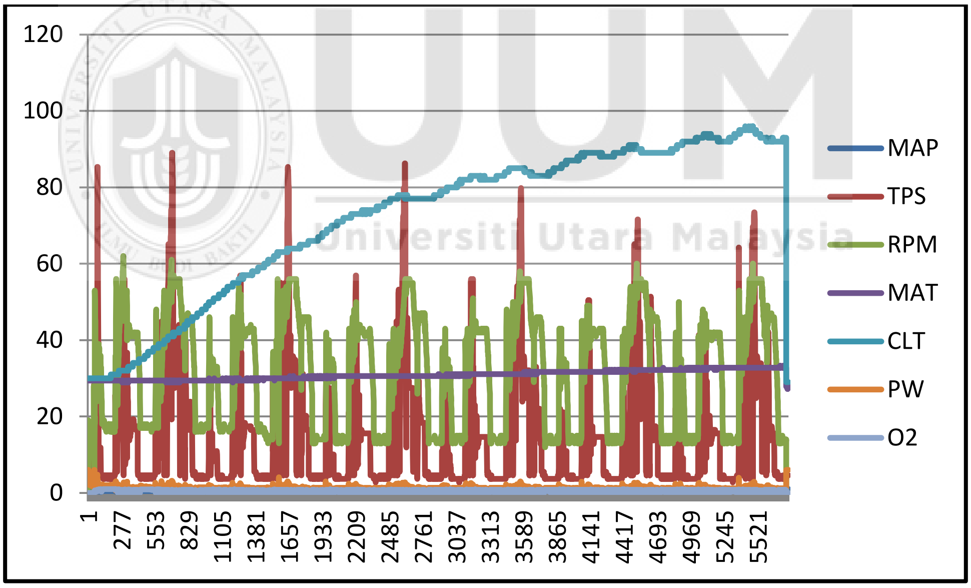


Figure 4.8. Data Set2

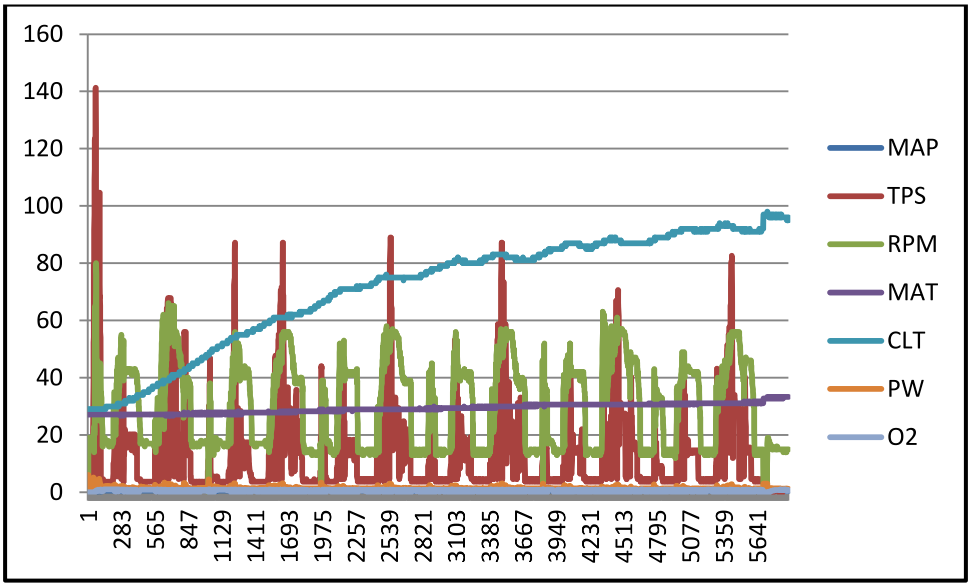


Figure 4.9. Data Set3

From the figures above, it can be noticed that the same cycle is repeated on and on but there are different responses depending on the circumstances faced by the experiments such as the tempreture, load, and others. This is the reason for chosing three data sets not more. These data sets are enough for the model constructed to achieve its aim in trying in discovering hidden information from the experiment data.

4.4 Summary

As a real world application, the data collection is critical and important to achieve the aims of the research.

In this chapter, the data used in the experiments, producing the data sets for the research concerned in the prediction of the AFR, is clarified and explained. The instruments and methods used are also discussed.

CHAPTER FIVE
FEATURE SELECTION AND RULE EXTRACTION ALGORITHM FORMULATION

5.1 Introduction

The construction of a feature selection, as well as the rule extraction algorithm for establishing rule-based fuzzy rules for ANFIS modelling using a GA based Fuzzy Bayesian Classifier (GA-FBC) developed based on the methodology explained in Chapter Three, will be demonstrated in this chapter. The objective of the reseach which is the data collection method used is presented in chapter four. On the other hand, the process of how the algorithm is established and the methods used is discussed in Section 5.2.

5.2 Algorithm Formulation

A number of rules provided in the form comprise a fuzzy rule

R : IF **x** is A_i , THEN **y** is C_i

for $i = 1, \dots, N_r$, where $A_i = \{ A_{i1} , A_{i2} , \dots , A_{in} \}$, $C_i = C_{i1} ,C_{i2} , \dots , C_{im}$, and A_{ij} and C_{ik} are the respective fuzzy sets that define input and output space partitioning. This condition, which is also termed as a premise, comprises a number of antecedents combined by different operators, including AND and OR. C stands for the consequent of the rule.

A zero-order Takagi-Sugeno fuzzy system is used in this study and takes the following form:

If x is A_i and y is B_i then $z = c$

where A_i and B_i are defined as fuzzy, whereas z is a crisply defined function.

Often, feature selection or structure identification is performed in the context of fuzzy rule extraction in a separate phase. Meanwhile, several methods consider the learning machines but typically eliminate one feature (or a set of features) at a time in a stepwise fashion. Thus, the selected feature set may not be the best for the considered problem because a features set can interact among themselves. Moreover, a feature can interact with the tool used to solve the problem. In other words, the best feature set for a NN may not necessarily be the best for either a support vector machine or a fuzzy ruled-based system. In principle, employing an exhaustive search with a machine learning tool used to design the final system can address the problem. Feature selection methods sometimes disregard the subtle nonlinear interaction that occurs between learning system and the features.

In this study, feature selection is applied at the rule level for Takagi-Sugeno-type FS to address the issue of structure identification. This approach is an integrated learning mechanism that can consider the nonlinear interactions that may occur among features and between features and fuzzy rule-based systems. A feature can be eliminated only if no rule uses this feature, such that not all features exist in one rule. This condition will decrease the number of rule attributes and will facilitate dimensionality reduction and rule interpretability.

FRBS building is a classification problem that requires a strategy that selects the class that best fits the rule antecedent as a consequent in the fuzzy rule.

The formulation of the algorithm of feature selection and rule extraction at one level inorder for the prediction of the AFR consists of two phases as mentioned below.

5.2.1 Genetic Algorithm Optimization

A genetic fuzzy rule-based system decreases the amount of knowledge and effort required from an expert in designing the inference system design. Evolutionary computation automates the design steps of the FRBS through rule learning. The GA here is the machine learning tool. The pseudocode of GA is illustrated in Figure 5.1.

Algorithm 5.1: Genetic Algorithm

```
Randomly generate an initial population of individuals of the training data.
Compute the fitness of each individual
repeat
    Select two individuals based on fitness
    Apply genetic operators to selected individuals, creating new individuals
    Compute fitness of each of the new individuals
    Update the current population (new individuals replace old individuals)
until (stopping criteria)
```

Figure 5.1. Pseudocode for GA

The subsequent sections explain the RB extraction steps.

5.2.1.1 Initial construction

This study employs the FCM clustering algorithm to initialise MFs. Meanwhile, subtractive clustering is applied to identify the number of clusters for the FCM clustering algorithm. This operation is a preprocessing for the GA, generating the MFs inorder to determine the final number of varibles in the premise part of the fuzzy rule. This is a prelude for the chromosome encoding. The data set was used as input to the clustering algorithm. The initial number of clusters was set to be relatively high, whereas the cluster merging threshold was set using trial and error. The objective is a minimised function, which approaches zero.

Gaussian MFs are fitted to the clusters using equation (2.1).

$$Guass(x; c, \sigma) = exp\left(-\frac{(c-x)^2}{2\sigma^2}\right)$$

Where, c is the centre of Gaussian MFs and σ refers to the cluster standard deviation.

5.2.1.2 Rule Extraction and Feature Selection Model

Several steps is used to guide in building a fuzzy RB by using an evolutionary algorithm as a genetic algorithm based and a probabilistic classifier as Bayesian. Both feature and rule extraction methods are simultaneously applied for the learning algorithm to be able to search for the best rule and the best set of variables for each rule, as illustrated in Figure 5.2 This work is based on the selection of a subset of features that can distinguish one class from another. This aspect of features that enables class discrimination forms the basis of class dependent feature selection techniques (Bailey, 2001).

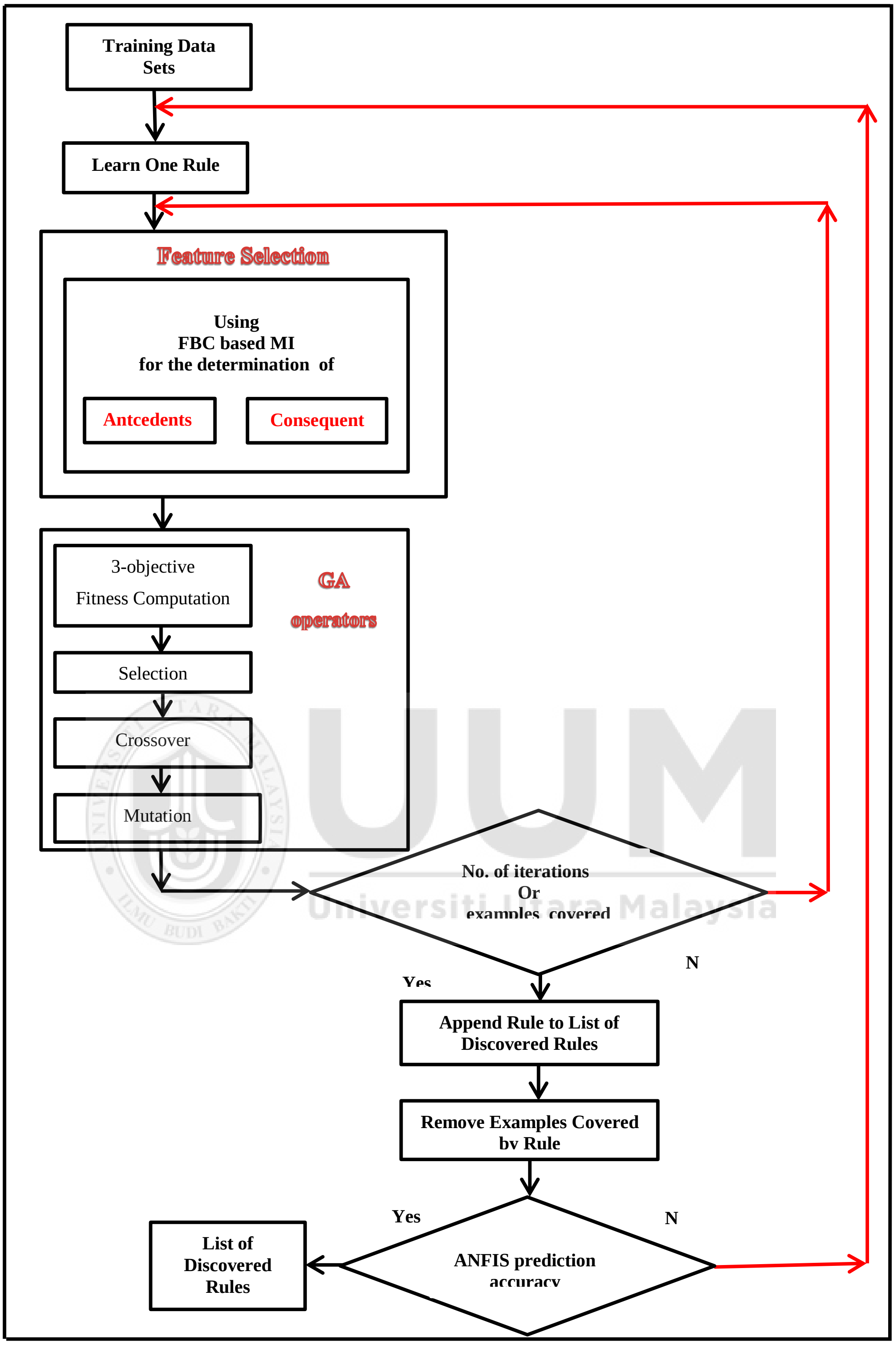


Figure 5.2. Flowchart of Proposed Rule-Based Fuzzy Extraction

The GA is developed employing an IRL approach with a variable length depending on the number of clusters from the initialization stage, but is fixed for each data set.

Each individual in a population contains a certain number of rules between 1 and *max*, where *max* is determined by a decision maker. The ANFIS here is the decision maker.

This work applies an iterative approach to reduce the search space for possible solutions. Although each individual represents a single rule, only the best individuals are selected as solutions. Therefore, the GA provides a partial solution to the learning problem in the iterative model.

Step 1: Fuzzy knowledge initialisation

The initial fuzzy rules should be produced to generate a FIS. The initial fuzzy KB can be produced either automatically with the use of a training dataset or non-automatically through expert knowledge. This study derives initial fuzzy rules from training data. The initialisation of the premise part of the rule is irrelevant here. This process is randomly performed by the GA. The rule consequent is not explicitly encoded in the string. By contrast, this value is given based on the FBC, which selects the class as a consequent in the fuzzy rule that best fits the rule antecedent. Each fuzzy rule represents a chromosome in the GA.

Step 2: Encoding

Encoding refers to the method of revealing a single gene in the genetic space. To optimise the continuous domain, genes should be directly represented as real numbers, such that a chromosome is a vector of floating point numbers especially if the data is aleady a real value. The chromosome size is kept constant depending on the vector length, which is the solution to the problem. Thus, each gene represents a problem variable.

This study employs value encoding. Each input X_i is encoded by n clusters. Owing to the continuous nature of MFs, a real-coded GA is selected to represent each cluster’s membership grade [0,1]. The final chromosome has six inputs and has a length of 6*n, as shown in Figure 5.3.

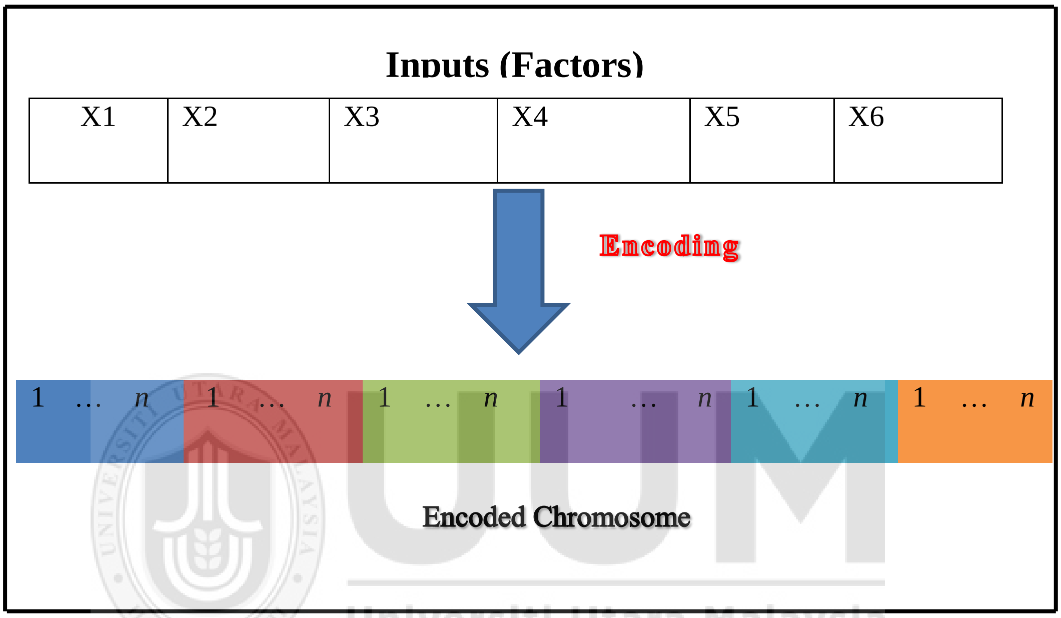


Figure 5.3. Chromosome Encoding Using Clustering

Step 3: Consequent Calculation

The fuzzy rule consequent or class is computed using Bayes’ rule (5.3) according to the fuzzy rule premise randomly generated by the GA.

The Bayes’ rule is important because it provides a principled way of combining observed information with prior beliefs about the state of the world.

Given that this work uses multi-sensors, the Bayes’ rule comes in the following form:

$$P(y1,\ \cdot\ \cdot\ \cdot\ , yn \setminus x\ ) = P(y1 \setminus x)\ \cdot\ \cdot\ \cdot\ P(yn \setminus x)$$

$$= P(\boldsymbol{x}) \prod_{i=1}^{n} P(\boldsymbol{yi} \setminus \boldsymbol{x}) \tag{5.1}$$

The Bayes' rule is used to identify the consequent (class) of the fuzzy rule.

For instance, the $f : x_p \rightarrow C$ is a function that maps x_p to C. x_p is a training pattern, whereas C is a set of classes of training patterns. Aiming to determine the class of pattern x_{test} using the Bayes' rule:

$$C_{xtest} = argmax \{ P(Cj) \prod_{i=1}^{n} P(ai = v) \backslash Cj \} \tag{5.2}$$

Using the maximum posterior (MAP) model to maximise the posterior probability, determing the best class (the consequent of the fuzzy rule) being related to the maximum posterior probability.

The prior probability P(C_j) value can be computed by counting the number of instances that belong to class C_j and dividing this value with the training set cardinality. The P (ai = v| C_j) value can be computed in the same way. The number of instances belonging to class C_j is computed, such that the value of the i`th attribute is equal to v. Then, this is divided by the cardinality of instances of class C_j. Or they can be computed using the uniform distribution as in equation (5.3), were gives equal weights for each class.

These probabilities for a dataset having discrete features can be computed by counting the relevant feature values for each class. However, a dataset with continuous features should first be discretised. No explicit discretisation phase exists in this work. Instead, we use MFs as discretised features. This process involves the replacement of each feature value with a relevant MF.

Membership values for different classes are not normalised and not disjoint, such that the probabilities do not have a sum of one. Therefore, actual probabilities should be normalised (Raitamäki, 2003) using equation (5.3)

$$\sum_i P(xi) \neq 1\,,\; P(x) = \frac{P(x)}{\sum_i P(xi)} \tag{5.3}$$

The likelihood of the Bayes' rule is computed using MI criteria without considering frequency. The criterion 'MIM' is employed to choose the features and classes with the most information. The feature that maximises the MI between such feature and the class label will be chosen to construct the rule. Only features that contribute to this rule will remain, whereas others will be removed. This method estimates entropies directly from data samples depending on the probabilities.

A measure of the value of information that one random variable has about another variable is MI. This meaning is helpful within the context of feature selection because it provides means to quantify the importance of a feature subset with respect to the output vector. This criterion (5.4) will be used to determine the features and to select the best class that fits such features, given that rule extraction and feature selection are done simultaneously.

$$P(Y \backslash X) = \mathrm{argmax}\{I(X;Y)\}$$

$$P(Y \backslash X) = \mathrm{argmax}\{\sum_{i=1}^{Mx}\sum_{j=1}^{Mx} p(X,Y)\log\frac{p(X,Y)}{p(X)p(Y)}\} \tag{5.4}$$

The pseudocode for Bayesian based MI aalgorithm is shown in Figure 5.4.

```
Algorithm 5.2: Bayesian based Mutual Information
1: Calculate Prior Probabilities

2: Normilize the Probabilities
    $\sum_i P(xi) \neq 1 \; , \; P(x) = \frac{P(x)}{\sum_i P(xi)}$
3: Calculate Conditional Probabilities for Feature Values
    $P(Y\backslash X) = argmax\{I(X;Y)\}$

4: Calculate Posterior Probabilities for Each Class
    $C_{xtest} = argmax \{ P(Cj) \prod_{i=1}^{n} P(ai = v)\backslash Cj\}$
5: Classify
end-algorithm
```

Figure 5.4. Pseudocode for Bayesian based MI algorithm

Step 4: Fitness Function

The fitness function, also known as the objective function, gives the measurement of the chromosome’s ability to meet the desired criteria and is employed in global and local search methods.

Using 3-objective fitness function will improve the optimization process. The objectives are the rule’s quality, degree of interpretability, and degree of interestingness. As a multi-objective problem, conjucting these objectives together has the obstacle of using the weights in a weighted sum approach. This research proposed a Composition method to overcome this problem:

- min (max (Quality, Interpretability), Interestingness)
- probor (min (Quality, Interpretability), Interestingness)
- probor (max (Quality, Interpretability), Interestingness)

In Table 5.1, 5.2, 5.3 are some examples to show these functions work:

Assuming A= Rule Quality, B= Rule interpretability, C= Rule interestness

Table 5.1

min(max(A,B),C) Examples

A	**B**	**C**	**min(A,B)**	**max(A,B)**	**min(max(A,B),C)**
0.6	0.5	0.3	0.5	0.6	**0.3**
0.6	0.5	0.9	0.5	0.6	**0.6**
0.6	0.3	0.3	0.3	0.6	**0.3**
0.6	0.3	0.5	0.3	0.6	**0.5**
0.6	0.6	0.4	0.6	0.6	**0.4**
0.6	0.2	0.6	0.2	0.6	**0.6**
0.3	0.9	0.1	0.3	0.9	**0.1**
0.3	0.9	0.9	0.3	0.9	**0.9**
0.3	0.9	0.5	0.3	0.9	**0.5**
0.3	0.1	0.9	0.1	0.3	**0.3**
0.9	0.7	0.3	0.7	0.9	**0.3**
0.9	0.5	0.1	0.5	0.9	**0.1**
0.4	0.8	0.5	0.4	0.8	**0.5**
0.2	0.3	0.5	0.2	0.3	**0.3**

Table 5.2

probor(min(A,B),C) Examples

A	B	C	min(A,B)	max(A,B)	probor(min(A,B),C
0.6	0.5	0.3	0.5	0.6	**0.65**
0.6	0.5	0.9	0.5	0.6	**0.95**
0.6	0.3	0.3	0.3	0.6	**0.51**
0.6	0.3	0.5	0.3	0.6	**0.65**
0.6	0.6	0.4	0.6	0.6	**0.76**
0.6	0.2	0.6	0.2	0.6	**0.68**
0.3	0.9	0.1	0.3	0.9	**0.37**
0.3	0.9	0.9	0.3	0.9	**0.93**
0.3	0.9	0.5	0.3	0.9	**0.65**
0.3	0.1	0.9	0.1	0.3	**0.91**
0.9	0.7	0.3	0.7	0.9	**0.79**
0.9	0.5	0.1	0.5	0.9	**0.1**
0.4	0.8	0.5	0.4	0.8	**0.7**
0.2	0.3	0.5	0.2	0.3	**0.6**

Table 5.3

probor(max(A,B),C) Examples

A	B	C	min(A,B)	max(A,B)	probor(max(A,B),C)
0.6	0.5	0.3	0.5	0.6	**0.72**
0.6	0.5	0.9	0.5	0.6	**0.96**
0.6	0.3	0.3	0.3	0.6	**0.72**
0.6	0.3	0.5	0.3	0.6	**0.8**
0.6	0.6	0.4	0.6	0.6	**0.76**
0.6	0.2	0.6	0.2	0.6	**0.84**
0.3	0.9	0.1	0.3	0.9	**0.91**
0.3	0.9	0.9	0.3	0.9	**0.99**
0.3	0.9	0.5	0.3	0.9	**0.95**
0.3	0.1	0.9	0.1	0.3	**0.93**
0.9	0.7	0.3	0.7	0.9	**0.93**
0.9	0.5	0.1	0.5	0.9	**0.91**
0.4	0.8	0.5	0.4	0.8	**0.9**
0.2	0.3	0.5	0.2	0.3	**0.65**

Considering the above tables, can notice the following:

- **min(max(A,B),C):** does not take in account the interestingness.
- **probor(min(A,B),C):** takes the less value between rule quality and rule interpretability but may ignore any of the values, and also takes in account the interestingness.
- **probor(max(A,B),C):** takes the highest value between rule quality and rule interpretability but will not ignore any of the values, and also takes in account the interestingness. When a rule has a good interestingness value, it’s fitness value will arise.

The **probor(min(A,B),C)** has the best effect on the fitness function, so this research adopted this function as a fitness function for the GA.

The pseudocode for 3-objective fitness computation algorithm is displayed in Figure 5.5.

```
Algorithm 5.3: Fitness Compution based Composition Method
1: For each chromosome in the population do
   1.1 Compute A= Rule’s Quality
   1.2 Compute B= Rule’s Interpretability
   1.3 Compute C= Rule’s Interestingness
   1.4 Compute min(A,B)
   1.5 Compute fitness = probor([min(A,B);C])
2:End
end-algorithm
```

Figure 5.5. Pseudocode of Fitness Compution based Composition Method

Step 5: Selection

The selection pattern is developed with care to make sure that better chromosomes are selected as parents. Thus, the best chromosomes remain in place.

The selection operator, tournament selection, obtains two individuals needed for generating four offspring. The better individuals among the set of offspring replace the two inferior individuals in the population.

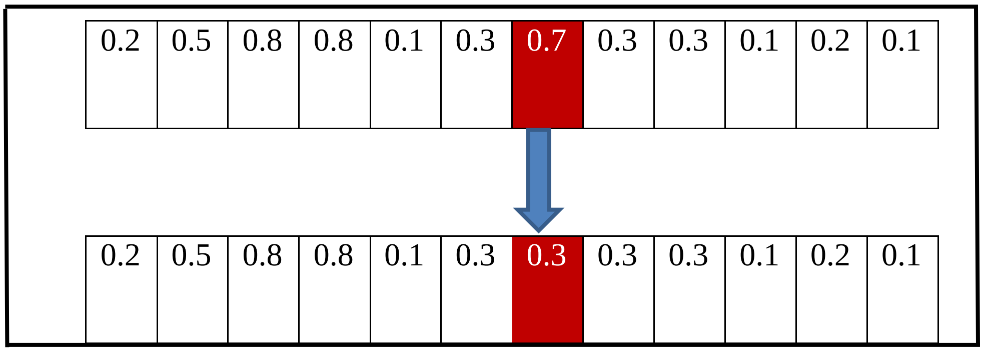
Step 6: Crossover

Crossover is defined as the genetic operation that creates two new chromosomes from two parent chromosomes (of the previous generation) by trading some of the parameters or genes. A uniform crossover is used in this study because it allows the offspring chromosomes to look for all possibilities of re-combining the different genes in the parents. The value of crossover probability is 1.

Step 7: Mutation

The mutation stage selects a number n of genes for changing, in this algorithm is 1. These bits are randomly selected with a mutation probability of 0.1. The selected gene value will be changed using a complementary operation to achieve diversity. This process will make sure that the value of the gene is retained in the membership range [0,1] and can be changed into another membership set.

As an example, if the number of the membership sets = 2 (low, high), having 6-inputs, the chromosome will be of length = 12. If the gene's value is 0.7 (high), it will become 0.3 (1 - 0.7); changing from high to low as shown in Figure 5.6.



0.2	0.5	0.8	0.8	0.1	0.3	0.7	0.3	0.3	0.1	0.2	0.1
0.2	0.5	0.8	0.8	0.1	0.3	0.3	0.3	0.3	0.1	0.2	0.1

Figure 5.6. Complementary Mutation

The weakness of the complementary mutation is when the gene's value is equal to medium range of membership [0,1], resulting in no modification.

Step 8: Module of Example Elimination - The Concept of Covering

In this study, as noted by several researchers (Gonzblez & Pérez, 1999; Ishida et al., 2009), the algorithm looks for the best rule on each iteration and takes away all the examples that are covered by said rule from the data set. The process is then repeated using the remaining examples. The process is continued till all the examples have been covered or when some stop criterion has been reached. Through this process, a new rule is discovered in every iteration.

Step 9: Termination Condition

The Genetic Algorithm (GA) concludes by returning the population's best rule if one of these following criteria is verified:

- The number of iterations is bigger than a fixed limit;
- The examples are all covered.

The final number of rules will be decided by the ANFIS depending on the predictive accuracy of network for the AFR prediction after applying the rules generated from the GA-FBC algorithm.

5.2.2 ANFIS Modelling

The ANFIS can refine the fuzzy ‘if-then’ rules and MFs towards describing a complex nonlinear system’s input/output behaviour. Because of their representation of data, which is compact and computationally-efficient, Sugeno type FISs are considered more suitable in constructing fuzzy models in practical applications. What has been used here is a zero-order Takagi-Sugeno fuzzy system. Many times, a singleton spike is totally sufficient towards satisfying the needs of a given problem.

In this study, ANFIS is selected as a control strategy because of its simple structure, taking advantage of both the fuzzy logic and adaptive NNs. It has also been known to be more advantageous than the pure fuzzy paradigm, where, in order to make adjustments on the bounds of the MFs to tune the system, the need for a human operator is removed. The pseudocode of ANFIS is in Figure 5.7.

Algorithm 5.4: ANFIS

```
1: Given n candidate inputs, a subset of " k " inputs is selected as an input
   to the ANFIS for training.
2: Normalized train and test data.
3: The ANFIS model is constructed.
4: ANFIS model is trained by training data.
5: Evaluation with test data.
end-algorithm
```

Figure 5.7. Pseudocode of ANFIS

Towards achieving a low training error for the engine AFR prediction, ANFIS used the training data set as well as the fuzzy rules constructed by the GA-based FBC.

5.3 Summary

The methodology, which is described previously and applied in this chapter, comprises three phases. Initially, it shows the behaviour of the data collection and selection. Next, it clarifies the algorithm formulation steps to enable feature selection and rule extraction. Lastly, it demonstrates the verification of the model’s effectiveness.

CHAPTER SIX
EXPERIMENTAL RESULTS AND EVALUATION

6.1 Introduction

This chapter is divided into four parts. This first part discusses the gathering of the data. The second part addresses the steps in constructing the proposed algorithm to perform the required target. Thereafter the experimental results will be clarified. The last part will discuss the evaluation of the proposed algorithm with other algorithms used in the same field. A working example is provided to explain the sequence of the algorithm.

6.2 Data Gathering

As a step for gathering the data and also as a preprocessing step for the validation of the group of data used, an ANFIS was used to train each combination of factors for one epoch to display the most influential factors with the lowest testing and training errors as the pre-processing procedure. It is illustrated in Figure 6.1 that different combinations that include all these factors resulted in close results that have the lowest testing and training errors. The checking and training errors are comparable, which means that there is no over fitting and that all of the factors have roughly the same effect on the AFR.

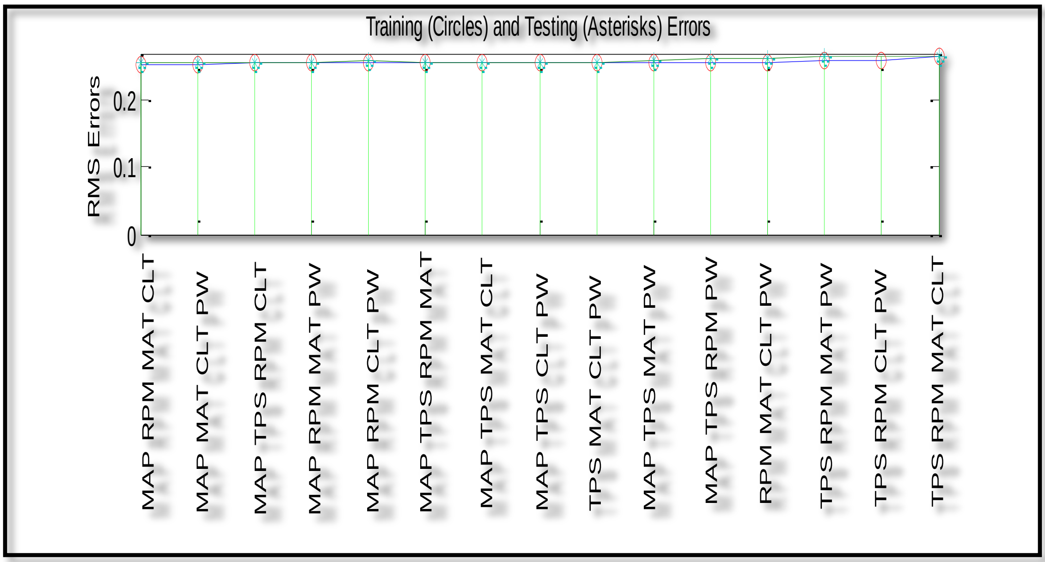


Figure 6.1. Most Influence Factors

The data distribution for the Data Set1, Data Set2, and Data Set3 are illustrated in Figure 6.2, 6.3, 6.4 respectively.

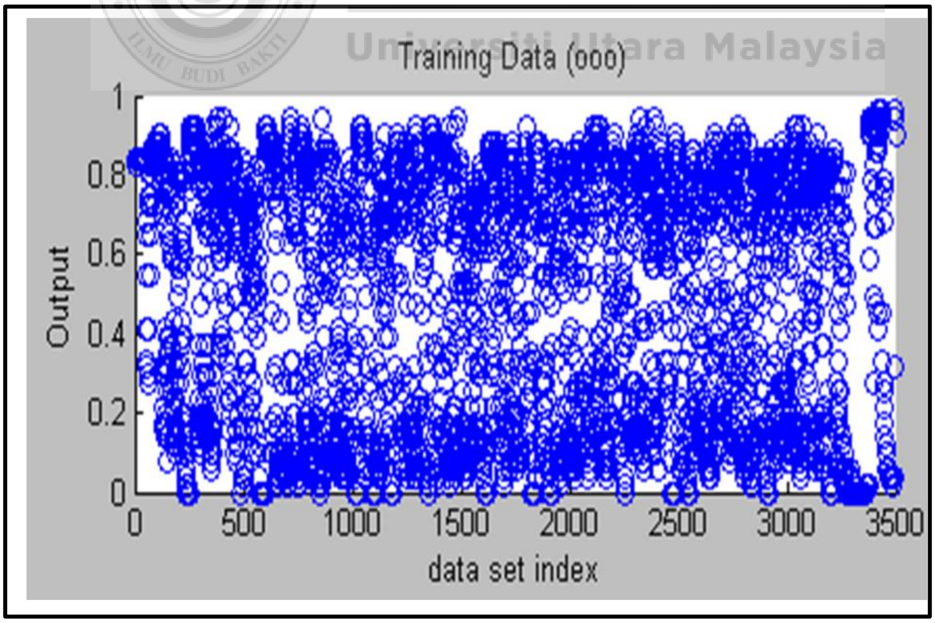


Figure 6.2. Data Distribution of Data Set1

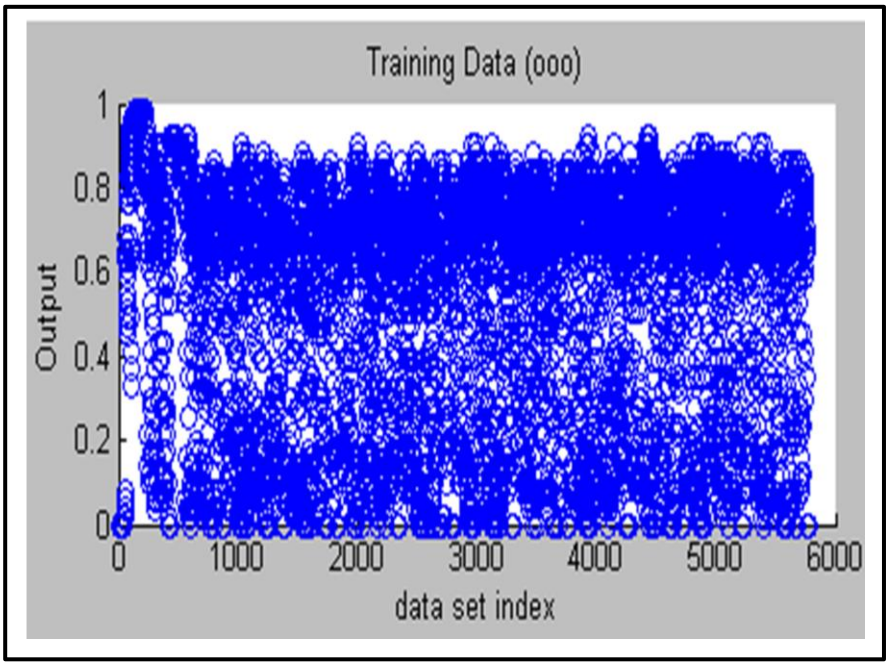


Figure 6.3. Data Distribution of Data Set2

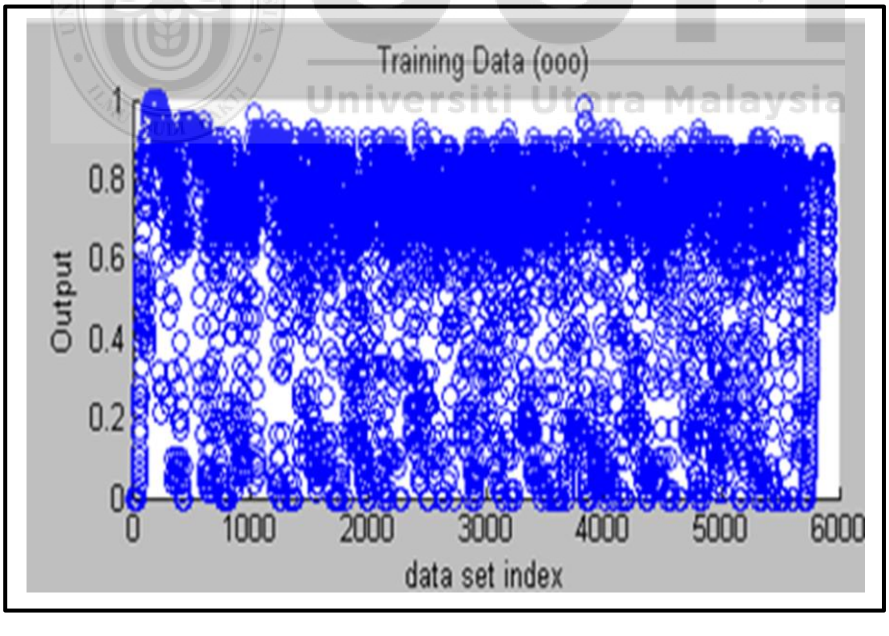


Figure 6.4. Data Distribution of Data Set3

6.3 Algorithm Construction

To build a FRBS consisting of *n* rules each having *m* inputs as shown in Figure 6.5, the feature selection and rule extraction is applied at the same time. Noticing that, the number of inputs is not fixed to a specific number is variable *m*.

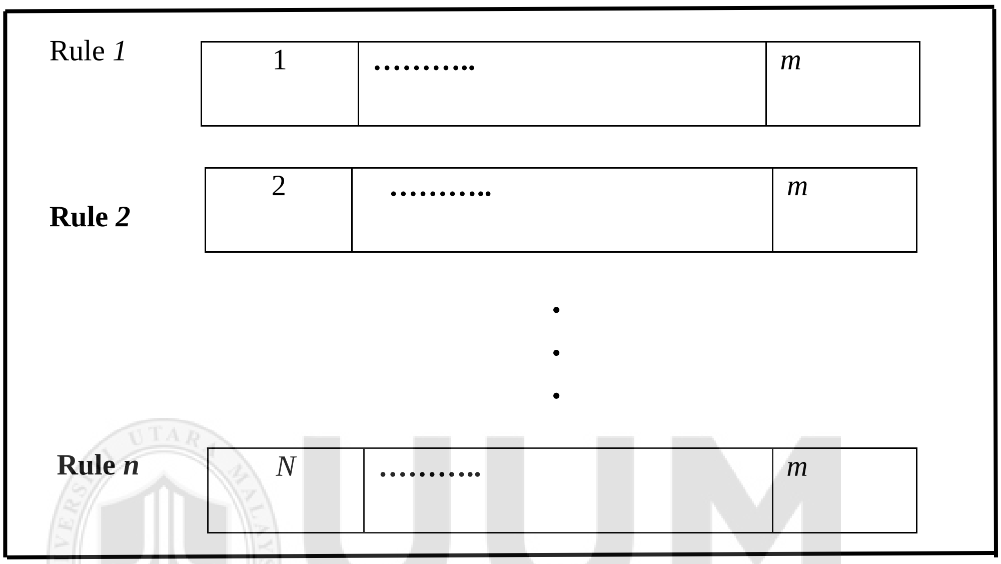


Figure 6.5. FRBS of n rules

The construction of the proposed algorithm involves two stages:

1. Rule based FIS extraction, and
2. ANFIS modeling for the prediction of the AFR.

A hybrid GA-ANFIS algorithm is used to achieve the two stages.

6.3.1 Genetic Algorithm for Optimization

In this work, a new GA-based method is implemented to generate satisfying fuzzy rules for the control a nonlinear system. The following sections explain the steps done to reach this target.

6.3.1.1 Initial designing

The first objective of this work, the initialization of the fuzzy sets of the rules is applied using FCM algorithm, given the initial values of the membership, clarifying the number of clusters for each fuzzy set. The subtractive clustering is used here to determine the number of clusters while the FCM is doing the clustering, taking only the initial values given by this algorithm including the membership function ranges for each cluster.

For example, taking the Data Set1 from AFR data set, containing of six factors and one output, the minimization of the objective function as in equation (2.4) reached the value zero in two iterations.

```
Iteration count = 1, obj. fcn = 0.000000
Iteration count = 2, obj. fcn = 0.000000
```

For Data Set2,

```
Iteration count = 1, obj. fcn = 0.001419
Iteration count = 2, obj. fcn = 0.000000
```

For Data Set3,

```
Iteration count = 1, obj. fcn = 0.015112
Iteration count = 2, obj. fcn = 0.000000
Iteration count = 3, obj. fcn = 0.000000
Iteration count = 1, obj. fcn = 0.000002
Iteration count = 2, obj. fcn = 0.000000
```

The number of clusters are set to 17, 9, 8 by the FCM for each data set respectively, given the memberships' ranges for each cluster. The ranges for inputs and output of each data set, are illustrated in table 6.1, 6.2, 6.3 respectively.

Table 6.1

Data Set1 Input and Output range

Input / Output	Min	Max
Input1	0.413	0.425
Input2	1.8349	92.6606
Input3	0	63
Input4	30	34.4
Input5	34.4	95
Input6	0.9	7.9
Output	0	0.961

Table 6.2

Data Set2 Input and Output range

Input / Output	Min	Max
Input1	0	1.025
Input2	2.7523	88.9908
Input3	0	62
Input4	7.2	33.3
Input5	29	96
Input6	0.6	6
Output	0	0.98

Table 6.3

Data Set3 Input and Output range

Input / Output	Min	Max
Input1	0.163	1.05
Input2	0	141.28
Input3	0	80
Input4	26.7	33.3
Input5	29	98
Input6	0.6	6
Output	0	1

In Figures 6.6, 6.7, 6.8 displays the membership sets values for Data Set1, Data Set2, and Data Set3 respectively.

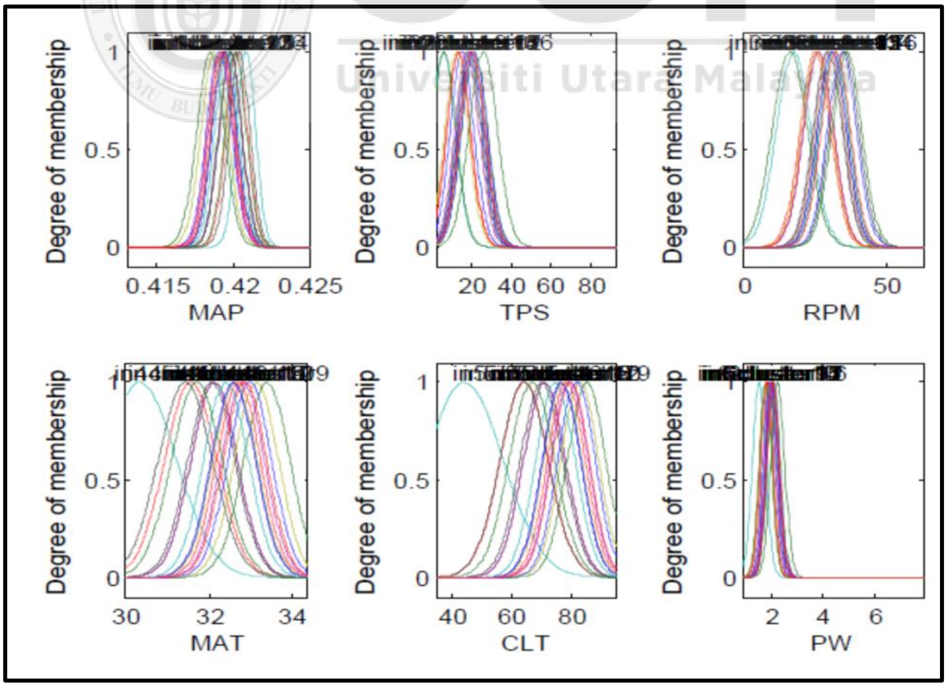


Figure 6.6. Data Set1 Membership Sets

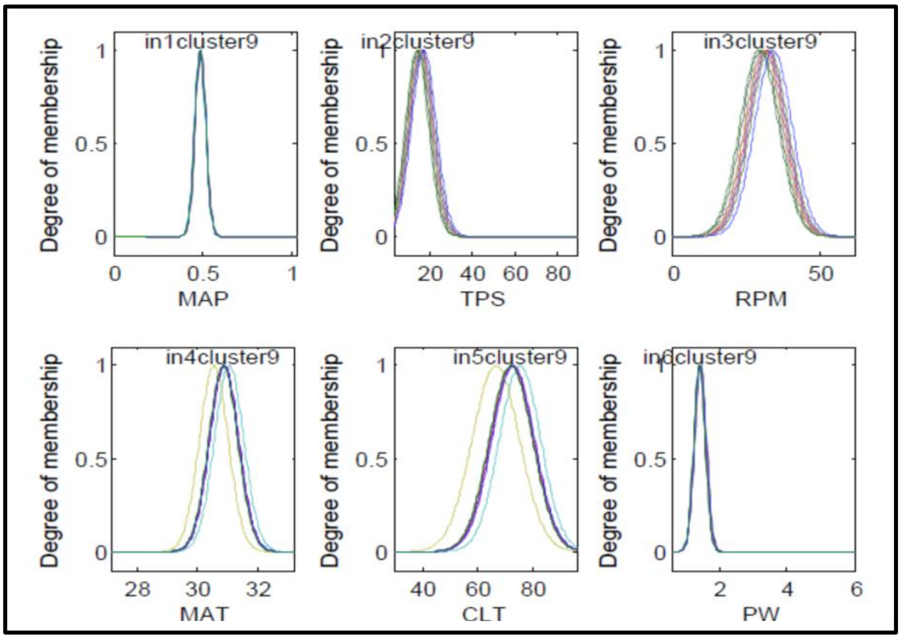


Figure 6.7. Data Set2 Membership sets

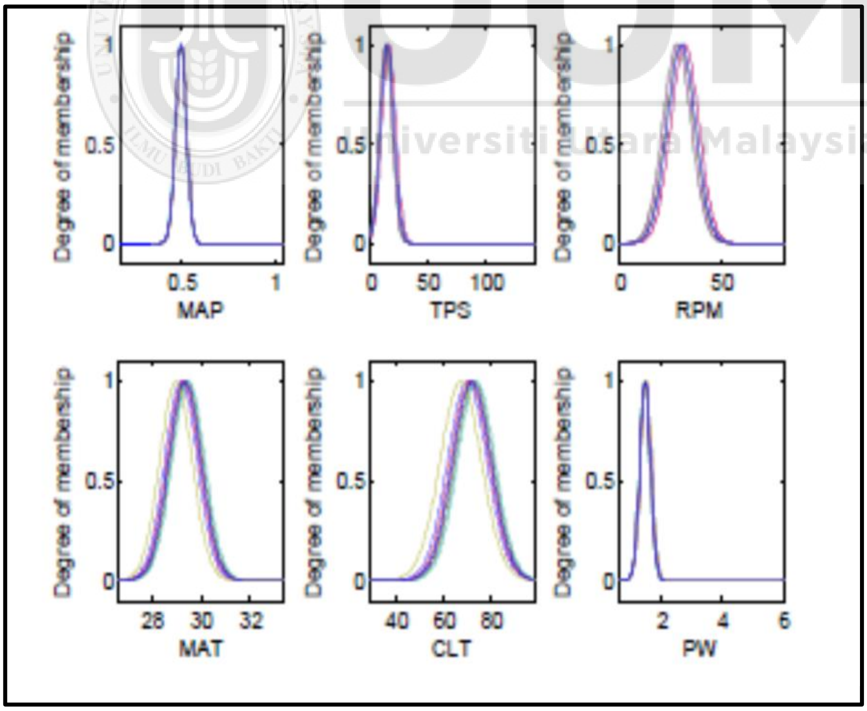


Figure 6.8. Data Set3 Membership sets

6.3.1.2 Rule Extraction and Feature Selection Model

In order to manage the learning algorithm to reach not only the best rule but the best selection of features also, a GA based FBC is used to apply the feature selection and rule extraction at the same level.

For the beginning of GA, we do not need to initialize the rules' number, the positions of the premise and the consequent fuzzy sets. The only thing need to do, is the setting of the length and the structure of the chromosome. With the specific structure of the chromosome, the crossover and special mutation operation and the appropriate fitness function, GA will produce the fuzzy rule base naturally which has small number of rules, arranges the adequate placement of the premise's fuzzy sets and choose the proper location of the consequent singletons.

Using the IRL approach as in Figure 6.2, each population contains a variable number of rules but produces only one rule, which should be the best rule depending of the fitness function. The number of the final rules of the FRBS is decided by the ANFIS depending on the predictive accuracy leading in a kind of cooperative rules rather than evaluating each rule by itself that is done already in the GA learning stage.

There are a number of steps need to be followed to apply the evolutionary algorithm as explained in the following sections.

Step1: Encoding

As the aim of this work, is automatically extracting the rules from the experimental data. The GA takes its values of genes from the data sets directly. But while we are using fuzzy logic as a way of dealing with uncertainty of the sensors producing the data, the encoding of the individual of GA is applied using the membership function for each fuzzy set. As the inputs are represented by more than one fuzzy set, the FCM decides the number of fuzzy sets where each cluster stands for a fuzzy set that are overlapped by the fuzzy nature. So each input can be represented by *n* of membership grades produced by applying the Gaussian membership function. Depending on equation (2.1), need to find the center *c* and standard deviation *δ* of the clusters.

$$c =(b-a)/2 \qquad (6.1)$$

$$\delta = ((((b-c)^\wedge 2)+((a-c)^\wedge 2))/2)^2 \qquad (6.2)$$

Where, *a* and *b* are the membership range of the fuzzy set.

For example, if we have the following inputs from the Data Set3 which is collected from the sensors at the same time:

Input1= 0.492, Input2 = 0.9174, Input3 = 15, Input4 = 32.8, Input5 = 96, Input6=1.3.

As noticed from table 6.3, different inputs have different ranges. Converting the real data value to the membership grade value gives a benefit of the ability in dealing with different forms of numbers with different ranges (as in real data) that can in this method be represented in one range [0,1].

The variables are encoded using continuous values in the range of [0,1], by assuming the number of fuzzy sets is 8 (from Data set 3), as Figure 6.9. Data Set3 is chosen for the simplicity, encoded by only 8 clusters.

Input 1	0	0.5812	0.6741	0	0.539	0	0	0.5456
Input 2	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Input 3	**0.901**	**0.9956**	**0.998**	**0.9713**	**0.7835**	**0.9944**	**0.9487**	**0.677**
Input 4	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Input 5	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Input 6	**0.561**	**0.6799**	**0.8439**	**0.5606**	**0.5241**	**0.6119**	**0.559**	**0.5004**

Figure 6.9. Chromosome Encoding

The chromosome will be of length 48 gene. As noticed from Figure 6.6, in Input1 the features 1,4,6, and 7 are not included for this input. Also to that, Input 2, Input 4, and Input 5 are their values are zero meaning that these inputs are not included in this rule, as each chromosome represents a rule.

The output is classified into the cluster with the highest degree of membership as in Figure 6.10. It is not included in the chromosome.

```
Algorithm 6.1: Pseudocode for Output Classification
1: Count membership grade matrix U
2: Find the highest membership for each output
for i=1to length of output
   maxU=U(1,i);
    for j=1to number of clusters
      if maxU < U(j,i)
          maxU = U(j,i);
          class1(i)=j;
      end
    end
  end
3:end algorithm
```

Figure 6.10. Pseudocode for Output Classification

For example, five values from output matrix. Each output has eight clusters.

For output values matrix = [0.8040 0.8430 0.8820 0.9020 0.2550].

Membership grade (U) for output1 = [0.1268 0.1250 0.1356 0.1185 **0.1401** 0.1183 0.1099 0.1253].

Membership grade (U) for output2 = [0.1269 0.1256 0.1328 0.1203 **0.1353** 0.1201 0.1128 0.1258].

Membership grade (U) for output3 = [0.1268 0.1257 0.1312 0.1213 **0.1330** 0.1212 0.1146 0.1259].

Membership grade (U) for output4 = [0.1267 0.1257 0.1306 0.1217 **0.1321** 0.1215 0.1153 0.1259].

Membership grade (U) for output5 = [0.1230 0.1235 0.1215 0.1260 0.1211 0.1262 **0.1349** 0.1234].

Each output value will be classified with the highest value from U matrix.

Output class = [5   5   5   5   7].

The first four values have near values so they are classified to the same class.

Step2: Consequent Calculation

BC is employed in the prediction of the consequent class of data instances in classification problems, here we utilize this type of classifier to specify the consequent class of rules. This leads to an accurate, fast and maybe optimal method which accelerates rule generation methods. Here the BC, uses the MAP model that maximizes the posterior probability where the best class is with the maximum posterior probability as in equation (5.2).

Another goodness of the BC is its ability in dealing with the collection of data from different sources as engine sensors in this work. It can apply the multi-data sensor fusion technology, especially in the case of the work for calculating the consequent of a rule which depends on the states of the all the antecedent considered in the same rule, taking in account the nonlinear interaction between the variables. Making it possible, to reach to an accurate rule consequent. In this work, we used a FBC for this need. Depending on Equation (5.2), the rule consequent is computed according to the steps:

1. The probability is represented by the membership grades encoded in the chromosome. Due to the fact that membership values for different classes and inputs are not normalized, it is possible that probabilities do not sum to one. Therefore the actual probabilities must be normalized depending on the equation (5.3).

This can be illustrated through the last example, as shown in Figure (6.6):

For example, for the second membership grade for Input 1, its probability is:

$$\frac{0.5812}{0+0.5812+0.6741+0+0.539+0+0+0.5456} = \frac{0.5812}{2.3399} = 0.2483$$

For the third membership grade for Input 1, its probability is:

$$\frac{0.6741}{2.3399} = 0.2880$$

For the fifth membership grade for Input 1, its probability is:

$$\frac{0.539}{2.3399} = \ 0.2303$$

For the eighth membership grade for Input 1, its probability is:

$$\frac{0.5456}{2.3399} = \ 0.2331$$

And so on for the other inputs as shown in Figure 6.11.

Input 1	0	0.2483	0.288	0	0.2303	0	0	0.2331
Input 2	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Input 3	**0.1239**	**0.1369**	**0.1372**	**0.1336**	**0.1077**	**0.1367**	**0.1305**	**0.0931**
Input 4	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Input 5	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Input 6	**0.1158**	**0.1404**	**0.1743**	**0.1158**	**0.1082**	**0.1264**	**0.1154**	**0.1033**

Figure 6.11. Probability Calculation for Chromosome

2. Calculating the probability of the class, P(C$_j$). Which is equal to, how many times each class is exists in the output divided by the sum of all classes. In this work, the classes of the output will be as the same number of clusters of the input. Also the algorithm will not fix the class for the rule but will leave the FBC decide which one is the best class.
3. Calculating the likelihood or conditional probability of the Bayes rule, $P(ai = v)\backslash Cj)$. It can be calculated by the same way as P(C$_j$). In this work,

the conditional probability is calculated using MI, not depending on the frequency. Depending on Equation (5.4), we calculate the conditional probability. In this work, MI will be used to select the features and chose the best class that fits those features by giving more information about the interaction between the class and feature and also participate in the relation between the features. As the Bayesian have a tendency to put all patterns of the smaller class to error which represents that no information is obtained from classifiers (Hu, 2014).

This approach applies the concept of class-dependent feature selection method which is likely to be more computationally expensive than other conventional feature selection methods. However, the extra computational cost may be worthwhile in certain applications where improvements of accuracy or reduction of data dimensionality are very important and meaningful especially in real world applications.

Calculating the conditional probability with MI, requires first finding the entropies that are estimated directly from the data samples that are represented here by the membership grade that already have calculated its probability.

For example, for 6 inputs each with 8 clusters. Each input will have 8 terms ($v_1 \ldots . v_8$).

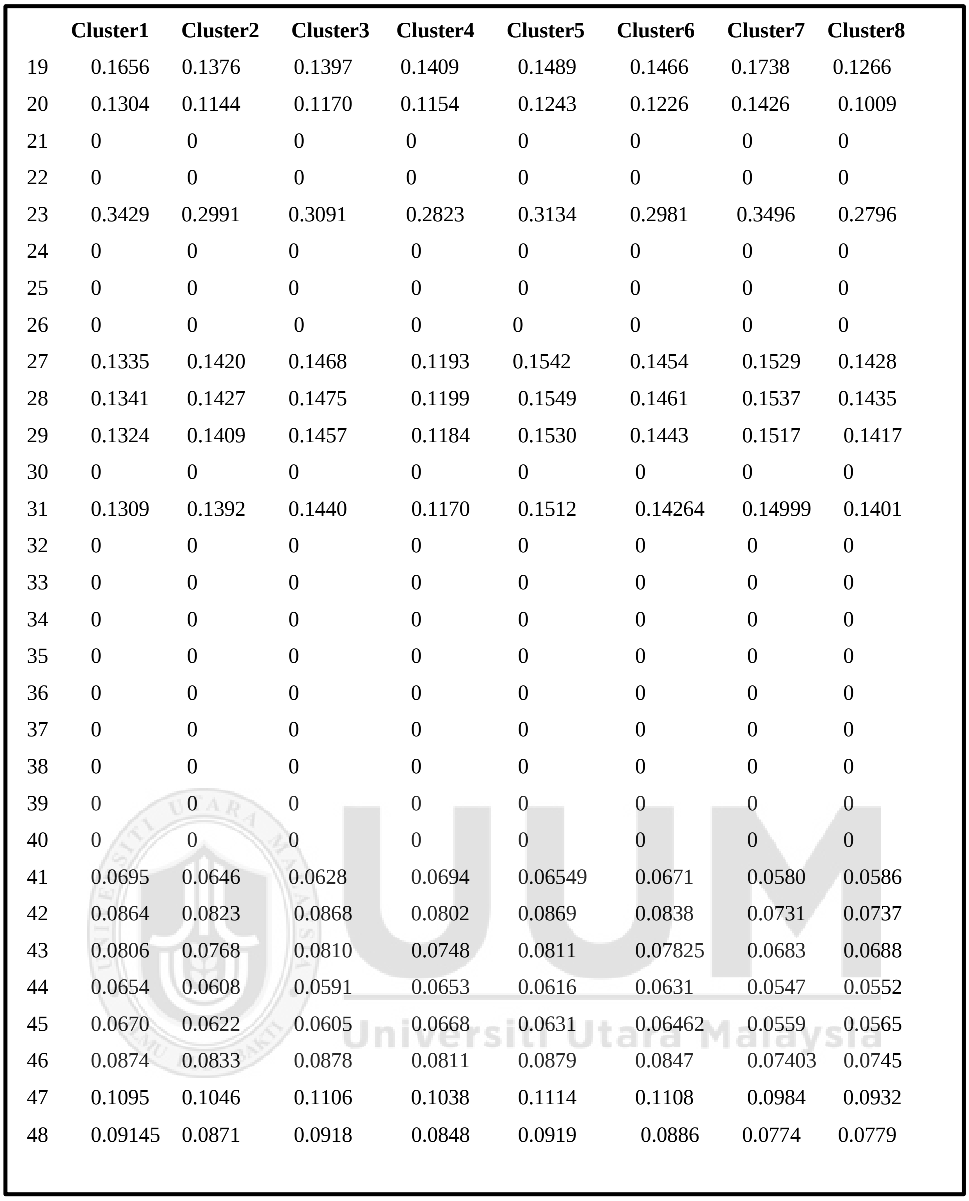
For class1 (C_1) and term1 or feature1 (v_1) for input1:

$$P(\text{input } 1 = v_1 \backslash C_1) = \frac{\mathrm{P(v1\backslash\ C1){*}log(P((\ v1)\backslash C1)}}{\mathrm{P(v1){*}P(C1)}}$$

Each term here represents a membership grade. As illustrated in Figure (6.12).

	Cluster1	**Cluster2**	**Cluster3**	**Cluster4**	**Cluster5**	**Cluster6**	**Cluster7**	**Cluster8**
1	0.09068	0.08610	0.07896	0.07923	0.08689	0.0829	0.07564	0.07336
2	0.07971	0.07431	0.06744	0.0727	0.07438	0.0646	0.0702	0.0656
3	0.04487	0.02860	0.0247	0.0324	0.0285	0.0297	0.0372	0.0317
4	0.0381	0.0200	0.0213	0.0263	0.02271	0.0245	0.0282	0.0228
5	0.0505	0.0379	0.0358	0.0372	0.0361	0.034 1	0.0447	0.0370
6	0.0508	0.0382	0.0360	0.0374	0.03638	0.0343	0.0450	0.0373
7	0.1088	0.1073	0.1029	0.1070	0.1108	0.1073	0.0985	0.0955
8	0.0792	0.0739	0.0670	0.0723	0.0739	0.06427	0.0698	0.0652
9	0	0	0	0	0	0	0	0
10	0.0218	0.0044	0.0046	0.0059	0.0048	0.0079	0.0134	0.0042
11	0.0308	0.0266	0.0191	0.0305	0.0195	0.0212	0.0395	0.0290
12	0.0191	0.0080	0.0071	0.0087	0.0055	0.0084	0.0226	0.0113
13	0.0167	0.0029	0.0039	0.0049	0.0030	0.0055	0.0082	0.0028
14	0.0189	0.0079	0.0070	0.0086	0.0055	0.0083	0.0223	0.0112
15	0.0531	0.0627	0.0652	0.0549	0.0612	0.0529	0.0642	0.0542
16	0236	0.0052	0.0047	0.0060	0.0058	0.0084	0.0166	0.0068
17	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0

Figure 6.12. Results of Likelihood Probability



	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6	Cluster7	Cluster8
19	0.1656	0.1376	0.1397	0.1409	0.1489	0.1466	0.1738	0.1266
20	0.1304	0.1144	0.1170	0.1154	0.1243	0.1226	0.1426	0.1009
21	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0
23	0.3429	0.2991	0.3091	0.2823	0.3134	0.2981	0.3496	0.2796
24	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0
27	0.1335	0.1420	0.1468	0.1193	0.1542	0.1454	0.1529	0.1428
28	0.1341	0.1427	0.1475	0.1199	0.1549	0.1461	0.1537	0.1435
29	0.1324	0.1409	0.1457	0.1184	0.1530	0.1443	0.1517	0.1417
30	0	0	0	0	0	0	0	0
31	0.1309	0.1392	0.1440	0.1170	0.1512	0.14264	0.14999	0.1401
32	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0
41	0.0695	0.0646	0.0628	0.0694	0.06549	0.0671	0.0580	0.0586
42	0.0864	0.0823	0.0868	0.0802	0.0869	0.0838	0.0731	0.0737
43	0.0806	0.0768	0.0810	0.0748	0.0811	0.07825	0.0683	0.0688
44	0.0654	0.0608	0.0591	0.0653	0.0616	0.0631	0.0547	0.0552
45	0.0670	0.0622	0.0605	0.0668	0.0631	0.06462	0.0559	0.0565
46	0.0874	0.0833	0.0878	0.0811	0.0879	0.0847	0.07403	0.0745
47	0.1095	0.1046	0.1106	0.1038	0.1114	0.1108	0.0984	0.0932
48	0.09145	0.0871	0.0918	0.0848	0.0919	0.0886	0.0774	0.0779

Figure 6.12. Continued

For class (C1) with all terms of input1 ($v_1 \ldots v_8$):

P(input1\C1)=

$$\frac{\mathrm{P(v1\backslash C1)*log(P((v1)\backslash C1)}}{\mathrm{P(v1)*P(C1)}}+\frac{\mathrm{P(v2\backslash C1)*log(P((v2)\backslash C1)}}{\mathrm{P(v2)*P(C1)}}+\cdots+\frac{\mathrm{P(v8\backslash C1)*log(P((v8)\backslash C1)}}{\mathrm{P(v8)*P(C1)}}$$

After implementing the likelihood probability for the other classes (C2…C8), we can get the output class (consequent) with the highest posterior probability depending on Equation 5.2.

Posterior probability= [0.0060 0.0016 0.0003 0.0001 0.0009 0.0006   0.0071 0.00019].

So the consequent for the first chromosome (rule) is 7. And so on for the other chromosomes.

Step3: Fitness Function

The fitness function for each rule is evaluated using Equation (3.1), (3.2), (3.3) for the rule quality, rule interpretability, and rule interestingness respectively.

In this work, for a multi-objective fitness function, we used a composition function to aggregate the objectives without the need for weights. The objectives are conflicting in nature such as the accuracy and interpretability. Trying to avoid this problem, we took the inverse of the interpretability value in Equation (3.2). The function tries to make a trade-off between the objectives.

Table (5.1), (5.2), (5.3) displays examples of different composition functions. In this research we used the probor(min) to apply the fitness function.

We can notice that depending on the first part from of the function, the second part gives an importance to the interestingness which we require. The fitness of a rule with low interestingness, though it has high quality or interpretability, it will be weaken depending on the degree of interestingness of the rule. But if it has high interestingness, the fitness of the rule will be supported. Probor function is the most that supports this theory.

6.3.2 ANFIS Modeling

After applying a GA based FBC with a population= 500, Tournament crossover was used with Probability= 1. The Complementary Mutation was implemented with probability= 0.1.

The FRBS generated will be used by ANFIS. A zero-order Takagi-Sugeno fuzzy system is used with 6-inputs and one output for the prediction of AFR. The ANFIS will help in the membership tuning and in selecting the number of rules generated by GA-FBC depending on the predictive accuracy of the network.

6.4 Experiment and Results

This chapter tested the proposed algorithm on 3 AFR data sets as illustrated in Chapter Four, using a *ten*-fold cross-validation technique (Kohavi, 1995). Each data set is randomly split into ten approximately equally sized subsets. First, the experiment takes one subset as the testing set, while the rest subsets become the training set. After the proposed algorithm trained the training set, the proposed algorithm evaluated the discovered rules against the test subset. The experiment repeats this processes ten times, producing ten individual sets of performance statistics, such as predictive accuracy, number of rules and number of terms in the rules. Finally, the experiment averages these performance statistics.

The model will be compared to other learning algorithms for NFS such as fuzzy clustering using FCM, grid partition algorithm. It also will be compared to the original GA-FBC depending on the frequency not on MI taking in account the complexity of the FRBS, number of rules and number of terms in each rule.

Comparing the behavior of the proposed model output with the output of the real model (experimental data), showing how close does the model represent the data (prediction error).

Figure 6.13, 6.14, 6.15 presents examples of the fuzzy rules resulted from applying the Original algorithm were the FBC depends on the frequency for Data Set1, Data Set2, and Data Set3 respectively.

Rule1: If (in1 is in1cluster2) and (in2 is in2cluster6) and (in3 is in3cluster6) and (in4 is in4cluster11) and (in5 is in5cluster10) and (in6 is in6cluster6) then (out1 is out1cluster4)
Rule2: If (in2 is in2cluster12) and (in3 is in3cluster12) and (in4 is in4cluster11) and (in5 is in5cluster11) and (in6 is in6cluster6) then (out1 is out1cluster4)
Rule3: If (in2 is in2cluster16) and (in3 is in3cluster8) and (in4 is in4cluster2) and (in5 is in5cluster2) and (in6 is in6cluster4) then (out1 is out1cluster3)
Rule4: If (in1 is in1cluster4) and (in2 is in2cluster4) and (in3 is in3cluster4) and (in4 is in4cluster2) and (in5 is in5cluster2) then (out1 is out1cluster3)
Rule5: If (in1 is in1cluster3) and (in2 is in2cluster2) and (in3 is in3cluster1) and (in4 is in4cluster9) and (in5 is in5cluster1) then (out1 is out1cluster5)
Rule6: If (in2 is in2cluster15) and (in3 is in3cluster8) and (in4 is in4cluster2) and (in5 is in5cluster2) and (in6 is in6cluster4) then (out1 is out1cluster5)
Rule7: If (in1 is in1cluster17) and (in3 is in3cluster17) and (in4 is in4cluster8) and (in5 is in5cluster8) then (out1 is out1cluster6)
Rule8: If (in2 is in2cluster2) and (in3 is in3cluster2) and (in4 is in4cluster16) and (in5 is in5cluster1) and (in6 is in6cluster12) then (out1 is out1cluster17)
Rule9: If (in2 is in2cluster12) and (in3 is in3cluster12) and (in4 is in4cluster5) and (in5 is in5cluster15) then (out1 is out1cluster6)
Rule10: If (in1 is in1cluster5) and (in2 is in2cluster15) and (in3 is in3cluster15) and (in4 is in4cluster8) and (in5 is in5cluster8) then (out1 is out1cluster8)
Rule11: If (in1 is in1cluster8) and (in2 is in2cluster8) and (in3 is in3cluster8) and (in4 is in4cluster15) and (in5 is in5cluster15) then (out1 is out1cluster17)
Rule12: If (in1 is in1cluster13) and (in2 is in2cluster1) and (in3 is in3cluster1) and (in4 is in4cluster3) and (in5 is in5cluster3) and (in6 is in6cluster1) then (out1 is out1cluster15)
Rule13: If (in1 is in1cluster3) and (in2 is in2cluster7) and (in3 is in3cluster1) and (in4 is in4cluster2) and (in5 is in5cluster2) and (in6 is in6cluster7) then (out1 is out1cluster8)

Figure 6.13. Fuzzy Rules of Original Algorithm for Data Set1

Rule1: If (in1 is in1cluster9) and (in3 is in3cluster9) and (in4 is in4cluster2) and (in5 is in5cluster4) then (out1 is out1cluster4)
Rule2: If (in1 is in1cluster8) and (in2 is in2cluster8) and (in4 is in4cluster8) and (in5 is in5cluster8) then (out1 is out1cluster3)
Rule3: If (in1 is in1cluster8) and (in2 is in2cluster8) and (in3 is in3cluster1) and (in4 is in4cluster8) and (in5 is in5cluster8) then (out1 is out1cluster6)
Rule4: If (in1 is in1cluster2) and (in2 is in2cluster2) and (in3 is in3cluster3) and (in4 is in4cluster8) and (in5 is in5cluster4) then (out1 is out1cluster7)
Rule5: If (in1 is in1cluster8) and (in2 is in2cluster8) and (in4 is in4cluster4) and (in5 is in5cluster2) and (in6 is in6cluster1) then (out1 is out1cluster6)
Rule6: If (in1 is in1cluster6) and (in3 is in3cluster8) and (in4 is in4cluster1) and (in5 is in5cluster3) and (in6 is in6cluster9) then (out1 is out1cluster3)
Rule7: If (in1 is in1cluster2) and (in3 is in3cluster7) and (in4 is in4cluster5) and (in5 is in5cluster9) then (out1 is out1cluster4)
Rule8: If (in1 is in1cluster5) and (in2 is in2cluster5) and (in3 is in3cluster5) and (in4 is in4cluster5) and (in6 is in6cluster4) then (out1 is out1cluster8)
Rule9: If (in1 is in1cluster8) and (in2 is in2cluster1) and (in3 is in3cluster2) and (in4 is in4cluster6) and (in5 is in5cluster8) then (out1 is out1cluster9)
Rule10: If (in1 is in1cluster6) and (in3 is in3cluster3) and (in4 is in4cluster7) and (in5 is in5cluster3) then (out1 is out1cluster5)
Rule11: If (in1 is in1cluster1) and (in2 is in2cluster9) and (in3 is in3cluster9) and (in4 is in4cluster6) and (in5 is in5cluster1) and (in6 is in6cluster8) then (out1 is out1cluster2)
Rule12: 12. If (in1 is in1cluster8) and (in2 is in2cluster8) and (in3 is in3cluster8) and (in4 is in4cluster5) and (in5 is in5cluster5) and (in6 is in6cluster8) then (out1 is out1cluster1)

Figure 6.14. Fuzzy Rules of Original Algorithm for Data Set2

Rule1: If (input1 is mf4) and (input3 is mf4) and (input4 is mf6) and (input5 is mf8) then (output1 is mf2)
Rule2: If (input1 is mf6) and (input2 is mf4) and (input3 is mf2) and (input4 is mf7) and (input5 is mf7) and (input6 is mf4) then (output1 is mf5)
Rule3: If (input1 is mf3) and (input2 is mf3) and (input3 is mf3) and (input4 is mf8) and (input5 is mf1) then (output1 is mf5)
Rule4: If (input1 is mf5) and (input3 is mf5) and (input4 is mf1) and (input5 is mf4) then (output1 is mf4)
Rule5: If (input3 is mf5) and (input4 is mf3) and (input5 is mf6) then (output1 is mf3)
Rule6: If (input1 is mf4) and (input3 is mf5) and (input4 is mf2) and (input5 is mf7) then (output1 is mf6)
Rule7: 7. If (input1 is mf7) and (input2 is mf7) and (input3 is mf7) and (input4 is mf4) and (input5 is mf7) then (output1 is mf7)
Rule8: If (input1 is mf2) and (input2 is mf3) and (input3 is mf4) and (input4 is mf7) and (input5 is mf7) then (output1 is mf8)
Rule9: If (input1 is mf4) and (input2 is mf4) and (input4 is mf3) and (input5 is mf2) then (output1 is mf1)
Rule10: If (input1 is mf4) and (input4 is mf4) and (input5 is mf6) and (input6 is mf7) then (output1 is mf2)
Rule11: If (input1 is mf1) and (input2 is mf6) and (input3 is mf6) and (input4 is mf8) and (input5 is mf5) and (input6 is mf3) then (output1 is mf8)

Figure 6.15. Fuzzy Rules of Original Algorithm for Data Set3

Figure 6.16, 6.17, 6.18 present examples of the fuzzy rules resulted from applying the Proposed algorithm on the Data Set1, Data Set2, and Data Set3 respectively.

Rule1: If (in1 is in1cluster4) and (in2 is in2cluster11) and (in6 is in6cluster15) then (out1 is out1cluster10)

Rule2: If (in1 is in1cluster4) and (in2 is in2cluster9) and (in6 is in6cluster15) then (out1 is out1cluster12)

Rule3: If (in2 is in2cluster9) and (in4 is in4cluster6) and (in5 is in5cluster17) then (out1 is out1cluster12)

Rule4: If (in4 is in4cluster14) and (in5 is in5cluster13) then (out1 is out1cluster5)

Rule5: If (in1 is in1cluster17) and (in2 is in2cluster10) and (in3 is in3cluster16) then (out1 is out1cluster4)

Rule6: If (in1 is in1cluster2) and (in2 is in2cluster16) and (in4 is in4cluster14) and (in5 is in5cluster5) then (out1 is out1cluster4)

Rule7: If (in1 is in1cluster3) and (in2 is in2cluster4) and (in6 is in6cluster2) then (out1 is out1cluster13)

Rule8: If (in1 is in1cluster5) and (in3 is in3cluster15) and (in4 is in4cluster8) and (in5 is in5cluster8) then (out1 is out1cluster8)

Rule9: If (in2 is in2cluster5) and (in4 is in4cluster7) and (in6 is in6cluster5) then (out1 is out1cluster7)

Rule10: If (in1 is in1cluster17) and (in2 is in2cluster1) then (out1 is out1cluster7)

Rule11: If (in4 is in4cluster7) and (in5 is in5cluster5) and (in6 is in6cluster2) then (out1 is out1cluster16)

Rule12: If (in1 is in1cluster5) and (in2 is in2cluster12) and (in6 is in6cluster3) then (out1 is out1cluster11)

Rule13: If (in1 is in1cluster11) and (in3 is in3cluster3) and (in4 is in4cluster1) and (in5 is in5cluster7) and (in6 is in6cluster9) then (out1 is out1cluster2)

Rule14: If (in1 is in1cluster2) and (in5 is in5cluster13) and (in6 is in6cluster11) then (out1 is out1cluster9)

Rule15: If (in2 is in2cluster5) and (in4 is in4cluster1) and (in5 is in5cluster11) then (out1 is out1cluster7)

Rule16: If (in2 is in2cluster17) and (in4 is in4cluster1) and (in5 is in5cluster2) then (out1 is out1cluster14)

Rule17: If (in1 is in1cluster11) and (in2 is in2cluster17) and (in3 is in3cluster2) then (out1 is out1cluster10)

Rule18: If (in1 is in1cluster15) and (in2 is in2cluster14) and (in3 is in3cluster1) then (out1 is out1cluster8)

Figure 6.16. Fuzzy Rules of Proposed Algorithm for Data Set1

Rule1: If (in1 is in1cluster9) and (in2 is in2cluster9) and (in3 is in3cluster3) then (out1 is out1cluster4)

Rule2: If (in2 is in2cluster9) and (in3 is in3cluster1) and (in6 is in6cluster3) then (out1 is out1cluster4)

Rule3: If (in3 is in3cluster3) and (in4 is in4cluster5) and (in5 is in5cluster7) then (out1 is out1cluster5)

Rule4: If (in1 is in1cluster2) and (in4 is in4cluster2) and (in5 is in5cluster9) then (out1 is out1cluster9)

Rule5: If (in1 is in1cluster8) and (in4 is in4cluster4) and (in5 is in5cluster2) and (in6 is in6cluster1) then (out1 is out1cluster6)

Rule6: If (in2 is in2cluster6) and (in3 is in3cluster6) and (in6 is in6cluster2) then (out1 is out1cluster3)

Rule7: If (in3 is in3cluster1) and (in4 is in4cluster8) and (in6 is in6cluster9) then (out1 is out1cluster8)

Rule8: If (in2 is in2cluster3) and (in3 is in3cluster3) and (in5 is in5cluster4) then (out1 is out1cluster2)

Rule9: If (in3 is in3cluster2) and (in4 is in4cluster6) then (out1 is out1cluster8) (1)

Rule10: If (in1 is in1cluster5) and (in2 is in2cluster5) and (in4 is in4cluster4) and (in5 is in5cluster2) then (out1 is out1cluster7)

Rule11: If (in1 is in1cluster9) and (in3 is in3cluster1) and (in4 is in4cluster2) and (in5 is in5cluster7) then (out1 is out1cluster6)

Figure 6.17. Fuzzy Rules of Proposed Algorithm for Data Set2

Rule1: If (input1 is mf1) and (input2 is mf1) then (output1 is mf4)

Rule2: If (input3 is mf5) and (input4 is mf3) and (input5 is mf6) then (output1 is mf3)

Rule3: If (input3 is mf2) and (input4 is mf4) and (input5 is mf5) then (output1 is mf6)

Rule4: If (input3 is mf6) and (input4 is mf5) and (input5 is mf1) then (output1 is mf1)

Rule5: If (input1 is mf4) and (input3 is mf4) and (input4 is mf7) and (input5 is mf6) then (output1 is mf7)

Rule6: If (input2 is mf4) and (input3 is mf4) and (input4 is mf7) and (input5 is mf8) then (output1 is mf7)

Rule7: If (input3 is mf6) and (input4 is mf4) and (input5 is mf3) then (output1 is mf4)

Rule8: If (input1 is mf5) and (input2 is mf5) then (output1 is mf2)

Rule9: If (input1 is mf3) and (input3 is mf6) and (input4 is mf4) and (input5 is mf6) then (output1 is mf1)

Rule10: If (input3 is mf1) and (input4 is mf1) and (input5 is mf5) then (output1 is mf2)

Rule11: If (input2 is mf4) and (input3 is mf1) and (input4 is mf1) then (output1 is mf7)

Rule12: If (input1 is mf5) and (input4 is mf7) and (input5 is mf2) then (output1 is mf6)

Rule13: If (input3 is mf1) and (input4 is mf5) and (input5 is mf7) then (output1 is mf5)

Figure 6.18. Fuzzy Rules of Proposed Algorithm for Data Set3

The Average number of rules and terms for each of the Original algorithm and Proposed algorithm is presented in table 6.4 and 6.5.

Table 6.4

Average number of rules of Original algorithm and Proposed Algorithm

Data Sets	Original Algorithm	Proposed Algorithm
Data Set1	13	18
Data Set2	12	14
Data Set3	12	13

Table 6.5

Average number of Terms of Original algorithm and Proposed Algorithm

Data Sets	Original Algorithm	Proposed Algorithm
Data Set1	68	56
Data Set2	55	42
Data Set3	52	40

Noticing from the above tables, the average number of rules for the Proposed algorithm is higher than the Original algorithm. But at the same time, the average number of terms in each rule is lesser in the model using the proposed algorithm which depends on the mutual information that provides more information about the interactions among the features and between the features and classes, Figure 6.19, 6.20 illustrates that.

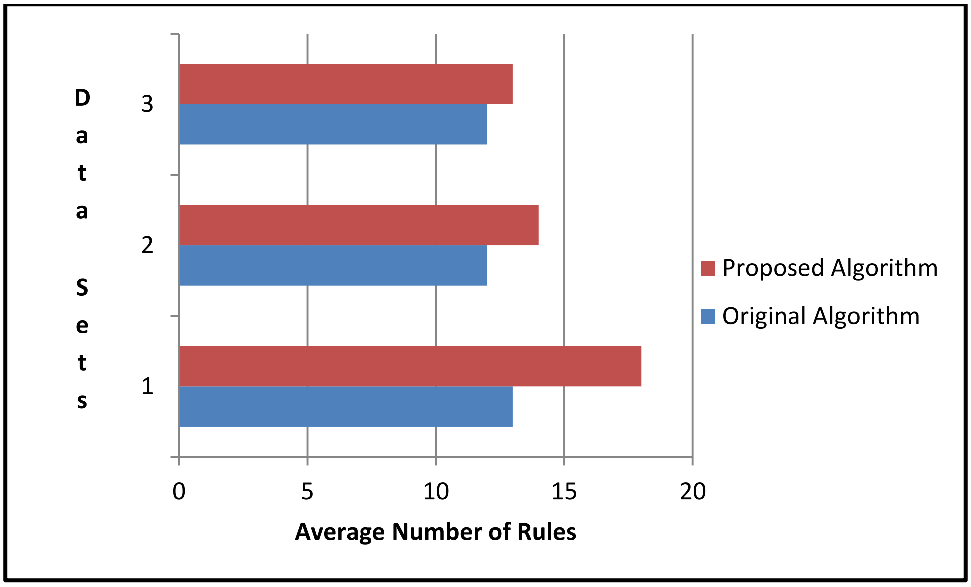


Figure 6.19. Comparison of Average Number of Rules between Proposed and Original Algorithms

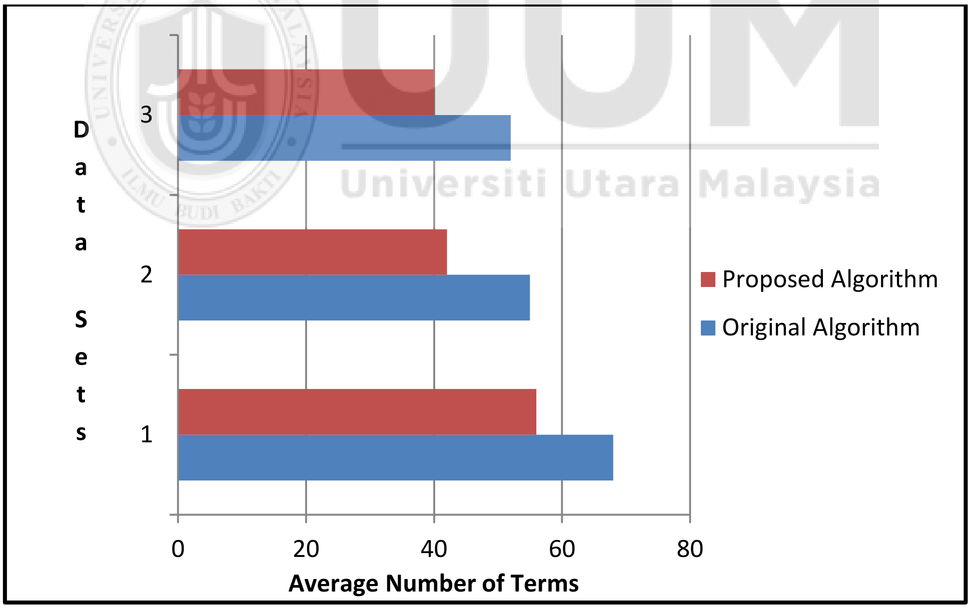


Figure 6.20. Comparison of Average Number of Terms between Proposed and Original Algorithms

For the average predictive accuracy, Table 6.6 illustrates the average predictive accuracy of ANFIS with the Original algorithm and Proposed algorithm, using fuzzy sets of 17-cluster for Data Set1, 9-cluster for Data Set2, and 8-cluster for Data Set3.

Table 6.6

Average Predictive Accuracy of ANFIS with Original Algorithm and Proposed Algorithm

Data Sets	Original Algorithm (%)	Proposed Algorithm (%)
Data Set1	66	70
Data Set2	75	75
Data Set3	76	78

The comparison of the average predictive for the two algorithms is presented in Figure 6.21.

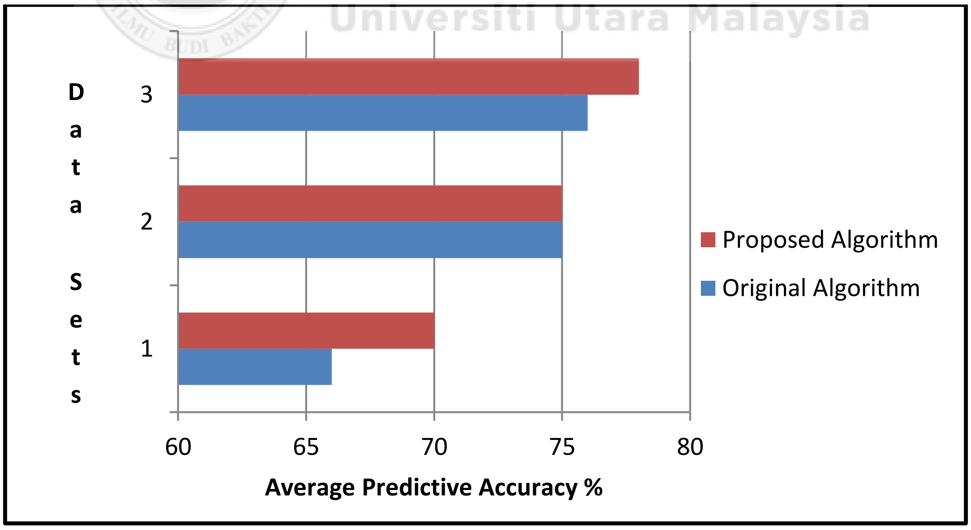


Figure 6.21. Comparison of Average Predictive Accuracy between Proposed and Original Algorithms

As illustrated in table 6.6 and figure 6.21, the average predictive accuracy of the algorithms are convergent, especially for Data Set2. For the complexity including he number of rules and terms as illustrated in table 6.4 and 6.5, and Figure 6.19 and 6.20, the Proposed algorithm has lesser terms in the rules than the other which means its interpretability is higher, whereas the Original algorithm has lessr number of rules than the other.

For the comparison of the Proposed algorithm to other learning algorithms for NFS such as fuzzy clustering using FCM, grid partition algorithm. Figure 6.22, 6.23, 6.24 presents examples of fuzzy rules generated by the FCM algorithm for Data Set1, Data Set2, and Data Set3 respectively.

Rule1: If (in1 is in1cluster1) and (in2 is in2cluster1) and (in3 is in3cluster1) and (in4 is in4cluster1) and (in5 is in5cluster1) and (in6 is in6cluster1) then (out1 is out1cluster1)
Rule2: If (in1 is in1cluster2) and (in2 is in2cluster2) and (in3 is in3cluster2) and (in4 is in4cluster2) and (in5 is in5cluster2) and (in6 is in6cluster2) then (out1 is out1cluster2)
Rule3: If (in1 is in1cluster3) and (in2 is in2cluster3) and (in3 is in3cluster3) and (in4 is in4cluster3) and (in5 is in5cluster3) and (in6 is in6cluster3) then (out1 is out1cluster3)
Rule4: If (in1 is in1cluster4) and (in2 is in2cluster4) and (in3 is in3cluster4) and (in4 is in4cluster4) and (in5 is in5cluster4) and (in6 is in6cluster4) then (out1 is out1cluster4)
Rule5: If (in1 is in1cluster5) and (in2 is in2cluster5) and (in3 is in3cluster5) and (in4 is in4cluster5) and (in5 is in5cluster5) and (in6 is in6cluster5) then (out1 is out1cluster5)
Rule6: If (in1 is in1cluster6) and (in2 is in2cluster6) and (in3 is in3cluster6) and (in4 is in4cluster6) and (in5 is in5cluster6) and (in6 is in6cluster6) then (out1 is out1cluster6)
Rule7: If (in1 is in1cluster7) and (in2 is in2cluster7) and (in3 is in3cluster7) and (in4 is in4cluster7) and (in5 is in5cluster7) and (in6 is in6cluster7) then (out1 is out1cluster7)
Rule8: If (in1 is in1cluster8) and (in2 is in2cluster8) and (in3 is in3cluster8) and (in4 is in4cluster8) and (in5 is in5cluster8) and (in6 is in6cluster8) then (out1 is out1cluster8)
Rule9: If (in1 is in1cluster9) and (in2 is in2cluster9) and (in3 is in3cluster9) and (in4 is in4cluster9) and (in5 is in5cluster9) and (in6 is in6cluster9) then (out1 is out1cluster9)
Rule10: If (in1 is in1cluster10) and (in2 is in2cluster10) and (in3 is in3cluster10) and (in4 is in4cluster10) and (in5 is in5cluster10) and (in6 is in6cluster10) then (out1 is out1cluster10)
Rule11: If (in1 is in1cluster11) and (in2 is in2cluster11) and (in3 is in3cluster11) and (in4 is in4cluster11) and (in5 is in5cluster11) and (in6 is in6cluster11) then (out1 is out1cluster11)
Rule12: If (in1 is in1cluster12) and (in2 is in2cluster12) and (in3 is in3cluster12) and (in4 is in4cluster12) and (in5 is in5cluster12) and (in6 is in6cluster12) then (out1 is out1cluster12)
Rule13: If (in1 is in1cluster13) and (in2 is in2cluster13) and (in3 is in3cluster13) and (in4 is in4cluster13) and (in5 is in5cluster13) and (in6 is in6cluster13) then (out1 is out1cluster13)
Rule14: If (in1 is in1cluster14) and (in2 is in2cluster14) and (in3 is in3cluster14) and (in4 is in4cluster14) and (in5 is in5cluster14) and (in6 is in6cluster14) then (out1 is out1cluster14)
Rule15: If (in1 is in1cluster15) and (in2 is in2cluster15) and (in3 is in3cluster15) and (in4 is in4cluster15) and (in5 is in5cluster15) and (in6 is in6cluster15) then (out1 is out1cluster15)
Rule16: If (in1 is in1cluster16) and (in2 is in2cluster16) and (in3 is in3cluster16) and (in4 is in4cluster16) and (in5 is in5cluster16) and (in6 is in6cluster16) then (out1 is out1cluster16)
Rule17: If (in1 is in1cluster17) and (in2 is in2cluster17) and (in3 is in3cluster17) and (in4 is in4cluster17) and (in5 is in5cluster17) and (in6 is in6cluster17) then (out1 is out1cluster17)

Figure 6.22. Fuzzy Rules of FCM Algorithm for Data Set1

Rule1: If (in1 is in1cluster1) and (in2 is in2cluster1) and (in3 is in3cluster1) and (in4 is in4cluster1) and (in5 is in5cluster1) and (in6 is in6cluster1) then (out1 is out1cluster1)
Rule2: If (in1 is in1cluster2) and (in2 is in2cluster2) and (in3 is in3cluster2) and (in4 is in4cluster2) and (in5 is in5cluster2) and (in6 is in6cluster2) then (out1 is out1cluster2)
Rule3: If (in1 is in1cluster3) and (in2 is in2cluster3) and (in3 is in3cluster3) and (in4 is in4cluster3) and (in5 is in5cluster3) and (in6 is in6cluster3) then (out1 is out1cluster3)
Rule4: If (in1 is in1cluster4) and (in2 is in2cluster4) and (in3 is in3cluster4) and (in4 is in4cluster4) and (in5 is in5cluster4) and (in6 is in6cluster4) then (out1 is out1cluster4)
Rule5: If (in1 is in1cluster5) and (in2 is in2cluster5) and (in3 is in3cluster5) and (in4 is in4cluster5) and (in5 is in5cluster5) and (in6 is in6cluster5) then (out1 is out1cluster5)
Rule6: If (in1 is in1cluster6) and (in2 is in2cluster6) and (in3 is in3cluster6) and (in4 is in4cluster6) and (in5 is in5cluster6) and (in6 is in6cluster6) then (out1 is out1cluster6)
Rule7: If (in1 is in1cluster7) and (in2 is in2cluster7) and (in3 is in3cluster7) and (in4 is in4cluster7) and (in5 is in5cluster7) and (in6 is in6cluster7) then (out1 is out1cluster7)
Rule8: If (in1 is in1cluster8) and (in2 is in2cluster8) and (in3 is in3cluster8) and (in4 is in4cluster8) and (in5 is in5cluster8) and (in6 is in6cluster8) then (out1 is out1cluster8)
Rule9: If (in1 is in1cluster9) and (in2 is in2cluster9) and (in3 is in3cluster9) and (in4 is in4cluster9) and (in5 is in5cluster9) and (in6 is in6cluster9) then (out1 is out1cluster9)

Figure 6.23. Fuzzy Rules of FCM Algorithm for Data Set2

Rule1: If (in1 is in1cluster1) and (in2 is in2cluster1) and (in3 is in3cluster1) and (in4 is in4cluster1) and (in5 is in5cluster1) and (in6 is in6cluster1) then (out1 is out1cluster1)
Rule2: If (in1 is in1cluster2) and (in2 is in2cluster2) and (in3 is in3cluster2) and (in4 is in4cluster2) and (in5 is in5cluster2) and (in6 is in6cluster2) then (out1 is out1cluster2)
Rule3: If (in1 is in1cluster3) and (in2 is in2cluster3) and (in3 is in3cluster3) and (in4 is in4cluster3) and (in5 is in5cluster3) and (in6 is in6cluster3) then (out1 is out1cluster3)
Rule4: If (in1 is in1cluster4) and (in2 is in2cluster4) and (in3 is in3cluster4) and (in4 is in4cluster4) and (in5 is in5cluster4) and (in6 is in6cluster4) then (out1 is out1cluster4)
Rule5: If (in1 is in1cluster5) and (in2 is in2cluster5) and (in3 is in3cluster5) and (in4 is in4cluster5) and (in5 is in5cluster5) and (in6 is in6cluster5) then (out1 is out1cluster5)
Rule6: If (in1 is in1cluster6) and (in2 is in2cluster6) and (in3 is in3cluster6) and (in4 is in4cluster6) and (in5 is in5cluster6) and (in6 is in6cluster6) then (out1 is out1cluster6)
Rule7: If (in1 is in1cluster7) and (in2 is in2cluster7) and (in3 is in3cluster7) and (in4 is in4cluster7) and (in5 is in5cluster7) and (in6 is in6cluster7) then (out1 is out1cluster7)
Rule8: If (in1 is in1cluster8) and (in2 is in2cluster8) and (in3 is in3cluster8) and (in4 is in4cluster8) and (in5 is in5cluster8) and (in6 is in6cluster8) then (out1 is out1cluster8)

Figure 6.24. Fuzzy Rules of FCM Algorithm for Data Set3

The FCM algorithm has an acceptable number of rules but has low interpretability, were all the features and terms are included in each rule.

For another method, Grid partitioning is a subjective approach, where the number of clusters into variable is segmented, is initially selected by the user. The number of rules is determined by multiplying the number of clusters of each input variable. This introduces a dimensionality problem, as the number of variables increases the number of rules also increases, as real world classification problems usually involve many attributes. When we use K antecedent fuzzy sets for each attribute of an n dimensional classification problem, the total number of possible fuzzy if-then rules is K^n, which is huge for a large value of n (Ishibuchi, Nakashima, & Morisawa, 1997).

The grid partition provides the ANFIS with the fuzzy rules and membership functions, which the number of both should be determined by the user previously.

Figure 6.25, 6.26, 6.27 presents examples of fuzzy rules generated by the Grid partition algorithm for Data Set1, Data Set2, and Data Set3 respectively.

Rule1: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf1) then (output is out1mf1)
Rule2: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf2) then (output is out1mf2)
Rule3: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf1) then (output is out1mf3)
Rule4: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf2) then (output is out1mf4)
Rule5: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf1) and (input6 is in6mf1) then (output is out1mf5)
Rule6: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf1) and (input6 is in6mf2) then (output is out1mf6)
Rule7: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf2) and (input6 is in6mf1) then (output is out1mf7)
Rule8: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf2) and (input6 is in6mf2) then (output is out1mf8)
Rule9: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf2) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf1) then (output is out1mf9)

•
•
•

Rule24137569: If (input1 is in1mf17) and (input2 is in2mf17) and (input3 is in3mf17) and (input4 is in4mf17) and (input5 is in5mf17) and (input6 is in6mf17) then (output is

Figure 6.25. Fuzzy Rules of Grid partition Algorithm for Data Set1

Rule1: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf1) then (output is out1mf1)
Rule2: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf2) then (output is out1mf2)
Rule3: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf1) then (output is out1mf3)
Rule4: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf2) then (output is out1mf4)
Rule5: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf1) and (input6 is in6mf1) then (output is out1mf5)
Rule6: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf1) and (input6 is in6mf2) then (output is out1mf6)
Rule7: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf2) and (input6 is in6mf1) then (output is out1mf7)
Rule8: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf2) and (input6 is in6mf2) then (output is out1mf8)
Rule9: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf2) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf1) then (output is out1mf9)

•

•

•

Rule531441: If (input1 is in1mf9) and (input2 is in2mf9) and (input3 is in3mf9) and (input4 is in4mf9) and (input5 is in5mf9) and (input6 is in6mf9) then (output is out1mf531441)

Figure 6.26. *Fuzzy Rules of Grid partition Algorithm for Data Set2*

Rule1: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf1) then (output is out1mf1)
Rule2: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf2) then (output is out1mf2)
Rule3: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf1) then (output is out1mf3)
Rule4: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf2) then (output is out1mf4)
Rule5: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf1) and (input6 is in6mf1) then (output is out1mf5)
Rule6: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf1) and (input6 is in6mf2) then (output is out1mf6)
Rule7: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf2) and (input6 is in6mf1) then (output is out1mf7)
Rule8: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf2) and (input6 is in6mf2) then (output is out1mf8)
Rule9: If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf2) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf1) then (output is out1mf9)

•

•

•

Rule262144: If (input1 is in1mf8) and (input2 is in2mf8) and (input3 is in3mf8) and (input4 is in4mf8) and (input5 is in5mf8) and (input6 is in6mf8) then (output is out1mf262144)

Figure 6.27. Fuzzy Rules of Grid partition Algorithm for Data Set3

The Grid partition algorithm takes in its rules all the possible conditions but at the expense of its simplicity and interpretability.

Table 6.7, 6.8 shows the average number of rules of the Proposed algorithm and each of the Grid partition and FCM algorithm respectively.

Table 6.7

Average Number of Rules of Grid Partition and Proposed Algorithm

Data Sets	Grid partition	Proposed Algorithm
Data Set1	24,137,569	18
Data Set2	531,441	14
Data Set3	262,144	13

Table 6.8

Average Number of Rules of FCM and Proposed Algorithm

Data Sets	FCM	Proposed Algorithm
Data Set1	17	18
Data Set2	9	14
Data Set3	8	13

The below Figure 6.28 shows a comparison between the four algorithms including the average number of rules for the 3 data sets. For a more clear comparison, the Figure 6.29 compares between the three algorithms without the Grid Partition Algorithm.

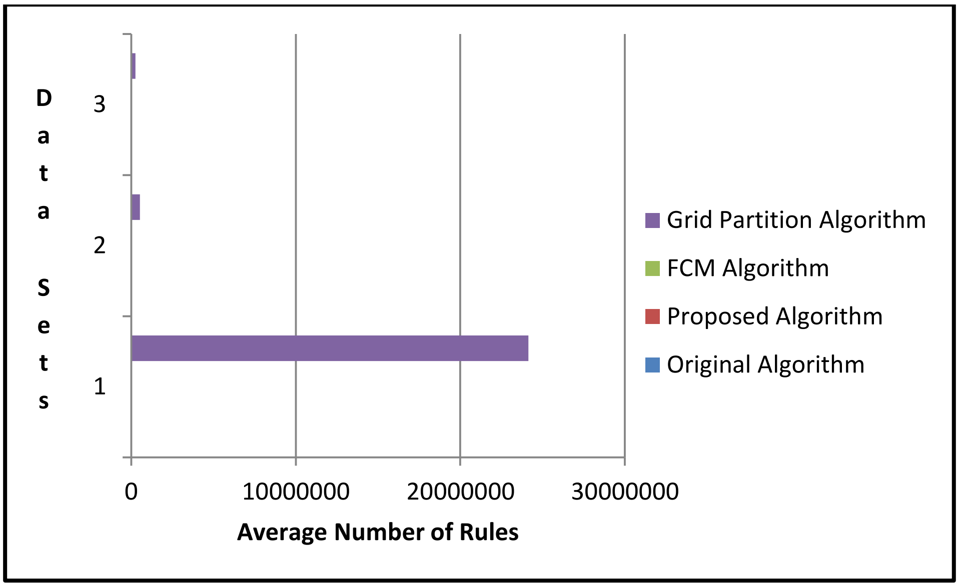


Figure 6.28. Comparison of Average Number of Rules for Four Algorithms

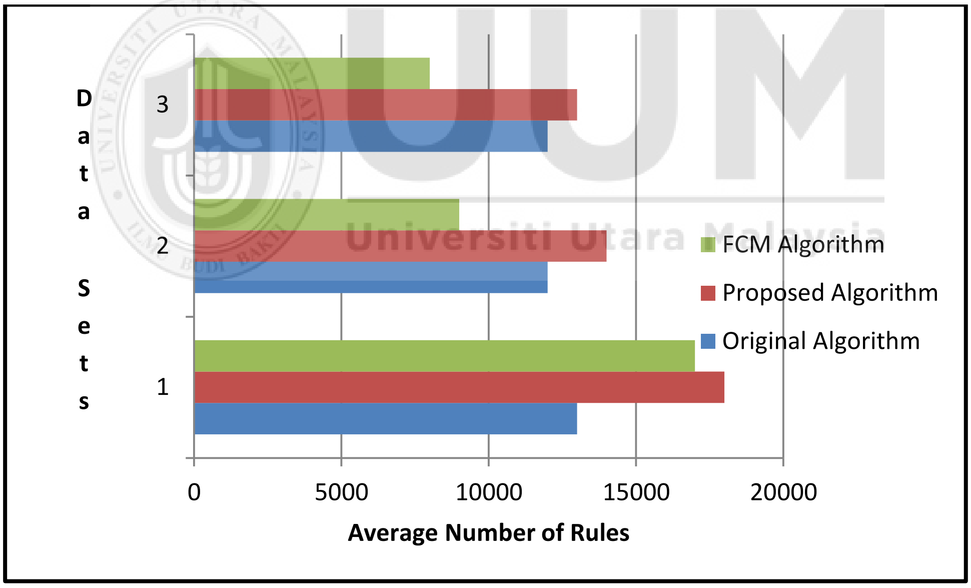


Figure 6.29. Comparison of Average Number of Rules for Three Algorithms

Table 6.9, 6.10 shows the average number of terms in the rules of the proposed algorithm and each of the Grid partition, and FCM algorithms respectively.

Table 6.9

Average Number of Terms of Grid partition and Proposed Algorithm

Data Sets	Grid partition	Proposed Algorithm
Data Set1	410,338,639	56
Data Set2	4,782,969	42
Data Set3	2,097,152	40

Table 6.10

Average Number of Terms of FCM and Proposed Algorithm

Data Sets	FCM	Proposed Algorithm
Data Set1	289	56
Data Set2	81	42
Data Set3	64	40

Figure 6.30 shows a comparison between the three algorithms including the average number of terms for the 3 data sets. For a more clear comparison, Figure 6.31 compares between the Proposed algorithm and the FCM algorithm.

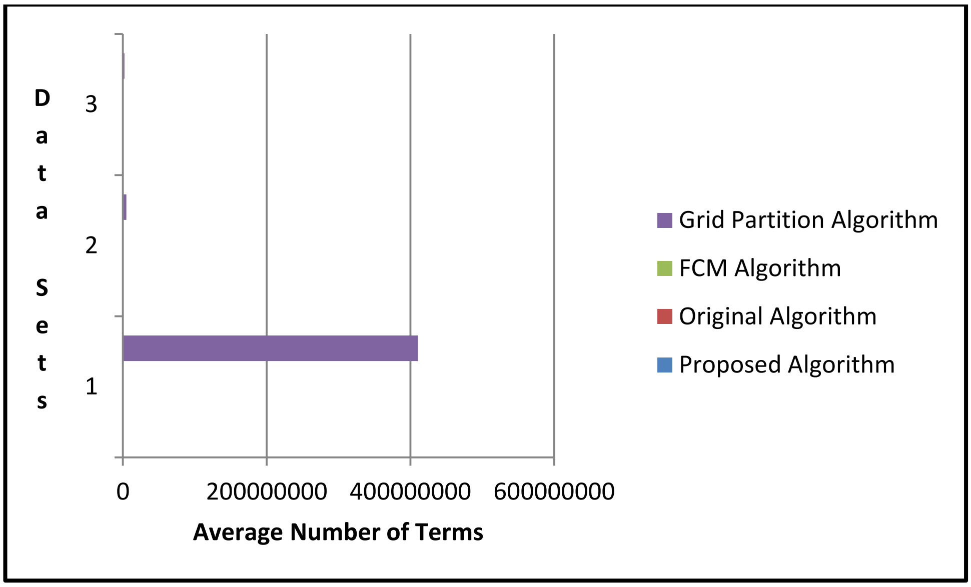


Figure 6.30. Comparison of Average Number of Terms for Four Algorithms

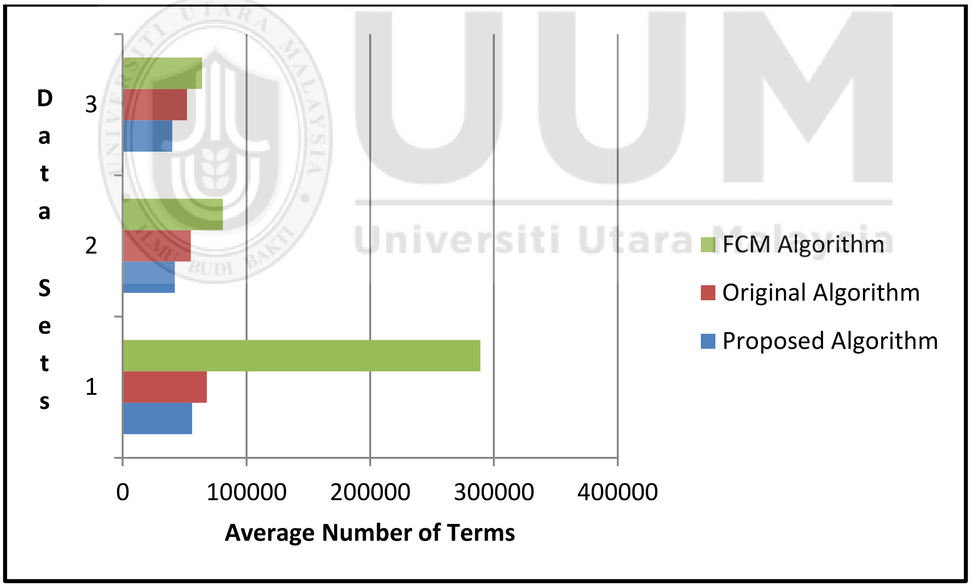


Figure 6.31. Comparison of Average Number of Terms for Three Algorithms

From the above tables and figures, we can notice that the proposed algorithm has number of rules higher than other algorithms except for the grid partition algorithm, but it has less attributes from all of them even though if not that difference except for grid partition algorithm.

Table 6.11 illustrates the average predictive accuracy of ANFIS with Grid partition and proposed algorithm, using fuzzy sets of 17-cluster for Data Set1, 9-cluster for Data Set2, and 8-cluster for Data Set3.

Table 6.11

Average Predictive Accuracy of ANFIS with Grid Partition and Proposed Algorithm

Data Sets	Grid Partition (%)	Proposed Algorithm (%)
Data set1	78	70
Data set2	80	75
Data set3	85	78

Table 6.12 illustrates the average predictive accuracy of ANFIS with FCM and proposed algorithm, using fuzzy sets of 17-cluster for data Set1, 9-cluster for data Set2, and 8-cluster for Data Set3.

Table 6.12

Average Predictive Accuracy of ANFIS with FCM and Proposed Algorithm

Data Sets	FCM (%)	Proposed Algorithm (%)
Data et1	69	70
Data set2	73	75
Data set3	74	78

Figure 6.32 presents the average predictive accuracy of the four algorithms for the three data sets.

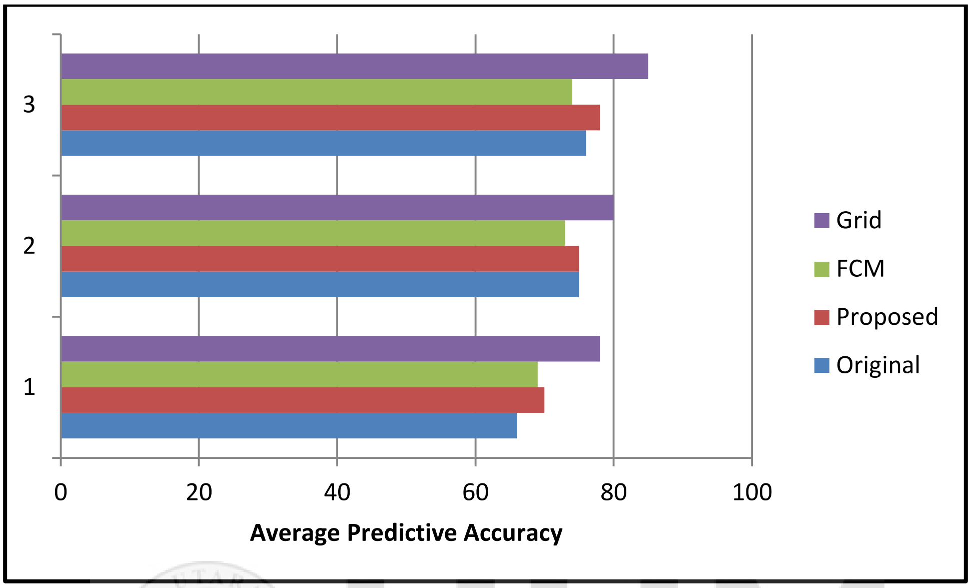


Figure 6.32. Comparison of Average Predictive Accuracy for Four Algorithms

According to the above tables, we notice that the average predictive accuracy of the proposed algorithm is higher than the FCM and Original algorithm, but less than the grid partition algorithm. The grid partition usually gives higher accuracy than other algorithms because it takes in its rules all the possible conditions but at the expense of its simplicity and interpretability where can be noticed from table (6.7) and table (6.9), the huge number of rules and terms.

It is a matter of trade-off between the accuracy and interpretability of the models.

6.5 Summary

This chapter had proposed a hybrid algorithm based on GA based FBC with MI to discover classification rules from data in order to build the FIS for the ANFIS. The GA acts as a local search algorithm in order to discover a good quality rule.

This chapter uses GA as the Learn-One-Rule function to extract rules from data sets. In order to improve the rule's quality, the proposed algorithm uses FBC with MI for terms selection while building the rule. The proposed algorithm used a simple heuristic that takes the relationship between terms into account, while selecting terms for growing a rule. Moreover, this chapter also proposed a fitness function to evaluate the rule from various criteria. Besides, the proposed algorithm will not fix the class before selecting terms in order to improve the rule's predictive accuracy.

This chapter compares the performance of the proposed algorithm to the original GA based FBC, fuzzy clustering with FCM, and grid partition algorithm concerning the predictive accuracy, and found to be competitive with them. Moreover, the experiments in this chapter found that the rules discovered by the proposed algorithm were simpler that the compared algorithms. Therefore, MI is able to help FBC produce simpler rules by aiding in the feature selection and rule extraction and performing it in a simpler way by applying simultaneously, even though the predictive accuracy is only at par with other algorithms compared with.

CHAPTER SEVEN

CONCLUSION AND PERSPECTIVES

7.1 General Discussion

This research workout on the limitation of constructing FRBS from human expert, employing the expert knowledge to describe variables that are the most influential, and utilising them in a few basic rules. The methodology was found unsuitable for dealing with intricate systems in cases where the expert knowledge is inadequate. It is difficult for an expert to formalise the interaction among many variables.

Generally, the main objective of the research is towards designing a feature selection and rule extraction algorithm that is based on a Genetic Algorithm-Based Approach using information theoretic properties. These important requirements should be satisfied by the new algorithm: one, reduction of the complexity of FRBS by reducing fuzzy rules as well as rule attributes, and two, increase of FBRS' predictive accuracy. In coming up with the new algorithm, these particular objectives should be reached:

- simultaneous application of feature selection and rule extraction;
- design of a FBC classifier-based MI;
- generation of a multi-objective evaluation function and the reduction of its complexity.

The evaluation of the new algorithm's strength was made through a verification of NFS using engine AFR prediction data.

This research study’s methodology has three phases:

- data gathering;
- formulation of algorithm for rule extraction and feature selection; and
- model evaluation.

7.2 Research Achievement

Summarised below are all the study objectives, which have been achieved successfully:

- Objective 1: To produce fuzzy rules from data without the need of expert.

  The use of multi-objective GFS for this study has been suggested. Executing this objective generated the search space for constructing the FRBS from sensor data. The results are duly presented in Chapter 6.
- Objective 2: To enhance the feature selection and rule extraction approach.

  A simultaneous application of the feature selection and rule extraction algorithms using Bayesian classifier-based MI was conducted and described in Chapter 5. The results are presented in detail in Chapter 6.
- Objective 3: To improve the evaluation function by reducing its intricacy.

  Using the Composition method, the evaluation function was developed successfully. By discarding the weights, its complexity was reduced. The algorithm is shown in Chapter 5 while the execution results are presented in Chapter 6.
- Objective 4: To assess the NFS -based generated fuzzy RB.

The final system summarised in the NFS underwent evaluation for its complexity and accuracy using the engine AFR ratio data. The results can be seen in Chapter 6.

7.3 Contributions

This research has three key contributions. First is, for the AFR ratio prediction problem, the automatic extraction of fuzzy rules from the sensor data without using the help of experts. This contributed in the prediction of a complex system (i.e. as automobile engines). This, in turn, generates a number of achievements, such as precise predictions which will help in controlling automobile emissions and in reducing one of the causes of pollution; minimisation of the risk of expensive engine damage and ensuring peak engine performance; and improvement of fuel efficiency and contributing in fuel economy.

The second contribution lies not only in finding an optimal rule for classification but in selecting a set of useful features which may provide the solution to the classification problem. For this, we offered an integrated mechanism for the purpose of simultaneously extracting fuzzy rules and selecting useful features.

The third contribution is towards building a minimal set of rules with a similarly minimal set of terms (i.e. fuzzified variables) towards classifying the observations. Decreasing the number of rules is intended to increase the FIS' generalisation ability, while, at the same time, keeping the FIS compact.

An additional contribution is the use of the complement mutation scheme with the uniform crossover; this seems to minimise the errors during a best-case scenario. This scheme maintains adequate genetic diversity within the population towards exploring the universe of discourse, while, at the same time, entirely utilising the local search so as to make a convergence on the optimal solution.

7.4 Limitation

The study has been undertaken in order to address the problems of accuracy and complexity. While the question of speed has not been addressed, this will be the focus of a future project.

7.5 Recommendations for Future Work

In this study's conclusion, several recommendations for future work are presented. Although this research has demonstrated that the proposed algorithm can reduce the intricacy and enhance the fuzzy rule-based system's predictive accuracy, further research needs to be conducted towards enhancing or supporting the proposed algorithm. Suggestions include the following:

- The use of the multi-objective cooperation-co-evolutionary techniques. These techniques are founded on the observation that dividing intricate tasks into subtasks may be helpful towards concentrating the search space exploration around useful niches and towards preventing loss of diversity in the population. Because the cooperative co-evolutionary approach designates an isolated species to each subtask, the division or splitting should take care of the existing dependencies that occur among the subtasks. Such dependencies are typically critical to fitness assessment. A problem is provided with complete solutions through coupling individuals from different species. Therefore, during the evolution, cooperation among the species is achieved through representatives that are used in assessing the fitness of complete solutions. At the algorithm’s conclusion, a set of FRBSs with different trade-offs between the different objectives is obtained by a combination of all possible RBs.

- Different partitioning of each input attribute to enhance the concept of interpretability. Using the same number of clusters for all the inputs will increase the dimensionality which increases the complexity and decreases the interpretability.

REFERENCES

Abadeh, M. S., Habibi, J., & Lucas, C. (2007). Intrusion detection using a fuzzy genetics-based learning algorithm. *Journal of Network and Computer Applications, 30*(1), 414-428.

Abraham, A. (2004). Intelligent systems: Architectures and perspectives. *arXiv preprint cs/0405009*.

Abraham, A. (2005). Adaptation of fuzzy inference system using neural learning *Fuzzy systems engineering* (pp. 53-83): Springer.

Alcala-Fdez, J., Alcala, R., & Herrera, F. (2011). A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. *Fuzzy Systems, IEEE Transactions on, 19*(5), 857-872.

Alcalá-Fdez, J., Herrera, F., Márquez, F., & Peregrín, A. (2007). Increasing fuzzy rules cooperation based on evolutionary adaptive inference systems. *International Journal of Intelligent Systems, 22*(9), 1035-1064.

Alcalá, R., Gacto, M. J., & Herrera, F. (2011). A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems. *Fuzzy Systems, IEEE Transactions on, 19*(4), 666-681.

Alsmadi, M. K. S., Omar, K. B., & Noah, S. A. (2009). Back propagation algorithm: the best algorithm among the multi-layer perceptron algorithm. *International Journal of Computer Science and Network Security, 9*(4), 378-383.

Amakali, S. (2008). *Development of models for short-term load forecasting using artificial neural networks.*

Antonelli, M., Ducange, P., & Marcelloni, F. (2012). *Multi-objective evolutionary rule and condition selection for designing fuzzy rule-based classifiers.* Paper presented at the Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on.

Bailey, A. (2001). *Class-dependent features and multicategory classification.* Citeseer.

Barghi, F., & Safavi, A. (2012). An intelligent control policy for fuel injection control of CNG engines. *Iranian Journal of Science and Technology. Transactions of Electrical Engineering, 36*(E1), 83.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *Neural Networks, IEEE Transactions on, 5*(4), 537-550.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*: Kluwer Academic Publishers.

Bose, N., & Kumar, N. S. (2007). *Prediction of engine emissions through Fuzzy logic Modeling.* Paper presented at the Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on.

Boyd, S., Kim, S.-J., Vandenberghe, L., & Hassibi, A. (2007). A tutorial on geometric programming. *Optimization and engineering, 8*(1), 67-127.

Brown, G., Pocock, A., Zhao, M.-J., & Luján, M. (2012). Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The Journal of Machine Learning Research, 13*(1), 27-66.

Cao, H., Du, D., Peng, Y., & Yin, Y. (2006). Air-Fuel-Ratio optimal control of a gas heating furnace based on fuzzy neural networks *Advances in Neural Networks-ISNN 2006* (pp. 876-884): Springer.

Carvalho, D. R., & Freitas, A. A. (2002). A genetic-algorithm for discovering small-disjunct rules in data mining. *Applied soft computing, 2*(2), 75-88.

Casillas, J., Cordón, O., Del Jesus, M. J., & Herrera, F. (2001). Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems. *Information Sciences, 136*(1), 135-157.

Casillas, J., Cordón, O., Del Jesus, M. J., & Herrera, F. (2005). Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction. *Fuzzy Systems, IEEE Transactions on, 13*(1), 13-29.

Chakraborty, D., & Pal, N. R. (2004). A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification. *Neural Networks, IEEE Transactions on, 15*(1), 110-123.

Chen, Y.-C., Pal, N. R., & Chung, I. (2012). An integrated mechanism for feature selection and fuzzy rule extraction for classification. *Fuzzy Systems, IEEE Transactions on, 20*(4), 683-698.

Chiu, S. L. (1994). Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems, 2*(3), 267-278.

Coello, C. A. C. (2005). An introduction to evolutionary algorithms and their applications *Advanced Distributed Systems* (pp. 425-442): Springer.

Cordón, O., Herrera, E., Gomide, E., Hoffman, E., & Magdalena, L. (2004). *Ten years of genetic fuzzy systems: current framework and new trends.* Paper presented at the IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th.

Cordón, O., Herrera, F., Hoffmann, F., & Magdalena, L. (2001). *Genetic fuzzy systems*: World Scientific Publishing Company Singapore.

Cover, T. M., & Thomas, J. A. (2012). *Elements of information theory*: John Wiley & Sons.

Dailey, D. J., Harn, P., & Lin, P.-J. (1996). ITS data fusion: Washington State Department of Transportation.

Dehuri, S., Patnaik, S., Ghosh, A., & Mall, R. (2008). Application of elitist multi-objective genetic algorithm for classification rule generation. *Applied soft computing, 8*(1), 477-487.

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters.

Durrant-Whyte, H., & Henderson, T. C. (2008). Multisensor data fusion *Springer Handbook of Robotics* (pp. 585-610): Springer.

Ebrahimi, B., Tafreshi, R., Masudi, H., Franchek, M., Mohammadpour, J., & Grigoriadis, K. (2012). A parameter-varying filtered PID strategy for air–fuel ratio control of spark ignition engines. *Control Engineering Practice, 20*(8), 805-815.

Emmanouilidis, C., Hunter, A., & MacIntyre, J. (2000). *A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator.* Paper presented at the Evolutionary Computation, 2000. Proceedings of the 2000 Congress on.

Fazzolari, M., Alcala, R., Nojima, Y., Ishibuchi, H., & Herrera, F. (2013). A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions. *IEEE Transactions on fuzzy systems, 21*(1), 45-65.

Fernández, A., López, V., del Jesus, M. J., & Herrera, F. (2015). Revisiting Evolutionary Fuzzy Systems: Taxonomy, applications, new trends and challenges. *Knowledge-Based Systems*.

Flach, P. (2012). *Machine learning: the art and science of algorithms that make sense of data*: Cambridge University Press.

Fogel, D. B., & Fogel, L. J. (1996). *An introduction to evolutionary programming.* Paper presented at the Artificial Evolution.

Gacto, M. J., Alcalá, R., & Herrera, F. (2010). Integration of an index to preserve the semantic interpretability in the multiobjective evolutionary rule selection and tuning of linguistic fuzzy systems. *Fuzzy Systems, IEEE Transactions on, 18*(3), 515-531.

Gacto, M. J., Alcalá, R., & Herrera, F. (2011). Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences, 181*(20), 4340-4360.

Garcia, A. M. (2013). *Feed-Forward Air-Fuel Ratio Control during Transient Operation of an Alternative Fueled Engine.* The Ohio State University.

Ghaffari, A., Shamekhi, A. H., Saki, A., & Kamrani, E. (2008). Adaptive fuzzy control for air-fuel ratio of automobile spark ignition engine. *World Academy of Science, Engineering and Technology, 48*, 284-292.

Gliwa, B., & Byrski, A. (2011). Hybrid neuro-fuzzy classifier based on NEFCLASS model. *Computer Science*, 115-135.

Gonzalez, A., & Perez, R. (2001). Selection of relevant features in a fuzzy genetic learning algorithm. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 31*(3), 417-425.

Gonzblez, A., & Pérez, R. (1999). SLAVE: A genetic learning system based on an iterative approach. *Fuzzy Systems, IEEE Transactions on, 7*(2), 176-191.

Gopalakrishnan, K., Ceylan, H., & Attoh-Okine, N. O. (2009). *Intelligent and soft computing in infrastructure systems engineering: recent advances* (Vol. 259): Springer Science & Business Media.

Gopalakrishnan, V., Lustgarten, J. L., Visweswaran, S., & Cooper, G. F. (2010). Bayesian rule learning for biomedical data mining. *Bioinformatics, 26*(5), 668-675.

Gorrostieta, E., & Pedraza, C. (2006). *Neuro fuzzy modeling of control systems.* Paper presented at the Electronics, Communications and Computers, 2006. CONIELECOMP 2006. 16th International Conference on.

Grosan, C., & Abraham, A. (2011). *Intelligent Systems*: Springer.

Guyon, I. (2006). *Feature extraction: foundations and applications* (Vol. 207): Springer Science & Business Media.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research, 3*, 1157-1182.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., & Tibshirani, R. (2009). *The elements of statistical learning* (Vol. 2): Springer.

Herrera, F. (2008). Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence, 1*(1), 27-46.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: U Michigan Press.

Hong, T.-P., Wang, H.-S., Lin, W.-Y., & Lee, W.-Y. (2002). Evolution of appropriate crossover and mutation operators in a genetic process. *Applied Intelligence, 16*(1), 7-17.

Hu, B.-G. (2014). What are the differences between Bayesian classifiers and mutual-information classifiers? *Neural Networks and Learning Systems, IEEE Transactions on, 25*(2), 249-264.

Hu, B.-G. (2015). *Information Theory and Its Relation to Machine Learning.* Paper presented at the Proceedings of the 2015 Chinese Intelligent Automation Conference.

Huysmans, J., Baesens, B., & Vanthienen, J. (2006). Using rule extraction to improve the comprehensibility of predictive models. *Available at SSRN 961358*.

Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J., & Baesens, B. (2011). An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, *51*(1), 141-154.

Ichihashi, H., Shirai, T., Nagasaka, K., & Miyoshi, T. (1996). Neuro-fuzzy ID3: a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning. *Fuzzy Sets and Systems, 81*(1), 157-167.

Ishibuchi, H., Nakashima, T., & Morisawa, T. (1997). *Simple fuzzy rule-based classification systems perform well on commonly used real-world data sets.* Paper presented at the Fuzzy Information Processing Society, 1997. NAFIPS'97., 1997 Annual Meeting of the North American.

Ishibuchi, H., Nakashima, T., & Morisawa, T. (1999). Voting in fuzzy rule-based systems for pattern classification problems. *Fuzzy Sets and Systems, 103*(2), 223-238.

Ishibuchi, H., Nakashima, T., & Murata, T. (2001). Three-objective genetics-based machine learning for linguistic rule extraction. *Information Sciences, 136*(1), 109-133.

Ishibuchi, H., & Nojima, Y. (2007). Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. *International Journal of Approximate Reasoning, 44*(1), 4-31.

Ishibuchi, H., & Yamamoto, T. (2004). Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems, 141*(1), 59-88.

Ishibuchi, H., Yamamoto, T., & Nakashima, T. (2005). Hybridization of fuzzy GBML approaches for pattern classification problems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 35*(2), 359-365.

Ishida, C. Y., Pozo, A., Goldbarg, E., & Goldbarg, M. (2009). Multiobjective optimization and rule learning: Subselection algorithm or meta-heuristic algorithm? *Innovative applications in data mining* (pp. 47-70): Springer.

Jin, Y. (2000). Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement. *Fuzzy Systems, IEEE Transactions on, 8*(2), 212-221.

Jin, Y. (2012). *Advanced fuzzy systems design and applications* (Vol. 112): Physica.

Juang, C.-F., Lin, J.-Y., & Lin, C.-T. (2000). Genetic reinforcement learning through symbiotic evolution for fuzzy controller design. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 30*(2), 290-302.

Karegowda, A. G., Jayaram, M., & Manjunath, A. (2010). Feature subset selection problem using wrapper approach in supervised learning. *International journal of Computer applications, 1*(7), 13-17.

Khaleghi, B., Khamis, A., Karray, F. O., & Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion, 14*(1), 28-44.

Kira, K., & Rendell, L. A. (1992). *A practical approach to feature selection.* Paper presented at the Proceedings of the ninth international workshop on Machine learning.

Knowles, J. D., & Corne, D. W. (2000). *M-PAES: A memetic algorithm for multiobjective optimization.* Paper presented at the Evolutionary Computation, 2000. Proceedings of the 2000 Congress on.

Koch, W. (2010). On Bayesian tracking and data fusion: a tutorial introduction with examples. *Aerospace and Electronic Systems Magazine, IEEE, 25*(7), 29-52.

Kohavi, R. (1995). *A study of cross-validation and bootstrap for accuracy estimation and model selection.* Paper presented at the Ijcai.

Kumar, V., & Minz, S. (2014). Feature Selection. *SmartCR, 4*(3), 211-229.

Lakhmi, J., & Lim, C. (2010). Handbook on Decision Making: Techniques and Applications. *Intelligent Systems Reference Library, 4*, 548.

Lee, H.-M., Chen, C.-M., Chen, J.-M., & Jou, Y.-L. (2001). An efficient fuzzy classifier with feature selection based on fuzzy entropy. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 31*(3), 426-432.

Lee, S. H., Howlett, R., & Walters, S. (2004). Engine fuel injection control using fuzzy logic. *Intelligent Sistems & Signal Processing Laboratories*.

Liu, Z.-q., & Zhou, Y.-c. (2010). *A fuzzy neural network and application to air-fuel ratio control under Gasoline Engine Transient Condition.* Paper presented at the Intelligent System Design and Engineering Application (ISDEA), 2010 International Conference on.

Lughofer, E., & Kindermann, S. (2010). SparseFIS: data-driven learning of fuzzy systems with sparsity constraints. *Fuzzy Systems, IEEE Transactions on, 18*(2), 396-411.

Lv, H., Zhu, B., & Tang, Y. (2007). Fuzzy Classifier with Probabilistic IF-THEN Rules *Foundations of Fuzzy Logic and Soft Computing* (pp. 666-676): Springer.

Ma, B. (2001). *Parametric and nonparametric approaches for multisensor data fusion.* The University of Michigan.

Mahajan, S. R. (2013). Air Pollution from IC Engines & It's Control. *Carbon, 1*(50), 500.

Mandal, S., & Pal, M. K. K. S. K. (2011). Pattern Recognition and Machine Intelligence.

Mansoori, E. G., Zolghadri, M. J., & Katebi, S. D. (2007). A weighting function for improving fuzzy classification systems performance. *Fuzzy Sets and Systems, 158*(5), 583-591.

Mansoori, E. G., Zolghadri, M. J., & Katebi, S. D. (2008). SGERD: A steady-state genetic algorithm for extracting fuzzy classification rules from data. *Fuzzy Systems, IEEE Transactions on, 16*(4), 1061-1071.

Michalewicz, Z. (2013). *Genetic algorithms+ data structures= evolution programs*: Springer Science & Business Media.

Mitchell, H. (2007). Introduction *Multi-Sensor Data Fusion* (pp. 3-13): Springer.

Mitra, S., & Hayashi, Y. (2000). Neuro-fuzzy rule generation: survey in soft computing framework. *Neural Networks, IEEE Transactions on, 11*(3), 748-768.

Mumford, C. L. (2009). *Computational intelligence: collaboration, fusion and emergence* (Vol. 1): Springer Science & Business Media.

Nauck, D., & Kruse, R. (1997). A neuro-fuzzy method to learn fuzzy classification rules from data. *Fuzzy Sets and Systems, 89*(3), 277-288.

Nauck, D., & Nurnberger, A. (2005). *The evolution of neuro-fuzzy systems.* Paper presented at the Fuzzy Information Processing Society, 2005. NAFIPS 2005. Annual Meeting of the North American.

Navot, A. (2006). *On the role of feature selection in machine learning.* Hebrew University.

Negnevitsky, M. (2005). *Artificial intelligence: a guide to intelligent systems*: Pearson Education.

Noda, E., Freitas, A., & Lopes, H. S. (1999). *Discovering interesting prediction rules with a genetic algorithm.* Paper presented at the Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on.

Noda, E., Freitas, A. A., & Yamakami, A. (2003). A Distributed-Population GA for Discovering Interesting Prediction Rules *Advances in Soft Computing* (pp. 287-296): Springer.

Nozaki, K., Ishibuchi, H., & Tanaka, H. (1996). Adaptive fuzzy rule-based classification systems. *Fuzzy Systems, IEEE Transactions on, 4*(3), 238-250.

Oh, I.-S., Lee, J.-S., & Moon, B.-R. (2004). Hybrid genetic algorithms for feature selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26*(11), 1424-1437.

Pal, N. R., & Saha, S. (2008). Simultaneous structure identification and fuzzy rule generation for Takagi–Sugeno models. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 38*(6), 1626-1638.

Peña-Reyes, C. A. (2004). *Coevolutionary fuzzy modeling* (Vol. 3204): Springer Science & Business Media.

Picek, S., & Golub, M. (2010). Comparison of a crossover operator in binary-coded genetic algorithms. *WSEAS transactions on computers, 9,* 1064-1073.

Pomares, H., Rojas, I., Ortega, J., Gonzalez, J., & Prieto, A. (2000). A systematic approach to a self-generating fuzzy rule-table for function approximation. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 30*(3), 431-447.

Qin, Y., Langari, R., & Gu, L. (2015). A new modeling algorithm based on ANFIS and GMDH. *Journal of Intelligent & Fuzzy Systems,* 1-9.

Quinlan, J. (2014). *MDL and categorical theories (continued).* Paper presented at the Proceedings of the Twelfth International Conference on Machine Learning.

Raitamäki, J. (2003). *An approach to linguistic pattern recognition using fuzzy systems*: Jyväskylän yliopisto.

Ravi, V., & Zimmermann, H.-J. (2000). Fuzzy rule based classification with FeatureSelector and modified threshold accepting. *European Journal of Operational Research, 123*(1), 16-28.

Rawat, K., & Burse, K. (2013). A Soft Computing Genetic-Neuro fuzzy Approach for Data Mining and Its Application to Medical

Diagnosis. *International Journal of Engineering and Advanced Technology, 3*(1).

Reid, D. J. (2000). Feasibility and Genetic Algorithms: The Behaviour of Crossover and Mutation: DTIC Document.

Rezaee, M. R., Goedhart, B., Lelieveldt, B. P., & Reiber, J. H. (1999). Fuzzy feature selection. *Pattern Recognition, 32*(12), 2011-2019.

Richter, T., Oliveira, A. F., & da Silva, I. N. (2010). Virtual oxygen sensor implementation using artificial neural networks *Technological Developments in Education and Automation* (pp. 219-224): Springer.

Rigatos, G. G. (2011). *Modelling and Control for Intelligent Industrial Systems*: Springer.

Riza, L. S., Bergmeir, C. N., Herrera, F., & Benítez Sánchez, J. M. (2015). *frbs: fuzzy rule-based systems for classification and regression in R*.

Ross, T. J. (2013). *Fuzzy logic with engineering applications* (Vol. 761): Wiley.

Rudas, I. J., & Kaynak, M. O. (1998). Entropy-based operations on fuzzy sets. *Fuzzy Systems, IEEE Transactions on, 6*(1), 33-40.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling, 5*.

Sadoghi Yazdi, H., & Vahedian Mazloum, A. (2009). Fuzzy Bayesian classification of LR Fuzzy numbers. *International Journal of Engineering and Technology, 1*.

Sánchez, L., Suárez, M. R., Villar, J. R., & Couso, I. (2008). Mutual information-based feature selection and partition design in fuzzy rule-based classifiers from vague data. *International Journal of Approximate Reasoning, 49*(3), 607-622.

Sarkar, B. K., Sana, S. S., & Chaudhuri, K. (2012). A genetic algorithm-based rule extraction system. *Applied soft computing, 12*(1), 238-254.

Saxena, V., & Pratap, A. (2012). Genetic Algorithm Based Bayesian Classification Algorithm for Object Oriented Data. *IRACST-International Journal of Computer Science and Information Technology & Security (IJCSITS), 2*(6), 1177-1182.

Setnes, M., & Roubos, H. (2000). GA-fuzzy modeling and classification: complexity and performance. *Fuzzy Systems, IEEE Transactions on, 8*(5), 509-522.

Silipo, R., & Berthold, M. R. (2000). Input features' impact on fuzzy decision processes. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 30*(6), 821-834.

Sivanandam, S., & Deepa, S. (2007). *Introduction to genetic algorithms*: Springer Science & Business Media.

Tang, Y., & Sun, S. (2004). A mixture model of classical fuzzy classifiers *Advances in Web-Age Information Management* (pp. 616-621): Springer.

Triantaphyllou, E., & Felici, G. (2006). *Data mining and knowledge discovery approaches based on rule induction techniques* (Vol. 6): Springer Science & Business Media.

Ukil, A. (2007). *Intelligent systems and signal processing in power engineering*: Springer Science & Business Media.

Vancoillie, F. (2003). *Design and application of artificial neural networks for digital image classification of tropical savanna vegetation.* Ghent University.

Wang, L.-X., & Mendel, J. M. (1992). Generating fuzzy rules by learning from examples. *Systems, Man and Cybernetics, IEEE Transactions on, 22*(6), 1414-1427.

Wang, L., & Fu, X. (2006). *Data mining with computational intelligence*: Springer Science & Business Media.

Wang, Y., Chang, H., Wang, Z., & Li, X. (2003). Input selection and rule generation in Adaptive Neuro-Fuzzy Inference System for protein structure prediction *Intelligent Data Engineering and Automated Learning* (pp. 514-521): Springer.

Wargo, J., Wargo, L. E., & Alderman, N. (2006). *The harmful effects of vehicle exhaust: A case for policy change*: Environment & Human Health.

Wong, P. K., Wong, H. C., Vong, C. M., & Wong, K. I. (2016). Online wavelet least-squares support vector machine fuzzy predictive control for engine lambda regulation. *International Journal of Engine Research*, 1468087415623909.

Wu, H.-C. (2004). Fuzzy reliability estimation using Bayesian approach. *Computers & Industrial Engineering, 46*(3), 467-493.

Xu, J., & Zhou, X. (2011). *Fuzzy-like multiple objective decision making* (Vol. 263): Springer.

Zarandi, M. H. F., Mohammadhasan, N., & Bastani, S. (2012). A fuzzy rule-based expert system for evaluating intellectual capital. *Advances in Fuzzy Systems, 2012*, 7.

Zhang, C. K., & Hu, H. (2005). An effective feature selection scheme via genetic algorithm using mutual information *Fuzzy Systems and Knowledge Discovery* (pp. 73-80): Springer.

Zhang, X., & Hu, B.-G. (2014). A new strategy of cost-free learning in the class imbalance problem. *Knowledge and Data Engineering, IEEE Transactions on, 26*(12), 2872-2885.

Zhao, H.-c., & Sun, Z.-s. (2012). A Heuristic Algorithm for Rule Extraction Based on Conditional Information Entropy. *International Journal of Advancements in Computing Technology, 4*(21).

Zilouchian, A., & Jamshidi, M. (2000). *Intelligent control systems using soft computing methodologies*: CRC Press, Inc.