NANYANG
TECHNOLOGICAL
UNIVERSITY

SPARSE DYNAMIC REORGANISING FUZZY NEURAL NETWORKS

by

ZHOU WEIGUI JAIR

A thesis submitted to Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

School of Computer Science and Engineering
Nanyang Technological University, Singapore

2017

NANYANG TECHNOLOGICAL UNIVERSITY

SPARSE DYNAMIC REORGANISING FUZZY NEURAL NETWORKS

A thesis submitted to Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

by

ZHOU WEIGUI JAIR

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2017

Acknowledgments

I would to take this opportunity to express my utmost gratitude to both my supervisor, Associate Professor Douglas Maskell, and co-supervisor, Associate Professor Quek Hiok Chai, for their patience and guidance throughout my Ph.D journey. I am apologetic in making them worry and not being able to meet time lines that were initially set out to be met.

I would also like to thank the many people that had been along with me in this journey. I would like to especially thank a couple of these wonderful people. They are Dr Tung Whye Loon, Dr Javan Tan, Dr Cheu Eng Yeow, Dr Liu Yuan, Dr Zhang Yu, Dr Wu Qiong, Dr Liu Siyuan, Ms Su Yng Lih, and many many more. They had shared their experiences in different life journey with me, be it joy or sadness, or be it separation or getting together. These experiences are not able to be learnt from research papers or textbooks.

I would like to thank all staffs and members of Computation Intelligence Laboratory for giving me a conducive environment during my journey. Especially thankful to the lab executive Mr Kesavan for all the technical assistance that he had provided.

Last but not least, I am thankful that I have been given wonderful parents and siblings that are understanding and attentive to me, whom excused me from family gatherings due to my slow progression of my work.

Contents

Acknowledgments	i
Contents	iii
List of Tables	vi
List of Figures	viii
Abstract	xi
1 Introduction	1
1.1 Introduction	1
1.2 Objectives and Aims	4
1.3 Report Organisation	4
2 Literature Review	7
2.1 Fuzzy Logic Systems	7
2.1.1 Fuzzy Logic	7
2.1.2 Mamdani Fuzzy System	11
2.1.3 Takagi-Sugeno-Kang (TSK) Fuzzy System	12
2.1.4 Fuzzy Rule Interpolation/Extrapolation	13
2.2 Fuzzy Neural Networks	22
2.2.1 Motivation	22
2.2.2 Fuzzy Neural Network Categories	23
2.3 Summary	26

3	Fuzzy Rule Interpolation and Extrapolation	29
3.1	Fuzzy Rule Interpolation/Extrapolation	31
3.1.1	Triangular fuzzy sets	31
3.1.2	Trapezoidal fuzzy sets	36
3.1.3	Gaussian fuzzy sets	38
3.1.4	Multiple antecedent fuzzy interpolative reasoning	40
3.1.5	Multiple fuzzy rules with multiple antecedent fuzzy interpola- tive reasoning	43
3.1.6	Fuzzy rule extrapolation	48
3.2	Fuzzy Rule Interpolation Results	49
3.3	Fuzzy Rule Extrapolation Results	51
3.4	Summary	53
4	Offline Learning Fuzzy Neural Networks	55
4.1	Fuzzy input Takagi-Sugeno-Kang (FITSK) model	56
4.1.1	CMAC Layer	57
4.1.2	Localized Fuzzy Rule Computation Layer	59
4.1.3	Output Layer	60
4.1.4	FITSK Learning Scheme	60
4.2	Fuzzy Interpolation and Extrapolation FITSK (FIE-FITSK) Model . .	61
4.2.1	CMAC Layer	61
4.2.2	Localized Fuzzy Rule Computation Layer	62
4.2.3	Output Layer	63
4.2.4	FIE-FITSK Learning Scheme	64
4.3	Experiments and Results	65
4.3.1	Prediction accuracy	66
4.3.2	Handling concept drift/shift	69
4.3.3	Application of Diabetes Modelling	73
4.4	Summary	97
5	Evolving Fuzzy Neural Networks	101

5.1	Generic Self-Evolving Takagi-Sugeno-Kang (GSETSK) Model	102
5.1.1	Forward Reasoning	104
5.1.2	Backward Computations of GSETSK	106
5.1.3	Parameter Learning of GSETSK	109
5.2	Evolving Fuzzy Interpolation and Extrapolation Takagi-Sugeno-Kang (eFIE-TSK) Model	111
5.2.1	Forward Reasoning	111
5.3	Experiments and Results	114
5.3.1	Mackey-Glass time series prediction	114
5.3.2	Non-linear system identification problem	116
5.3.3	Glucose-insulin prediction	122
5.4	Summary	127
6	Conclusions and Future works	129
6.1	Conclusions	129
6.2	Future Works	134

List of Tables

2.1	Existing fuzzy interpolation techniques.	16
2.2	Summary of existing fuzzy interpolation and extrapolation techniques. .	20
2.2	Summary of existing fuzzy interpolation and extrapolation techniques. .	21
2.3	Existing fuzzy neural networks.	23
3.1	Table of notations.	31
3.2	Fuzzy interpolation result for the multiple antecedent rules example. . .	50
3.3	Fuzzy extrapolation result for the multiple antecedent rules example. .	52
4.1	Mackey-Glass: RMSE result for all models	67
4.2	Non-linear system identification problem: RMSE result for all models .	69
4.3	Dynamic non-linear system identification: RMSE results	71
4.4	Notations used in the GlucoSim model.	80
4.5	Blood glucose maximum and minimum limit of different dietary habit trained models and the respective standard deviation of each dietary habit data.	81
4.6	Overall comparison of RMSE between FITSK and FIE-FITSK in dia- betes experiment.	83
4.7	Overall comparison of Correlation (R) between FITSK and FIE-FITSK in diabetes experiment.	83
4.8	Two days meal schedule.	91
5.1	Mackey-Glass: RMSE result for all models	115
5.2	Results of online identification of a non-linear system.	117
5.3	Results of time-varying online identification of a non-linear system. . .	119

5.4	RMSE of the models in predicting insulin residual under different dietary habits and ranking of the models for having the smallest rule size and lowest RMSE as shown in the parenthesis.	122
-----	---	-----

List of Figures

2.1	Type-1 fuzzy logic system, adapted from [82].	8
2.2	(a) Triangular membership function (b) Trapezoidal membership function (c) Piecewise linear membership function (d) Gaussian membership function.	9
2.3	Fuzzy operations: (a) union or <i>max</i> operation, (b) intersection or <i>min</i> operation, and (c) negation or complement operation.	10
2.4	Mamdani inference example.	11
2.5	TSK inference example.	13
2.6	Fuzzy reasoning on tomatoes.	14
2.7	An ordered 2 rules antecedent and consequent.	15
2.8	An extrapolation example on tomato ripeness.	19
3.1	Triangular fuzzy set.	32
3.2	Triangular membership functions for fuzzy interpolation reasoning. . .	32
3.3	Left and right area.	33
3.4	Special case of triangular membership functions fuzzy interpolation reasoning.	35
3.5	Trapezoidal fuzzy set.	36
3.6	Trapezoidal membership functions fuzzy interpolation reasoning. . . .	37
3.7	Trapezoidal fuzzy set collapses in becoming a triangular fuzzy set. . .	38
3.8	Gaussian fuzzy set.	39
3.9	Gaussian membership functions fuzzy interpolation reasoning.	39
3.10	Multi-antecedent fuzzy interpolation reasoning.	41
3.11	Three rules and two antecedent fuzzy interpolation reasoning.	45

3.12 Extrapolation with two rules and two antecedent fuzzy interpolation reasoning.	48
3.13 Fuzzy interpolation for the multiple antecedent rules example.	50
3.14 Fuzzy extrapolation for the multiple antecedent rules example.	52
4.1 FITSK structure.	56
4.2 Fuzzy interpolation and extrapolation FITSK (FIE-FITSK) structure .	62
4.3 Non-linear system prediction result from FIE-FITSK.	70
4.4 Dynamic non-linear system prediction results of FITSK and FIE-FITSK.	71
4.5 Zoom in of Dynamic non-linear system $\delta_A = 1.5$ prediction results of FITSK and FIE-FITSK.	72
4.6 Pharmacokinetic diagram of glucose model.	77
4.7 Pharmacokinetic diagram of healthy person insulin model.	78
4.8 Pharmacokinetic diagram of type-1 diabetic patient insulin model. . . .	79
4.9 Blood glucose fluctuation from the four groups of different dietary habits.	81
4.10 Group A trained network result plots.	84
4.11 Group B trained network result plots.	85
4.12 Group C trained network result plots.	86
4.13 Group D trained network result plots.	87
4.14 Zoom in on all Group A trained Group D* tested, Group B trained Group D* tested, and Group C trained Group D* tested.	88
4.15 Rules involved in inferencing by FITSK and FIE-FITSK techniques. . .	89
4.16 Insulin scheduler system.	94
4.17 Result of simulation with Irregular dietary habit trained models.	94
4.18 Result of simulation with Normal dietary (Group A) trained models. .	95
4.19 Result of FIE-FITSK trained models.	96
5.1 Structure of GSETSK network, adapted from [89].	103
5.2 The Gaussian membership function with mean=0 and standard deviation represented as $\sigma_{back}(t)$	107
5.3 Test result of eFIE-TSK on Mackey-Glass time series experiment. . . .	115

5.4	Test result of eFIE-TSK on online identification of a non-linear system experiment.	117
5.5	Behaviour of system with four different levels of δ_A included between $1001 \leq t \leq 2000$	118
5.6	Prediction results between $501 \leq t \leq 1001$ where $\delta_A = 0.0$	120
5.7	Prediction results between $801 \leq t \leq 1800$ where $\delta_A = 1.5$	120
5.8	Predicted insulin residual by the models for the span of 9 days.	124
5.9	Predicted insulin residual by the models for Day 1.	125
5.10	First three peaks of predicted insulin residual by the models for Day 1.	125
5.11	Predicted insulin residual by the models for Day 2.	126
5.12	First peak of predicted insulin residual by the models for Day 2.	126
5.13	Predicted insulin residual by the models for Day 8.	127
6.1	Suggested rule creation by converting the derived consequent as a rule.	135

Abstract

Fuzzy logic systems can broadly be grouped into two main types; namely: linguistic fuzzy systems (Mamdani) and precise fuzzy systems (Takagi-Sugeno-Kang). Fuzzy neural systems have been extensively researched and utilized in recent years because of their learning capability and interpretability in data modelling.

Fuzzy rules are formed in fuzzy neural systems using either the batch learning approach (offline) or the sequential/evolving learning approach (online). Evolving fuzzy neural systems normally incorporate rule pruning mechanisms which refine the knowledge base to better represent the current trend in the observation, and also reduce the system complexity. In real life, many factors may contribute to uncertainty in data and these include unexpected movement in the data trend or noise corrupted data, etc. An unexpected trend movement in the data can occur in many different problem domain such as in the financial field (stock trading), the medical field (blood glucose regulation), or in nature (flash floods, earthquakes, tsunami, etc), and certain levels of noise interference during the data collection is inevitable.

Traditional fuzzy systems or offline models require a relatively complete rule base to provide good modelling of the data. Additionally, this approach is not designed to change after the training process. Hence, the offline model is constrained by the boundaries of the fixed rule base which limits the flexibility of handling data drift.

On the other hand, the online models often build up the rule base from scratch based on the data presented. As the size of rule grows incrementally, the problems of exploding rule base will be encountered. Rule pruning mechanism which removes rules that are of less significance was proposed. However such pruning mechanism may removes crucial, yet currently inactive, fuzzy rules with overlapping fuzzy sets from the rule base.

Some evolving models had claimed to be able to handle concept drift problem. When new data is presented to the model, it will be able to inference a result and create new rules if rule creating criteria is met. However, in the case of concept shift problem where new data is significantly different from the past, poor inference result may be produced.

Sparse rule base may be formed commonly in two conditions. One, due to the lack of expert knowledge in a specific problem domain, or two, when the fuzzy neural network with rule pruning ability removes overlapping fuzzy rules unintentionally, as mentioned above. The limited flexibility of the offline model to handle data drift, and deficient modelling ability of online model to handle data shift have encouraged the exploration of incorporating fuzzy interpolation and extrapolation technique (FIE) to both models.

Fuzzy interpolation technique is able to infer a result in a sparse or an incomplete fuzzy rule base. On the other hand, fuzzy extrapolation technique is able to infer a result when new data is found beyond the existing coverage of the rule base. Such technique enables the fixed rule base of an offline model to infer a result when new data is beyond the boundary of existing rule base, and also enable an online model to infer a result either in a sparse rule base or when new data is too far beyond the coverage of existing rule base.

In this thesis, a new fuzzy interpolation and extrapolation technique extended to the Takagi-Sugeno-Kang fuzzy inference system will be introduced. The incorporation of the proposed FIE technique into two existing fuzzy neural system, representing both offline and online models, will be presented. The results shall show how the incorporation of the proposed FIE technique handles a sparse rule base or concept drift/shift and produces better result as compared to the original extended model.

Chapter 1

Introduction

1.1 Introduction

Computational models are often developed to perform system modelling that are too complex where no other forms of solution are readily available. Tuning the parameters of computational models allow the complex system to be modelled and operation theories can be deduced from the experimentations. However, the data distribution may change and directly affecting the developed models making them less relevant to model or represent the current problem distribution.

Data can be categorised in two broad categories; namely: classification and regression. The classification category includes breast cancer [84, 123], artificial characters [26], thoracic surgery [133], audiology [18], etc. This type of data groups their inputs into different classes of the problem domain and their inputs can be unordered. The regression category includes stock analysis, Parkinson’s disease tele-monitoring data set [114], EEG database [130], automobile fuel consumption prediction data [97], etc. The inputs of this category are continuous or ordered values. The outputs are predicted values learnt from the training data.

In regression problem, the data are defined as single valued data that are continuously collected and evenly time spaced. These regression problems often face two concept changes, known as drift [99] and shift [102], where “concept” refers to the target variable, and “changes” refers to unforeseen behavior of the target variable. Concept drift is represented as a gradual change in the data distribution and concept

shift on the other hand can be seen as an extreme case of concept drift where data distribution may be entirely changed at the next time instance. Concept drift can be handled by evolving fuzzy neural networks where new rules are created on the go to adapt to the new data distribution. However, when concept shift occurs, with typical fuzzy inference techniques like Mamdani [80] or Takagi-Sugeno-Kang [109] fuzzy inferencing methods, current input may not be able to match any of the fuzzy rule antecedent using either methods, and therefore poor or null result will be produced.

Different problem domains may prefer different modelling methods, either online or offline [116]. When the data of a problem domain can be acquired completely before the development of the model, the offline method can be adopted. However, when data are acquired only at the time of modelling, the online method is preferred.

Offline models are presented with the complete data [7, 9, 96, 117, 121]; hence the range of data is made known during the development of the model. This allows pre-clustering of the data to form rules that cover the span of the data distribution. Some offline models, for example FITSK [96], requires knowledge of the entire spectrum of distribution in order to predefine the structure of the architecture before training.

Online models do not have *apriori* knowledge of the data distribution range as they are fed sequentially. However, they have the learning ability to expand their rule base by increasing the number of rules according to the current input data distribution [11, 58, 61, 113]. Rule pruning techniques have been incorporated to ensure that the model represents the current data distribution [10, 89, 104, 110].

Data sparseness, defined as sparse data distribution, can lead to a sparse fuzzy rule base. When clustering is performed on a sparsely distributed data, sparse clusters may eventually be formed. These sparsely distributed clusters will be used in fuzzy rules and eventually a non-overlapping sparse fuzzy rule base may be developed. A sparse fuzzy rule base may also be the after-product of the rule pruning techniques that are used to maintain a smaller and more recently used rule base that ensures the model relevancy with the current data distribution [10, 89, 104, 110].

A compact rule base is important as it impacts the memory and computation requirements of a practical implementation. In embedded applications, such as would

be needed for diabetes treatment (which represents a significant part of the test data used in this thesis), there is also the consideration of battery lifespan (i.e. power) and costs of the device due to the increased memory and computation requirements. It is absolutely necessary in these situations to ensure a compact and relevant rule base.

The techniques of rule pruning are proposed to maintain a compact and relevant rule base and each pruning criteria differs in each model ([11, 90, 95, 104, 110, 115] and many more). These techniques enable the model to represent the current data trend more accurately. Pruning is commonly based on the frequency of the rule being fired or a computed potential based on the error incurred by the rule. Once the frequency or potential degrades below a specific threshold, the rule will be pruned. As the potentials focuses on the rule's usability, it does not consider whether the rule is important in the coverage perspective. Therefore, the pruning of rules may lead to other problems like insufficient rules and a sparse rule base. These problems will result in poor rule coverage and hence impact the ability in handling sudden changes in the data movements, or if the data trend returns to the region covered by the pruned rule.

In this thesis, the fuzzy neural network (FNN) model has been adopted to represent the computational model. As mentioned above, the FNN model exists in either offline or online forms. In the context of an offline model, the complete data set is presented at the beginning of the modelling. In reality, the data distribution may change over time. However, offline models are not designed to evolve to cope with such an existence; hence the developed model may no longer be effective in representing the current problem. Although online models are enhanced with the ability to learn sequentially and evolve the rule base accordingly, the increasing size of the rule base demands a higher memory space and computational time. Therefore pruning techniques are proposed to reduce and maintain a rule base that is able to represent the current problem. However, the pruning technique may in turn create a sparse rule base or remove rules that are useful when there is a reversal in the data trend.

Based on the discussion above, the existing problems can be broadly grouped into two issues, namely: sparsity in the rule base and modelling the changes in the data

distribution. It is these problems that this thesis attempts to address.

1.2 Objectives and Aims

This thesis attempts to address two main objectives. The first objective is to ensure reasonable inference arising from a sparse rule base situation be it due to the sparse nature of data or the operation of the fuzzy rule learning system. The second objective is to develop the ability to handle changes in the data distribution or also known as concept drift/shift.

Thus, the main aims of this thesis are to address the issues of:

A - Sparse fuzzy rule systems:

1. Propose a new fuzzy interpolation technique that is able to handle multiple antecedents and multiple rules.
2. Extend the technique with interpolation and extrapolation capability.
3. Implement fuzzy neural networks that incorporate the proposed fuzzy interpolation and extrapolation techniques using different learning mechanisms.

B - Concept drift/shift:

1. Handling concept drift/shift with proposed fuzzy extrapolation technique.

Thus, the focus of this thesis is in developing a fuzzy interpolation and extrapolation technique that will be able to handle situations of sparse rule base and producing plausible result when concept shift occurs. The examination of clustering methods to determine the best clustering and learning method is not within the scope of this thesis.

1.3 Report Organisation

The thesis is organised as follows:

- Chapter 1 provides the introduction and scope of the thesis.
- Chapter 2 discusses the related works of various researchers supporting this thesis.

- Chapter 3 provides a detailed explanation on the fuzzy interpolation and extrapolation technique proposed.
- Chapter 4 focuses on the implementation of the proposed offline learning fuzzy neural networks named fuzzy interpolation and extrapolation fuzzified input Takagi-Sugeno-Kang (FIE-FITSK) model.
- Chapter 5 focuses on the implementation of the proposed online learning fuzzy neural network named the evolving fuzzy interpolation and extrapolation Takagi-Sugeno-Kang (eFIE-TSK) model.
- Chapter 6 concludes the report with future directions and lessons learnt.

Chapter 2

Literature Review

In this chapter, the following topics will be discussed:

- Fuzzy logic systems (FLS)
- Fuzzy neural networks (FNN).

2.1 Fuzzy Logic Systems

Words mean different things to different people and therefore there is vagueness associated with words. Fuzzy logic is used in this context to address the vague meaning when it computes with words [83] where the degree of truth is computed rather than a definite “true” or “false”. Fuzzy logic systems (or fuzzy systems) make use of fuzzy logic approach which revolves around the use of linguistic IF-THEN rules to establish how inputs are matched to rules, which in turn are used to derive the output. In this section, the basics of fuzzy logic and fuzzy systems will be introduced. Fuzzy systems can be broadly grouped into two main types; namely: linguistic and precise fuzzy logic systems, also known as Mamdani [80] and Takagi-Sugeno-Kang (TSK) [109] fuzzy systems, respectively.

2.1.1 Fuzzy Logic

This section introduces the key concepts of the type-1 fuzzy logic system. Figure 2.1 shows a fuzzy logic system which is the fundamental component of fuzzy systems. It consists of the crisp inputs being fuzzified and inferred with the rules to output a

fuzzy set which is defuzzified to derive the crisp output.

Figure 2.1: Type-1 fuzzy logic system, adapted from [82].

Crisp sets

A conventional set with 2-valued logic will be referred to as a crisp set in this discussion. A crisp set takes on the value of either 1 or 0 for each individual x in the universe of discourse X , thereby discriminating between members and non-members.

Fuzzy sets

A fuzzy set is the conceptual building block upon which the entire fuzzy paradigm is built. It generalizes the notion of membership on a binary categorization in classical set theory into one that allows for partial membership. The notion of membership in fuzzy sets thus becomes a matter of degree, which is set in the range of $[0, 1]$.

Membership Functions

Membership functions are the building blocks of fuzzy set theory [23]. A membership function can be a curve which defines how each point x in the input space X is mapped, or belongs, to a degree of membership, thus providing a measure of the degree of similarity of an element in X to the fuzzy set. Figure 2.2 illustrates some commonly used shapes for membership functions; they are namely: (a) triangular, (b) trapezoidal, (c) piece-wise linear, and (d) Gaussian. The use of membership functions achieves greater resolution in the representation, at the expense of higher computational complexity. Triangular and trapezoidal membership functions are named after their appearance. Piece-wise linear membership functions are obtained by joining multiple linear functions at certain points as illustrated by Figure 2.2(c), where point

a to point b is a linear function and point b to c is another linear function. Figure 2.2(d) shows a Gaussian membership function with the center at point a and spread to the left and right by the value of σ .

Figure 2.2: (a) Triangular membership function (b) Trapezoidal membership function (c) Piecewise linear membership function (d) Gaussian membership function.

Set operations

Each fuzzy set has a well-defined set of operations. Consider the fuzzy sets A and B over the domain X described by Equation (2.1).

$$\mu_A(x) : X \rightarrow [0, 1]; \mu_B(x) : X \rightarrow [0, 1] \quad (2.1)$$

where μ_A is a fuzzy membership function of the fuzzy set A of the independent variable x , and μ_B is a fuzzy membership function of the fuzzy set B of the independent variable x . Fuzzy sets A and B can have similar or different membership functions as described in Section 2.1.1.

The union of A and B at the domain point x is described by Equation (2.2) as shown in Figure 2.3(a).

$$\mu_{A \cup B}(x) = \mu_A(x) \bullet \mu_B(x) \quad (2.2)$$

where $\bullet$ represents a t-conorm operator. The *max* operator is commonly used for the t-conorm operator.

Figure 2.3: Fuzzy operations: (a) union or *max* operation, (b) intersection or *min* operation, and (c) negation or complement operation.

The intersection of A and B at a specific domain point x is described by Equation (2.3), as shown in Figure 2.3(b).

$$\mu_{A \cup B}(x) = \mu_A(x) * \mu_B(x) \quad (2.3)$$

where $*$ represents a t-norm operator. The t-norm is commonly represented by a *min* or product operator.

The negation or complement of a fuzzy set is shown in Figure 2.3(c) and is described by Equation (2.4).

$$\neg \mu_A(x) = 1 - \mu_A(x) \quad (2.4)$$

Fuzzy rules

A fuzzy rule is the basic unit for capturing the human reasoning process that is able to make decisions in an environment of imprecision. It consists of an antecedent (IF-clause) and a consequence (THEN-clause), which can be represented by Equation (2.5).

$$\text{IF } x \text{ is } A, \text{ THEN } y \text{ is } B \quad (2.5)$$

where A and B are fuzzy sets representing the antecedent and consequent sets, respectively.

A fuzzy IF-THEN rule associates a condition described using linguistic variables and fuzzy sets to a conclusion. Linguistic variables mean that the values of the variables are in the form of words or sentences in a natural or artificial language [129]. For example:

- IF **Pressure** is *HIGH*, THEN **Volume** is *SMALL*
- IF **Road** is *SLIPPERY*, THEN **Driving** is *DANGEROUS*
- IF **Apple** is *RED*, THEN **Ripeness** is *RIPE*

where **Pressure**, **Volume**, **Road**, **Driving**, **Apple**, and **Ripeness** are linguistic variables with respective linguistic values represented as fuzzy sets.

2.1.2 Mamdani Fuzzy System

Mamdani fuzzy-rule based systems consist of a linguistic description in both the antecedent parts and the consequent parts [80]. Each rule is a description of a condition-action statement that may be clearly interpreted by the user. Equation (2.6) is the linguistic rule that describes the mapping from $U_1 \times \dots \times U_i \times \dots \times U_I$ to W .

$$R_k : \text{IF } x_1 \text{ is } A_{1,j}^k \text{ and } \dots \text{ and } x_i \text{ is } A_{i,j}^k \text{ and } \dots \text{ and } x_I \text{ is } A_{I,j}^k, \text{ THEN } y^k \text{ is } B^k \quad (2.6)$$

where x_i are the input variables with $i = (1, 2, \dots, I)$, y_k is the output variable, $A_{i,j}^k$ is the j^{th} fuzzy set of the i^{th} input that is connected to rule R_k , and B^k is the fuzzy set of the output y_k that belongs to rule R_k .

The mechanism of the Mamdani inference is illustrated using a simplified model described using two-antecedent rules, as shown in Figure 2.4.

Figure 2.4: Mamdani inference example.

In Figure 2.4, let x_1 and x_2 be the observation inputs in this example. The minimum operation is performed at the antecedent X_1 and X_2 to derive the firing

levels from each rule of the fuzzy system. These firing levels determine its fulfillment to the fuzzy output in obtaining a trapezoidal membership function. The maximum operator is hence applied to the R_1 and R_2 outputs to derive the combined output as illustrated in Figure 2.4. An output fuzzy set is formed and different defuzzification methods can be used to derive the final crisp output value.

2.1.3 Takagi-Sugeno-Kang (TSK) Fuzzy System

Instead of working with linguistic rules as in Mamdani models, Takagi, Sugeno, and Kang proposed a new model based on rules where the antecedent is composed of linguistic variables and the consequent is represented by a function of the input variables [105, 109]. The main difference between Mamdani and TSK is the output membership functions. The TSK outputs are either constant or linear (zero or first-order) instead of a fuzzy set representation. A typical first-order rule with two antecedents of a TSK fuzzy model has the form described by Equation (2.7), where the consequent comprises of a linear combination of the variables involved in the antecedent.

$$R_k : \text{ IF } X_{1k} \text{ is } x_1 \text{ and } X_{2k} \text{ is } x_2, \text{ THEN output is } y_k = ax_1 + bx_2 + c \quad (2.7)$$

where X_{1k} and X_{2k} are the system input variables, y_k is the output variable, a , b , and c are the numerical constant parameters, and x_1 and x_2 are linguistic labels in the form of fuzzy sets.

For a zero-order TSK model, the output of y_k is a constant when $a = b = 0$. The output level Z_i of each rule is weighted by the firing strength W_i of the rule. For example, for an AND rule with X_{1k} is x_1 and X_{2k} is x_2 , the firing strength is described as Equation (2.8).

$$w_i = \bigcap (F_1(x_1), F_2(x_2)) \quad (2.8)$$

where F_1 , and F_2 are the membership functions for x_1 and x_2 , respectively. The final output of the system is the weighted average of all rule outputs, computed by

Equation (2.9).

$$Final\ Output = Y = \frac{\sum_{i=1}^N w_i y_k}{\sum_{i=1}^N w_i} \quad (2.9)$$

The inference performed by the TSK model is an interpolation of the entire relevant linear model. The degree of relevance of a linear model is determined by the degree by which the input data belongs to the fuzzy subspace associated with the linear model. These degrees of belongingness become the weight in the interpolation process. Figure 2.5 demonstrates a 2 input and 1 output TSK model. The minimum or product operation is performed on the inputs x_1 and x_2 to derive the weight w_i where i represents the rule identity. The output Y is computed by taking the weighted average of the 2 input values using Equation (2.9).

Figure 2.5: TSK inference example.

2.1.4 Fuzzy Rule Interpolation/Extrapolation

Conventional FLS are used with compact rule bases; however, in situations where there is a sparse rule base, no inference may be made as the fuzzy input or output mapping may not have adequate coverage in the sparse rule base. Rule bases can be created in two ways, one is by expert knowledge where experts will define the rules for the system, the other is where the rule base is extracted from existing data. A sparse rule base may occur when there is missing expertise for certain system configurations and state space regions, or when the collected data may not be significantly adequate to cover the whole problem domain. When the number of inputs is large, the size of

the fuzzy rule base and complexity will grow exponentially. To resolve the difficulty of comprehension in a complex and large rule base, a possible solution is to prune redundant rules which will lead to a reduction in complexity but may lead to an incomplete rule base and possibly a sparse rule base when a large number of redundant or non current rules exist.

Fuzzy rule interpolation

Figure 2.6 shows a classical example using tomato ripeness as an illustration. Ripeness of a tomato involves a gradual coloration change process from green to red representing unripe to fully ripe, respectively. Figure 2.6(a) shows a rather compact rule base that provides a good coverage of the problem space while Figure 2.6(b) shows a sparser rule base. If we are only given the extreme knowledge of an unripe tomato being green and a fully ripe being red, using the conventional FLS, under the compact rule base scenario, when a yellow tomato is provided as the observation, a half-ripe result cannot be deduced. However, if the rule base is sparse as shown by Figure 2.6(b), a logical answer of half-ripe should be inferred when an observation of a yellow tomato is given, however no result can be deduced by a conventional FLS because of the inadequate coverage of the problem space because of the sparse rule base.

Figure 2.6: Fuzzy reasoning on tomatoes.

In 1993, Kóczy and Hirota introduced approximate reasoning by linear rule interpolation in sparse fuzzy rule bases [62, 63]. With linear interpolation, a possible solution to the sparse rule base is shown in Figure 2.7. Having an observation A^* falling in between the two disjoint rules, $R_1 : A_1 \Rightarrow B_1$ and $R_2 : A_2 \Rightarrow B_2$,

with no overlap, as shown in Figure 2.7, the consequent fuzzy set B^* should be found in between the two consequences of the flanking rules.

Figure 2.7: An ordered 2 rules antecedent and consequent.

The antecedent and consequent have to be comparable in the sense of the partial ordering in each respective space, and all the fuzzy sets are expected to be normal and convex.

$$\min\{A_1\} \prec \min\{A^*\} \prec \min\{A_2\} \text{ and } \max\{A_1\} \prec \max\{A^*\} \prec \max\{A_2\}$$

and

$$\min\{B_1\} \prec \min\{B^*\} \prec \min\{B_2\} \text{ and } \max\{B_1\} \prec \max\{B^*\} \prec \max\{B_2\}$$

Fuzzy interpolation techniques can be broadly classified into two categories, direct inference methods or analogical methods. The direct inference method of interpolation has been developed based on the decomposition and resolution principles that mainly employ the alpha-cut methodology to represent individual fuzzy sets, while fuzzy interpolation based on the analogical reasoning mechanism creates intermediate rules such that its output is derived based on the closeness of the antecedent and the observation.

Table 2.1 shows some of the established methods classified in terms of individual categories and also their capabilities in handling two or multiple rules. This table does not include other implementations that directly use the techniques already listed for other extension.

Table 2.1: Existing fuzzy interpolation techniques.

Paper	Direct inference	Analogical	Two rules	Multi-rules
Kóczy <i>et al.</i> 1993[62, 63]	x		x	
Wu <i>et al.</i> 1996[124]		x	x	
Brayani <i>et al.</i> 1996 [15]		x	x	
Hsiao <i>et al.</i> 1998[48]	x		x	
Baranyi <i>et al.</i> 1999[16]	x		x	
Tikk <i>et al.</i> 2000[40]	x		x	
Ughettor <i>et al.</i> 2000[119]	x		x	
Baranyi <i>et al.</i> 2004[17]	x		x	
Kovac <i>et al.</i> 2006[66]	x		x	
Johanyak <i>et al.</i> 2006[57]		x	x	
Huang <i>et al.</i> 2006[49]		x	x	
Chen <i>et al.</i> 2008[33]	x		x	
Chang <i>et al.</i> 2008[29]		x		x
Huang <i>et al.</i> 2008[50]		x		x
Chen <i>et al.</i> 2011[32]	x			x
Chen <i>et al.</i> 2012[34]	x			x
Yang <i>et al.</i> 2013[126]		x		x
Chen <i>et al.</i> 2016[31]		x		x

From Table 2.1, it can be seen that most methods are based on the direct inference method which may be computationally expensive if the levels of α -cuts are increased. Another point to note is that most implementations are based on two rule inference methods. However, such an approach in interpolating only two rules is inflexible to deal with fuzzy interpolative reasoning in sparse fuzzy-rule-based systems. Multiple fuzzy rule interpolation is desirable for deriving more relationships between the fuzzy rules in sparse fuzzy rule bases as this provides greater room for flexibility.

Shi and Mizumoto [100] identified that the method proposed by Kóczy and Hirota [62, 63] may result in a non-convex consequent fuzzy set, and hence they proposed two types of reasoning conditions and methods to deduce and rectify for a normal and convex fuzzy set in the interpolated conclusions. A further extension of [100] was proposed by Wu, Shi, and Mizumoto in [124], whereby a more general normal

and convex condition of [62, 63] was proposed and implemented. This approach guarantees the normality and convexity of the consequence fuzzy result, and is based on the supposition that there is a common similarity in the antecedent and consequent parts. Other researchers also explored alternative methods of interpolation based on different comparisons and measurements. Baranyi, Gedeon, and Kóczy [15] proposed a method that infers the conclusion based on the interpolation of relations. This method uses the analogical technique that creates an interpolated fuzzy rule that is near the observation and infers a conclusion from the fired rules according to the similarity between the rule antecedent and observation. This method works on crisp, triangular, and trapezoidal fuzzy sets. A rational fuzzy interpolation in the vague environment of the fuzzy rule base was proposed by Kovács and kóczy [65] and was benchmarked in an automatically guided vehicle control experiment, showing similar results to conventional fuzzy methods, but with a greater reduction in the number of rules used. Hsiao, Chen, and Lee [48], and Chen, Hsin, Yang, and Chang [34] exploited the slopes of the fuzzy sets of the antecedent to derive the similarity, which is projected on the consequences to derive the conclusion to a given observation. Chen and Chang [32] derived a weighted approximation using the n -nearest rules to infer a result.

Interpolation between Cartesian spaces is introduced by Yam and Kóczy [128], whereby a fuzzy rule base is converted as the mapping between the antecedent and consequent spaces. This interpolation method is based on preserving similarity triangles in the antecedent and consequent spaces. A modification to the initial interpolation implementation of [62, 63] that preserves the advantageous complexity behavior and alleviates the possibility of creating an abnormal conclusion is proposed by Baranyi, Tikk, Yam, *et al* [16]. The modification includes a coordinate transformation which ensures that the computed conclusion is always normal and convex. Jenei [53] proposed a method of compact fuzzy quantities for linear interpolation, and also extended this to non-linear interpolation. It was further extended by Jenei, Klement, and Konzel [54] for handling the n -dimensional input space case through aggregation operators.

Yang and Shen [126] presented fuzzy interpolation in a closed-form and ensured that the interpolated results are valid fuzzy sets. Jin, Diao and Shen [55, 56] proposed backward fuzzy interpolation to deal with missing antecedents in an existing rule base. These works made use of the inference method proposed in [50] to deduced the missing antecedents. Fuzzy interpolation using rough set theory was proposed by Chen, Neil, Li *et al.* [31] which includes approximation of a vague concept that improved the uncertainty handling capability. In [125], the method by [50] was used together with assumption-based-truth-maintenance system (ATMS) and general diagnostic engine (GDE) to develop an adaptive fuzzy interpolation system. [125] was further developed by Yang, Chao and Shen in [127] to allow diagnosis and modification of observations and rules.

The extension to interval type-2 fuzzy interpolation was proposed by Chen, Lee and Shen [35], and Chen, Quek and Shen [30] to handle a higher degree of uncertainty in the knowledge which was extended from [32] and [50], respectively.

Fuzzy rule extrapolation

In cases when the observation is found beyond the existing rules, fuzzy interpolation techniques are unable to infer a conclusion. An example is shown in Figure 2.8. From the figure, assuming a complete rule base with three rules; namely:

- Rule 1: IF **Color** is *GREEN*, THEN **Ripeness** is *UNRIPE*,
- Rule 2: IF **Color** is *YELLOW*, THEN **Ripeness** is *HALFRIPE*,
- Rule 3: IF **Color** is *RED*, THEN **Ripeness** is *RIPE*.

When the observation of a brown tomato is given to infer a conclusion, the interpolated conclusion will not be logical. The interpolated result using n -nearest rules will infer a result appearing in between the existing rule consequent. Therefore an extension of interpolation is designed to handle such situation, known as fuzzy extrapolation. Taking into consideration that the observation still falls within the same dimension, although it does not fall in between the existing rules, the deduced fuzzy consequent should also fall within the same dimension of all other fuzzy rule consequents. Using this consideration, the interpolation technique can be extended into an

extrapolation technique, as follows.

Figure 2.8: An extrapolation example on tomato ripeness.

Kóczy and Hirota [63] proposed a signed distance measure instead of a typical distance measure which does not ensure the order of fuzzy sets. If observations are found to be lying on a single side of existing rules, the deduced fuzzy consequent should be found on the same side of the existing fuzzy rule consequents. Jenei [53] introduced the fuzzy extrapolation method implemented in two ways: The first is to shift the “extreme” rule, whereby the last rule is shifted to the position of the observation and the consequent of the rule is shifted accordingly. The transformation that follows is similar to the implementation of interpolation previously discussed. The second creates another rule in such a way that the required monotonicity and non-positivity is guaranteed to prevent a non-convex fuzzy consequent, and the consequent is derived accordingly.

Huang *et al.* [50] proposed another extrapolation implementation whereby scaling and movement transformation is performed. This extrapolation technique is an extension to the interpolation technique discussed. It creates a new fuzzy set with a similar representative value as the observation. It is followed by a scaling transformation to transform the current support into a new support that is similar to the observation. In a trapezoidal fuzzy set, the top support will consist of a left and right support. The left and right support represents the range where the membership value at every point of the trapezoidal fuzzy set is 1. This is elaborated graphically in Figure 3.7, in the Chapter 3. The same representative value and ratio of the left-support to right-support of the transformed fuzzy set is maintained. An additional shift transformation is also required for the extrapolation, which is required because

the representative value found is not similar to the observation whenever the observation is not flanked by the rules. This additional shift transformation is also performed in the derivation of the observation's consequent to derive the final consequent fuzzy set result.

Table 2.2 summarises some of the existing fuzzy interpolation and extrapolation techniques.

Table 2.2: Summary of existing fuzzy interpolation and extrapolation techniques.

Year	Authors	Description
1993	Kóczy and Hirota (KH) [62, 63]	The first implementation of fuzzy interpolation method was proposed. It was based on the alpha-cut linear interpolation technique. However it was discovered that the derived consequent fuzzy set may be non-convex under certain situations.
1995	Shi and Mizumoto [100]	Additional conditions were applied on the KH implementation to ensure the derived consequent fuzzy set's convexity
1996	Wu, Mizumoto, and Shi [124]	With the motivation from [100], the authors proposed the first analogical interpolation technique. The analogical technique will derive an intermediate fuzzy rule with reference to the observation given. This ensures that the derived consequent fuzzy set is convex.
1997	Kovács and Kóczy [65]	A method of deriving the distance measure based on a vague environment was proposed. This method derived a similarity measure based on the distance measure between the rule base and the observation. The conventional distance measure range from [0-1] was unable to represent the distance between non-overlapping rule and observation. Therefore a vague distance measure was proposed with a range of [0- ∞]. However, this method had two drawbacks. It lacks the fuzziness on the observation and conclusion. The authors extended this method to handle the drawbacks [66].
1998	Hsiao, Chen, and Lee [48]	Hsiao <i>et al.</i> proposed another interpolative reasoning method, based on the work of Shi and Mizumoto [100], that ensured convexity was preserved. This made use of the derived highest point of the consequent fuzzy set, spreading it sideways according to the similarity of the slopes from the antecedent of the rules and observation.
2000	Tikk and Baranyi [40]	Introduced a transformation of the original coordinate system to a new coordinate system to guarantee convexity of the intermediate consequent fuzzy set. The consequent fuzzy set derived in this new coordinate system would be transformed back into the original coordinate system in deriving the convex consequent fuzzy set.
2000	Ughetto, Dubois, and Prade [119]	A new interpolative method was suggested by obtaining the consequent fuzzy set by using interval arithmetic.
2001	Jenei [53]	A list of conditions to be satisfied to validate the fuzzy interpolation and extrapolation techniques were also proposed. The method chooses two rules and determines the linear average between the chosen rules and the observation. A linear re-scale is performed to ensure that the input and output range is similar, achieved by shifting the nearest rule to the observation. This method was further extended to handle rules with multiple-dimension inputs [54] in 2002.

Table 2.2: Summary of existing fuzzy interpolation and extrapolation techniques.

Year	Authors	Description
2004	Baranyi, Kóczy, and Gedeon [17]	Proposed a number of rule interpolation methods based on alpha-cut using either fuzzy relation or fuzzy semantic interpolation. Introduced an interpolation generatrix which enabled both linear and nonlinear types of interpolation with the use of weightage from individual rules. These methods were able to handle a multi-variable rule base.
2006	Johanyak and Kovács [57]	A rule interpolation method based on the polar cut concept was proposed. An intermediate rule was created based on the weighted average of the polar distance between the rule and observation.
2006	Huang and Shen [49]	Derived a distance measure based on the representative value of each fuzzy rule and the given observation. The usage of the representative value substitutes for the alpha-cut point used in [62, 63]. An intermediate rule was created based on the distance measure to the representative value. A rate of scaling and a move ratio had to be computed based on the antecedent of the intermediate rule and the observation. This scale rate and move ratio were used on the intermediate rule consequent to derive the resulting fuzzy set. This method enables multi-antecedent variable interpolation. An extension of this work [50] enabled the ability to handle multiple rule environments and fuzzy extrapolation.
2008	Chen and Ko [33]	Used the cutting of geometric membership functions and incremental and ratio transformations for dealing with fuzzy interpolative reasoning. This method only enables a two rule environment.
2008	Chang, Chen, and Liao [29]	This method derives the fuzzy consequent according to the area ratio similarity, calculated by comparing the antecedent of the rule and observation, and casting the ratio transformation on the consequent of the rules. This method allows multiple antecedent and multiple rule scenarios. However, the experimental results reported that under the multiple rule scenario, the result does not consider the shifting ratio. This is crucial otherwise the derived intermediate rule's antecedent does not coincide with the position of the observation.
2012	Chen, Hsin, Yang, and Chang [34]	This method uses the properties of the slopes of the fuzzy sets to derive fuzzy interpolative reasoning. This method is applicable on Gaussian and polygonal fuzzy sets, and it allows multiple fuzzy rules with multi-antecedent. However this method is not capable of handling fuzzy extrapolation.
2013	Yang, Chao, and Shen [126]	This method performs fuzzy interpolation in a closed-form manner which ensures that the interpolated results are valid fuzzy sets.
2016	Chen, Neil, Li <i>et al.</i> [31]	Fuzzy interpolation using rough set theory was proposed in [31] which includes approximation of a vague concept that improved the uncertainty handling capability.

The important feature of fuzzy interpolation and extrapolation is the ability to inference when the rule base is too sparse or incomplete. Classical fuzzy inference method requires a rule base that has comprehensive coverage in each antecedent or input dimension. The lack of such coverage will in turn produce poor or even no

result.

From the above, many different types of fuzzy interpolation methods were discussed. However, many of which does not have the ability to handle multiple antecedents or multiple rules and there is also a lack of extrapolation ability. The importance of having the ability of handling multiple antecedents or multiple rules is to increase the flexibility of the inference method. In view of the extrapolation ability, it has the potential of handling concept shift of data.

2.2 Fuzzy Neural Networks

The combination of fuzzy logic, neural networks, and expert systems have proven their effectiveness in a wide variety of practical problems. In this section, the fuzzy neural network and its evolution will be examined.

2.2.1 Motivation

Fuzzy logic systems (FLS), as mentioned earlier, have the capability for reasoning with imprecise information to deduce a good explanation of the result. However, these systems are unable to generate their own rule base for reasoning. On the other hand, artificial neural network (ANN) are good at pattern recognition, however, they lack the ability to explain clearly the process of generating the result. Thus, the abilities of each system (FLS and ANN) are complementary. The use of such hybrid systems, known as fuzzy neural network (FNN) is growing with many successful applications in control, financial trading, credit evaluation, medical diagnosis, etc.

The membership functions used in fuzzy systems are tuned using neural network techniques so as to refine the decision making systems for modelling a knowledge representation. Although fuzzy logic is used to encode expert knowledge directly using rules with linguistic labels, it usually takes a lot of time to design and tune the membership function which quantitatively define these linguistic labels. This process can be automated by the learning mechanism in the neural network, resulting in a greatly reduced development time and cost, with a significantly improved performance. However, the fuzzy model is based on the data distribution, and when this

basic assumption is violated the model’s capability for explanation and prediction is compromised.

2.2.2 Fuzzy Neural Network Categories

Fuzzy neural networks can also be broadly classified under two broad categories; namely: offline learning and online learning. A list of some (but not all) recent work is shown in Table 2.3, separated according to the learning mode.

Table 2.3: Existing fuzzy neural networks.

Online Learning Networks			Offline Learning Networks		
Paper	Model Name	Year	Paper	Model Name	Year
[61]	DENFIS	2002	[52]	ANFIS	1993
[11]	eTS	2004	[121]	SANFIN	2002
[69]	SOFNN	2004	[117]	GenSoFNN	2002
[10]	simpl_eTS	2005	[9]	POPFNN-CRI	2003
[98]	SaFIS	2006	[7]	RSPOP	2005
[76]	FLEXFIS	2008	[96]	FITSK	2006
[38]	SOFMLS	2009	[74]	Hebb-RR	2007
[110]	SeroFAM	2010	[101]	MSBFNN	2009
[116]	eFSM	2010	[43]	FASCOM	2009
[77]	ds-FLEXFIS	2011	[75]	WFNN	2011
[51]	SeroFAM-TSK	2012			
[104]	McFIS	2012			
[90]	SPLAFIS	2014			
[95]	GENEFIS	2014			
[89]	GSETSK	2015			

Offline and online learning modes are two different supervised learning frameworks, where in both frameworks, a learning problem is primarily defined by an instance in space X and a label set Y , and the goal is to assign labels from Y to instances in X .

In offline learning, the data distribution over the product space $X \times Y$ is derived from the training set. An offline learning algorithm uses the training set to generate an output model, which is a function that maps instances in X to labels in Y . We expect an offline learning algorithm to generalize, in the sense that its output model should accurately predict the output of unseen examples (out-samples), which are sampled from the same distribution.

The adaptive neuro-fuzzy inference system (ANFIS) [52] is one of the popular

fuzzy neural network and is still widely used in different applications [5, 13, 85]. It is a neural fuzzy system with five-layer feed forward network structure using Kalman filter algorithm for consequent parameter tuning, and back-propagation algorithm using gradient descent method for antecedent parameter tuning. ANFIS uses the Takagi-Sugeno (TS) fuzzy system whereby the output is a polynomial function. When the polynomial function is a constant, a zero order TS model is formed which is also a special case of a fuzzy singleton Mamdani model.

The fuzzy input Takagi-Sugeno-Kang model (FITSK) [96] uses an underlying fuzzy cerebellar model articulation controller (FCMAC) learning structure that generalizes with localized learning with lesser computational complexity, improved local model interpretability, and maintaining the global learning accuracy. However, the FCMAC structure of this model is determined during the initialisation of the network. The boundary of the training data will determine the upper and lower bound of the FCMAC structure. Every interlace of the cells of will represent a TSK rule. Although its interlaced dimension structure is able to cover the entire spectrum of the data distribution, not all rules are trained due to the nature of the data distribution, potentially creating a sparse rule base. Another limitation that lies in the fixed boundary is when new data that are beyond the fixed boundary are presented to the model, traditional fuzzy inference system will not be able to select any available rules to deduce an output.

In the online learning framework, no assumptions are made regarding the characteristics of the data. An online learning algorithm sequentially receives the examples and processes them one after another. In each learning iteration, an instance is received and the model attempts to derive an output. The error incurred between the derived output and the actual result will in turn tune the model according to the learning mechanism. New rules were created based on individual learning framework.

In evolving Takagi-Sugeno model (eTS) [11], updating of network structure depends on the potential calculated based on the distance of the current sample from all other training samples. This work was extended in [10] which includes the pruning capability to manage the growing size of the rule in the former implementation.

The self-reorganising fuzzy associative machine (SeroFAM) [110], uses Bienenstock-Cooper-Munro (BCM) theory [22] which includes both associative and dissociative learning that was not used in other online learning models. SeroFAM makes use of Mamdani fuzzy inference system. It was pointed out in [110] that Hebbian learning [46] has a unidirectional growth in weightage which in turn causes the learning model to be unstable. In [110], rule pruning was implemented based on the half-life decay constant which reduces the firing strength of the rule by half after a specific period of time. Rules will be pruned once the firing strength hits below the pruning threshold. SeroFAM-TSK [51] is an extended version of SeroFAM with the same learning and pruning mechanism but implemented in TSK fuzzy inference system.

The meta-cognitive neuro-fuzzy inference system (McFIS) [104] incorporates a learning strategy. McFIS has two components, a cognitive component and a meta-cognitive component. The cognitive component consist of a TSK fuzzy inference system and the meta-cognitive component consist of a self-regulatory learning mechanism that will decide *what-to-learn*, *when-to-learn*, and *how-to-learn*. With the meta-cognitive component, the model is able to achieve high performance in terms of accuracy, size of rule base and also execution time. The mechanism of *what-to-learn* enables a reduction of samples that are of low significance to be learnt. This in turn reduced the amount of time required McFIS as compared to other models that basically learns every data.

The generic self-evolving TSK [89] model (GSETSK) incorporates the Hebbian learning mechanism with pruning capability. A recursively calculated rule potential is calculated based on the current training data, the previous rule potential, the forward firing strength and the backward firing strength of the rule. This potential indicates the importance of the rule is reinforced if its input antecedents and the computed output can closely mimic the information expressed in the training data. A fix decay rate is append to the previous rule potential within the calculation to simulate the decaying of old learning when new data are presented. The fuzzy sets derived at each antecedent is created by a clustering technique known as multidimensional-scaling growing clustering (MSGC) [87]. This clustering technique have been shown to create

fuzzy set partitioning at the antecedent dimension to be highly interpretable.

2.3 Summary

In this chapter, an overview of current existing techniques for fuzzy interpolation and extrapolation (FIE), and fuzzy neural networks (FNN) was presented.

The existing problems in FIE are a lack of a technique that is able to handle multiple antecedent and multiple rules with extrapolation capability. The method proposed in [49] has claimed to be able to handle the mentioned problems. However, to carry out the inferencing in this model, there is a requirement to know the upper and lower bound of the problem domain. This is feasible if pre-known limits can be acquired prior to the inferencing, but this is not feasible in online problem domain, e.g., stock market.

Many FNN were introduced with different learning methods and strategies, and pruning mechanisms that have rule potentials deduced using different functions. Such evolving networks have been shown to be able to adapt to new data distribution on the fly with newly created rules that are representative to handle concept drift problem. However, there are no existing models that have claimed in handling the concept shift problem.

In this thesis, a FIE method will be proposed that is able to handle multiple antecedent and multiple rules with extrapolation capability and does not requires any prior problem domain knowledge. The incorporation of this method to existing offline and online models will be proposed. FITSK and GSETSK have been selected to be extended with the FIE capability.

As FITSK is constrained by the FCMAC boundary issue, the proposed FIE method shall allow higher flexibility to the model beyond the boundary. GSETSK model has the common Hebbian learning with pruning mechanism included. The advantage of this GSETSK is also on the highly interpretable fuzzy antecedent that was not shown by most of the published models. As pruning is involved, chances of sparse rule base may exist. The handling of such situation can be overcome by the fuzzy interpolation technique. Another problem to be addressed is the concept shift

situation. Experiments will be designed to incorporate such behaviour and discussion will be made.

In the next chapter, the proposed FIE method will be discussed.

Chapter 3

Fuzzy Rule Interpolation and Extrapolation

In this chapter, a fuzzy rule interpolation and extrapolation technique based on an analogical reasoning mechanism, which was published in [131], will be discussed. The proposed method uses the area of the fuzzy set between rule antecedents and the observation, to infer the fuzzy interpolative reasoning result, inspired by [29]. It was reported in [29] that the method developed was able to handle multiple-antecedent rules condition. However, the result reported in the experiment showed a wrong comparison in the interpretation of the result obtained from [50].

In [131], two changes were made to the inspired method introduced in [29], namely: the weightage formulation, and an introduction of a shift process. The weightage represents the normalised distance between an observation and the nearest rules. In most method like e.g. [29], [50], [34], the normalised distance is computed as the distance between the observation to the nearest rule over the maximum distance between the n -nearest rules. This method normalised the distance correctly if the observation lies within the maximum distance of the n -nearest rules. However, this method lacks the flexibility of normalising the distance in the situation of extrapolation where the distance between the observation and the nearest rule may be greater than the maximum distance between the n -nearest rules causing the normalisation to fail. In [50], the maximum distance is predefined; hence extrapolation is possible because the predefined maximum distance is greater than the distance that will be computed between

the observation and nearest rules. However, predefined maximum distance is not applicable if the size of the problem domain is not made known beforehand. Therefore, [50] method is not directly extendable to an evolving fuzzy neural system in which no prior knowledge is available. On the other hand, the proposed method in [131] allows normalisation to be dynamically computed without prior knowledge of the size of the problem domain. This allows the proposed method to be directly applied to an evolving fuzzy neural system.

In [50], a shift process was proposed to correct the position of the derived consequent fuzzy set when more than two nearest rules are selected. This process ensures that the derived consequent fuzzy set will be realigned when there is an imbalance number of rules flanking the observation. However, the shift value defined in [50] is determined by a predefined maximum distance in the consequent domain. In [131], the maximum distance is derived from the consequent of the n -nearest rules. This will ensure that the proposed method is directly applicable in an evolving fuzzy neural system. The shift process is prominent in a situation when either an interpolation involving more than two nearest rules or extrapolation is involved. In the interpolation situation involving only two rules, the weightage derived will be balanced among the two rules therefore there is no explicit shifting required. However, when more than two rules are involved during an interpolation, the weightage derived will be imbalanced. The derived consequent will tend to be closer to the side with more rules. In the extrapolation situation, the derived consequent will fall in between the rules instead. A more detailed illustration will be provided in the later section.

Many techniques have been developed as shown in Table 2.2 in Section 2.1. However, most of the techniques developed are neither able to handle multiple antecedents, multiple rule, nor extrapolation situations. The model proposed in [40] requires complex transformation of coordinates to derive a guaranteed convex fuzzy consequent. This method maps the original coordinates of the domain to a new coordinate system and derive the consequent fuzzy set, followed by re-mapping the derived consequent fuzzy set back to the original coordinate. Such a method incurs additional mapping time while most other related works does the transformation at the same coordinate

system. In the proposed method, a direct comparison of the areas of fuzzy sets has been used, and this method does not require any complex transformation as all computations are based on the same domain. The weightage formulation and the shift action in the proposed method allows it to handle multiple rules and is also extendable to extrapolation.

3.1 Fuzzy Rule Interpolation/Extrapolation

Table 3.1 shows the list of notation that is be used in this section. The interpolation

Table 3.1: Table of notations.

Notation	Definition	Notation	Definition
α	Area	λ	Base width
δ	Shift ratio	W/w	Weight
A	Antecedent fuzzy set	B	Consequent fuzzy set
A'	Intermediate antecedent fuzzy set	B'	Intermediate consequent fuzzy set
A^*	Given antecedent fuzzy set	B^*	Derived consequent fuzzy set
$d(.)$	Distance measure	h	Height of fuzzy set
x	Input point	y	Output point
x'	Intermediate input point	y'	Intermediate output point
x^*	Given observation input point	y^*	Final derived consequent point
Subscripts			
i	Rule	j	Antecedent
n	Normal point	L/l	Left
R/r	Right		

process to derive the consequent requires different number of steps for different type of fuzzy sets. Three types of fuzzy sets will be discussed in the following, namely: triangular fuzzy sets, trapezoidal fuzzy sets and Gaussian fuzzy sets.

3.1.1 Triangular fuzzy sets

Triangular fuzzy membership functions are widely used, denoted by (x_l, x_n, x_r) as shown in Figure 3.1, where x_{il} , x_{in} and x_{ir} are the “left extreme point”, the “normal point” and the “right extreme point” of fuzzy set A_i , respectively.

Figure 3.1: Triangular fuzzy set.

Figure 3.2: Triangular membership functions for fuzzy interpolation reasoning.

A rule base consisting of two rules and a single antecedent, as shown in Figure 3.2, is represented as follows,

Rule 1: If X is A_1 Then Y is B_1 .

Rule 2: If X is A_2 Then Y is B_2 .

Observation: X is A^* .

Conclusion: Y is B^* .

Generally speaking, a rule can be represented by Rule i : If X is A_i Then Y is B_i where i represents the rule number.

In Figure 3.2, assuming that there are two flanking disjointed fuzzy rules covering the space of the discussion domain, $A_1 \Rightarrow B_1$ and $A_2 \Rightarrow B_2$ with the observation A^* found in between without a representative rule, then it is expected that the consequent

fuzzy set B^* will be found in between the consequents of the flanking rules. The three steps of fuzzy interpolation are as follows.

Step 1: Determine the normal point y_n^* of fuzzy set B^* . This is based on the observation of [29], as shown in Equation (3.1).

$$\frac{y_{2n} - y_{1n}}{x_{2n} - x_{1n}} = \frac{y_n^* - y_{1n}}{x_n^* - x_{1n}} \quad (3.1)$$

The normal point y_n^* is determined from Equation (3.1), as shown in Equation (3.2).

$$y_n^* = y_{1n} + \frac{(x_n^* - x_{1n}) \times (y_{2n} - y_{1n})}{x_{2n} - x_{1n}} \quad (3.2)$$

Step 2: Determine the left and right extreme points of the fuzzy set B^* . The areas of the left and the right sides from the normal point, as shown in Figure 3.3, are α_L and α_R , respectively. λ_L and λ_R denotes the left bottom and right bottom length of the fuzzy set B^* , respectively.

Figure 3.3: Left and right area.

The areas α_L and α_R of the consequent fuzzy set B^* are defined as follow in Equation (3.3).

$$\alpha_K(B^*) = \begin{cases} \alpha_K(A^*) \times \sum_{\substack{i=1, \\ \alpha_K(A_i) > 0}}^2 W_{A_i} \times \frac{\alpha_K(B_i)}{\alpha_K(A_i)}, & \text{if } \exists i \alpha_K(A_i) > 0 \\ \alpha_K(A^*), & \text{if } \forall i \alpha_K(A_i) = 0 \end{cases} \quad (3.3)$$

where:

$$K \in \{L, R\};$$

$\alpha_L(B^*)$ denotes the left area of B^* ;

$\alpha_R(B^*)$ denotes the right area of B^* ;

i representing the individual rules as shown in Figure 3.2; and

W_{A_i} is the normalized weight based on the distance measure between the observation and the rules with i representing the individual rules. W_{A_i} is defined by Equation (3.4).

$$W_{A_i} = 1 - \left| \frac{d(x_n^*, x_{in})}{\sum_{i=1}^2 |d(x_n^*, x_{in})|} \right| \quad (3.4)$$

where $d(a, b)$ represents the direct linear distance measured between point a and point b . In [29], the weight normalization is performed using the distance between the two extreme nearest rules as the denominator. This approach is applicable only if the distance between the observation and the selected rule is lesser than the distance between the two extreme nearest rules. However, when the distance between the observation and the selected rule is greater, the computed weight will be negative. In [34], to prevent negative weights, an additional check was performed. In the situation when all n -nearest rules appears on a single flank of the observation, the weightage will be set to 1. This infers that all n -nearest rules will share equal weightage on the derived consequent set. However, this concept is against the key idea where the observation should be more represented by the nearest rule. In [50], the denominator used to normalize the distance uses a predefined maximum distance that can be set to be larger than greatest distance between the observation and selected rule. Therefore, no negative weights will be computed and a higher weightage is given to the nearest rule.

The property that $W_{A_1} + W_{A_2} = 1$ must hold. The purpose of the equation is to ensure that the closer the rule is to the observation, the higher the weight it should be assigned to show the higher similarity. The area of the left ($\alpha_L(.)$) or the right

$(\alpha_R(.))$ side of the triangle is given in Equation (3.5).

$$\alpha_K(.) = (\lambda_K(.) \times h)/2 \quad (3.5)$$

If there exists a fuzzy rule whose area of the antecedent part is larger than zero, then the top equation of Equation (3.3) is used to infer the areas of the interpolated fuzzy set; otherwise, the bottom equation of Equation (3.3) is used when all the areas of the antecedent part of the given fuzzy rules are zero. This is illustrated in Figure 3.4.

Figure 3.4: Special case of triangular membership functions fuzzy interpolation reasoning.

In Figure 3.4, it shows the antecedents of two rules A_1 and A_2 , and the observation A^* . It can be observed that the right extreme point of x_{1r} and x_{2r} coincides with their normal point x_{1n} and x_{2n} , respectively, and both the right area of the rules fuzzy set, $\alpha_R(A_1)$ and $\alpha_R(A_2)$, are zero. In this scenario, when deriving the area $\alpha_L(B^*)$, the top equation of Equation (3.3) is used, and when deriving the area $\alpha_R(B^*)$, the bottom equation of Equation (3.3) is used instead.

Step 3: Derive the fuzzy interpolative reasoning result B^* . Using Equation (3.5), the two extreme points of B^* are derived as Equation (3.6) and Equation (3.7).

$$y_l^* = y_n^* - \lambda_l(B^*) \quad (3.6)$$

and

$$y_r^* = y_n^* + \lambda_r(B^*) \quad (3.7)$$

where $\lambda_l(B^*)$ and $\lambda_r(B^*)$ are $2\alpha_L(B^*)$ and $2\alpha_R(B^*)$, respectively, determined by re-

arranging (3.5) with $h = 1$ as shown in Figure 3.3. The left extreme point y_l^* and the right extreme point y_r^* of the fuzzy set B^* can be derived using Equation (3.8) and Equation (3.9), respectively.

$$y_l^* = y_n^* - 2\alpha_L(B^*) \quad (3.8)$$

and

$$y_r^* = y_n^* + 2\alpha_R(B^*) \quad (3.9)$$

Therefore, the triangular fuzzy interpolative reasoning result B^* can be represented as $(y_n^* - 2\alpha_L(B^*), y_n^*, y_n^* + 2\alpha_R(B^*))$.

3.1.2 Trapezoidal fuzzy sets

A trapezoidal fuzzy set A , as shown in Figure 3.5, can be denoted by $(x_l, x_{n1}, x_{n2}, x_r)$, where each variable represents the left extreme point, the left normal point, the right normal point and the right extreme point, respectively. The three steps required to derive the consequent of the trapezoidal fuzzy interpolative reasoning are:

Figure 3.5: Trapezoidal fuzzy set.

Step 1: Compute the composite normal points (CNP) of the trapezoidal fuzzy sets A_1 , A_2 , A^* , B_1 and B_2 using Equation (3.10).

$$CNP = \frac{\text{Left normal point} + \text{Right normal point}}{2} \quad (3.10)$$

Using A_1 of Figure 3.6 as an illustration, the CNP of A_1 will be $x_{1n} = ((x_{1n1} + x_{1n2})/2)$. With reference to Equation (3.2), the composite normal point of the conse-

Figure 3.6: Trapezoidal membership functions fuzzy interpolation reasoning.

quent fuzzy set B^* is derived as shown in Equation (3.11).

$$y_n^* = y_{1n} + \frac{(x_n^* - x_{1n}) \times (y_{2n} - y_{1n})}{(x_{2n} - x_{1n})} \quad (3.11)$$

Step 2: Compute the left normal point y_l^* and the right normal point y_r^* using Equation (3.12) and Equation (3.13), respectively.

$$y_l^* = y_n^* - \frac{d(y_{n1}^*, y_{n2}^*)}{2} \quad (3.12)$$

and

$$y_r^* = y_n^* + \frac{d(y_{n1}^*, y_{n2}^*)}{2} \quad (3.13)$$

where $d(y_{n1}^*, y_{n2}^*)$ represents the distance between the left normal point y_{n1}^* and right normal point y_{n2}^* of the consequent fuzzy set B^* , and is computed as shown in Equation (3.14).

$$d(y_{n1}^*, y_{n2}^*) = \begin{cases} d(x_{n1}^*, x_{n2}^*) \times \sum_{\substack{i=1, \\ d(x_{in1}, x_{in2}) > 0}}^2 W_{A_i} \times \frac{d(y_{in1}, y_{in2})}{d(x_{in1}, x_{in2})}, & \text{if } \exists d(x_{in1}, x_{in2}) > 0 \\ d(x_{n1}^*, x_{n2}^*), & \text{if } \forall d(x_{in1}, x_{in2}) = 0 \end{cases} \quad (3.14)$$

where W_{A_i} is the normalized weight depending on the distance measured between the

observation and the rules using Equation (3.4).

If there exists an antecedent fuzzy set of the given fuzzy rules having a top support larger than zero, then the top portion of Equation (3.14) is used; Else, the bottom portion of Equation (3.14) is used (that is, if all the top supports of the antecedent fuzzy sets of the given fuzzy rules are zero). This then represents the case where the trapezoidal fuzzy membership function collapses to a triangular fuzzy membership function, as shown in Figure 3.7. The collapse shows that the left and right normal points, x_{n1}^* and x_{n2}^* , respectively, are found to be at the same point as the composite normal point (CNP).

Figure 3.7: Trapezoidal fuzzy set collapses in becoming a triangular fuzzy set.

Step 3: Compute the left triangular area of B^* between y_l^* and y_{n1}^* and the right triangular area of B^* between y_{n2}^* and y_r^* , using Equation (3.3). Then, with Equation (3.5), the two extreme points of y_l^* and y_r^* can be deduced. Therefore the final trapezoidal fuzzy interpolative reasoning result of B^* is represented as $(y_{n1}^* - 2\alpha_L(B^*), y_{n1}^*, y_{n2}^*, y_{n2}^* + 2\alpha_R(B^*))$.

3.1.3 Gaussian fuzzy sets

Gaussian fuzzy sets are a special type of fuzzy set that covers the entire domain space. The tails of both end stretches infinitely; hence, when Gaussian fuzzy set is involved, a minimum membership value has to be defined as the end of the fuzzy set representation.

A Gaussian fuzzy membership function is specified by two variables; namely: the mean c and the standard deviation σ , with the function defined as $A = e^{-(x-c)^2/2\sigma^2}$,

as shown in Figure 3.8. The normal point of the Gaussian fuzzy set A is the mean of the Gaussian fuzzy membership function.

Figure 3.8: Gaussian fuzzy set.

With reference to Figure 3.9, x_{1n} , x_{2n} , x_n^* , y_{1n} , y_{2n} and y_n^* are the normal points of the fuzzy sets A_1 , A_2 , A^* , B_1 , B_2 and B^* , respectively, σ_{A_1} , σ_{A_2} , σ_{A^*} , σ_{B_1} , σ_{B_2} and σ_{B^*} are the standard deviations of the fuzzy sets A_1 , A_2 , A^* , B_1 , B_2 and B^* , respectively. The two steps necessary for deriving the fuzzy interpolation reasoning for fuzzy set B^* are:

Figure 3.9: Gaussian membership functions fuzzy interpolation reasoning.

Step 1: Derive the normal point y_n^* for fuzzy set B^* using Equation (3.2).

Step 2: Derive the standard deviation σ_{B^*} using Equation (3.15).

$$\sigma_{B^*} = \begin{cases} \sigma_{A^*} \times \sum_{\substack{i=1, \\ \sigma_{A_i} > 0}}^2 W_{A_i} \times \frac{\sigma_{B_i}}{\sigma_{A_i}}, & \text{if } \exists \sigma_{A_i} > 0 \\ \sigma_{A^*}, & \text{if } \forall \sigma_{A_i} = 0 \end{cases} \quad (3.15)$$

where W_{A_i} follows Equation (3.4) with the property that $W_{A_1} + W_{A_2} = 1$ holds.

In the situation when all antecedent of the rules are fuzzy singleton, the bottom portion of Equation (3.15) will be used, otherwise, the top portion of the equation will be used. With the normal point (step 1) and the standard deviation derived from Equation (3.15) (step 2), the Gaussian fuzzy membership function of the consequence fuzzy set B^* is $B^* = e^{-(x-b^*)^2/2\sigma_{B^*}^2}$.

3.1.4 Multiple antecedent fuzzy interpolative reasoning

This section discusses fuzzy interpolative reasoning with multiple antecedents. The trapezoidal fuzzy membership function will be used to illustrate the computations in deriving the fuzzy consequent under the multiple antecedent scenario, as shown in Figure 3.10. The following shows a rule base consisting of two rules and its observation with J antecedents.

Rule 1: If X_1 is A_{11} and X_2 is A_{12} and ... and X_J is A_{1J} Then Y is B_1 .

Rule 2: If X_1 is A_{21} and X_2 is A_{22} and ... and X_J is A_{2J} Then Y is B_2 .

Observation: X_1 is A_1^* and X_2 is A_2^* and ... and X_J is A_J^* .

Conclusion: Y is B^* .

The rule structure can be generally stated as Rule i : If X_1 is A_{i1} , X_2 is A_{i2} , ..., X_j is A_{ij} , Then Y is B_i where i represents rule and j represents the antecedent. In this section, only 2 rules and J antecedents will be discussed, therefore $i \in \{1, 2\}$ and $j \in [1, J]$. The following shows the three steps in deriving the fuzzy consequent set B^* .

Step 1: Compute the CNP x_{11n} , x_{21n} , x_{1n}^* , x_{12n} , x_{22n} , x_{2n}^* , y_{1n} , y_{2n} and y_n^* of the trapezoidal fuzzy sets A_{11} , A_{21} , A_1^* , A_{12} , A_{22} , A_2^* , B_1 , B_2 and B^* , respectively. Using the computed CNP, the normal point y_n^* of the consequent fuzzy set B^* will be

Figure 3.10: Multi-antecedent fuzzy interpolation reasoning.

computed using Equations (3.16)-(3.18).

$$w_{A_{ij}} = 1 - \left| \frac{d(x_{jn}^*, x_{ijn})}{\sum_{i=1}^2 |d(x_{jn}^*, x_{ijn})|} \right| \quad (3.16)$$

$$W_{A_i} = \frac{\sum_{j=1}^J w_{A_{ij}}}{J} \quad (3.17)$$

$$y_n^* = W_{A_1} y_{1n} + W_{A_2} y_{2n} \quad (3.18)$$

where:

$$0 \leq w_{A_{ij}} \leq 1;$$

$$w_{A_{1j}} + w_{A_{2j}} = 1;$$

$$W_{A_1} + W_{A_2} = 1;$$

$i = 1, 2$ representing two rules used; and

$j = [1, J]$, with J representing the number of antecedent variables available for each rule.

The distance measures $d(x_{jn}^*, x_{ijn})$ of Equation (3.16) are shown in Figure 3.10. The normalized weight of each rule is computed using Equation (3.17). This normalized weight is then used in Equation (3.18) to derive the CNP of the fuzzy consequent set B^* from the CNP of the consequent fuzzy sets of the rules.

Step 2: Compute the left and right normal points of the fuzzy set B^* using Equation (3.19) and Equation (3.20), respectively.

$$y_{n1}^* = y_n^* - \frac{d(y_{n1}^*, y_{n2}^*)}{2} \quad (3.19)$$

$$y_{n2}^* = y_n^* + \frac{d(y_{n1}^*, y_{n2}^*)}{2} \quad (3.20)$$

where

$$d(y_{n1}^*, y_{n2}^*) = \begin{cases} \left(\left(\sum_{j=1}^J d(x_{jn1}^*, x_{jn2}^*) \right) \times \left(\sum_{\substack{i=1, \exists ij \\ d(x_{ijn1}, x_{ijn2}) > 0}}^2 \left[W_{A_i} \times \frac{d(y_{in1}, y_{in2})}{\sum_{j=1}^J d(x_{ijn1}, x_{ijn2})} \right] \right) \right), & \text{if } \exists ij \, d(x_{ijn1}, x_{ijn2}) > 0 \\ \frac{\sum_{j=1}^J d(x_{jn1}^*, x_{jn2}^*)}{J}, & \text{if } \forall ij \, d(x_{ijn1}, x_{ijn2}) = 0 \end{cases} \quad (3.21)$$

The top portion of Equation (3.21) is used when at least one of the antecedents of the fuzzy rules is a trapezoidal fuzzy set. If all the fuzzy sets are triangular fuzzy sets, then the bottom portion of the Equation (3.21) is used. This is illustrated in Figure 3.7 where a trapezoidal fuzzy set is collapsed into a triangular fuzzy set with the left and right normal points of the top support collapsed onto the same point of the CNP.

Step 3: Compute the left and right area of the fuzzy set B^* , using Equation (3.22).

$$\alpha_K(B^*) = \begin{cases} \left(\sum_{j=1}^J \alpha_K(A_j^*) \right) \times \left(\sum_{\substack{i=1, \exists ij \\ \alpha_K(A_{ij}) > 0}} W_{A_i} \times \frac{\alpha_K(B_i)}{\sum_{j=1}^J \alpha_K(A_{ij})} \right), & \text{if } \exists ij \alpha_K(A_{ij}) > 0 \\ \frac{\sum_{j=1}^J \alpha_K(A_j^*)}{J}, & \text{if } \forall ij \alpha_K(A_{ij}) = 0 \end{cases} \quad (3.22)$$

where $K \in \{L, R\}$ represents the left and right area of fuzzy set B^* , respectively. Equation (3.22) is designed to handle fuzzy interpolation reasoning involving two rules having multiple antecedents. The top portion of Equation (3.22) is used when at least one area of the computed side of the antecedent of the fuzzy set is not zero. The bottom portion will be used when the areas of a specific side are all zero, as illustrated in Figure 3.4. The same method for deriving the four points required to specify a trapezoidal fuzzy set where $B^* = (y_{n1}^* - 2\alpha_L(B^*), y_{n1}^*, y_{n2}^*, y_{n2}^* + 2\alpha_R(B^*))$ is used to derive the multiple antecedent case.

3.1.5 Multiple fuzzy rules with multiple antecedent fuzzy interpolative reasoning

Next, fuzzy interpolative reasoning using multiple fuzzy rules with multiple antecedents, will be discussed. The reasoning scheme is an extension of the previous scheme by increasing the numbers of rules from two rules to I rules.

Rule 1: If X_1 is A_{11} and X_2 is A_{12} and ... and X_J is A_{1J} Then Y is B_1 .

Rule 2: If X_1 is A_{21} and X_2 is A_{22} and ... and X_J is A_{2J} Then Y is B_2 .

⋮

Rule I : If X_1 is A_{I1} and X_2 is A_{I2} and ... and X_J is A_{IJ} Then Y is B_I .

Observation: X_1 is A_1^* and X_2 is A_2^* and ... and X_J is A_J^* .

Conclusion: Y is B^* .

Figure 3.11 shows an example with the first observation A_1^* being flanked by three rules, having one rule on the left and two rules on the right, namely: A_{11} on the left, and A_{21} and A_{31} on the right of the first antecedent, and the second observation A_2^* being flanked by three rules, with one rule on the left and two rules on the right, namely: A_{12} on the left, and A_{22} and A_{32} on the right of the second antecedent. The

three fuzzy rules are: $A_{11} \wedge A_{12} \implies B_1$, $A_{21} \wedge A_{22} \implies B_2$ and $A_{31} \wedge A_{32} \implies B_3$. The three steps involved in deriving the fuzzy interpolative reasoning result B^* denoted by $(y_l^*, y_{n1}^*, y_{n2}^*, y_r^*)$, are as follows.

Step 1: Compute the CNPs $x_{11n}, x_{21n}, x_{31n}, x_{1n}^*, x_{12n}, x_{22n}, x_{32n}, x_{2n}^*, y_{1n}, y_{2n}$ and y_{3n} of the fuzzy sets $A_{11}, A_{21}, A_{31}, A_1^*, A_{12}, A_{22}, A_{32}, A_2^*, B_1, B_2$ and B_3 , respectively, using Equation (3.10). Computation of the consequent fuzzy set's intermediate normal point y'_n uses Equations (3.23)-(3.25).

$$w_{A_{ij}} = 1 - \left| \frac{d(x_{jn}^*, x_{ijn})}{\sum_{i=1}^I |d(x_{jn}^*, x_{ijn})|} \right| \quad (3.23)$$

$$W_{A_i} = \frac{\sum_{j=1}^J w_{A_{ij}}}{\sum_{i=1}^I \sum_{j=1}^J w_{A_{ij}}} \quad (3.24)$$

$$y'_n = \sum_{i=1}^I W_{A_i} y_{in} \quad (3.25)$$

where:

$$0 \leq w_{A_{ij}} \leq 1;$$

$$i = 1, 2, \dots, I;$$

$$j = 1, 2, \dots, J;$$

I denotes the number of rules; and

J denotes the number of antecedent variables.

In Equation (3.23), the individual distance is normalized according to the overall distance measured between the observation to all the fuzzy sets involved in the rule as illustrated in Figure 3.11. This normalization is to ensure that the weight distribution will not be affected by cases where the neighboring rules have very large distances.

In a multiple rule setting, a slight deviation in Equation (3.25) from the actual normal point is observed due to the unbalanced CNP ratio found on each side of the observation. This is illustrated in the example shown in Figure 3.11, which consists of two antecedents in each of the three rules. The observation A_j^* is found flanked by one fuzzy set A_{1j} on the left and two fuzzy sets A_{2j} and A_{3j} on the right, where j represents the antecedent. The normal points of both the given observations A_1^* and A_2^* are 5. The weights of the individual rules is derived from Equation (3.23)

Figure 3.11: Three rules and two antecedent fuzzy interpolation reasoning.

and Equation (3.24). In order to show the effect of the offset, the individual weights of each antecedent of each rule has to be applied to its normal point, as shown by Equation (3.26) and Equation (3.27).

$$w'_{A_{ij}} = \frac{w_{A_{ij}}}{\sum_{i=1}^n w_{A_{ij}}} \quad (3.26)$$

$$x'_{jn} = \sum_{i=1}^n w'_{A_{ij}} x_{ijn}. \quad (3.27)$$

The individual weights computed by Equation (3.26) across the first antecedent is $w'_{A_{11}} = 0.35$, $w'_{A_{21}} = 0.35$ and $w'_{A_{31}} = 0.3$, and across second antecedent is $w'_{A_{12}} = 0.3$, $w'_{A_{22}} = 0.4$ and $w'_{A_{32}} = 0.3$. With Equation (3.27), the normal point of the

intermediate observation fuzzy set at each antecedent can be derived. In the situation whereby only two rules are involved, the normal point intermediate fuzzy set will match onto the normal point of the observation perfectly. However, in this example, the normal point of the intermediate fuzzy sets are at $x'_{1n} = 6.2$ and $x'_{2n} = 5.8$. The derived intermediate normal points differ from the actual observation normal points, which are both at 5. Therefore a proposed shifting technique will be applied to the derived fuzzy consequent using Equation (3.25) to rectify the imbalance.

With the intermediate fuzzy set's normal point x'_{jn} from Equation (3.27), the individual antecedent shift ratio, δ_{a_j} , is derived using Equation (3.28).

$$\delta_{A_j} = \frac{d(x_{jn}^*, x_{jn}^{*'})}{d(x_{kn}, x_{1jn})} \quad (3.28)$$

where the direct distance between the intermediate fuzzy set's normal point x'_{jn} and actual observation's normal point x_{jn}^* is normalized against the span of the distance of the normal points of the fuzzy sets used. k indicates the furthest fuzzy set found in the antecedent and this distance is indicated in Figure 3.11 by $d(x_{31n}, x_{11n})$ and $d(x_{32n}, x_{12n})$. The ordering of the distance computation is important in this equation. The distance must be taken from the actual observation's normal point to the derived normal point so that the direction of the shift is preserved. This normalization is required to ensure that the shift maintains the consistency across different antecedent dimensionality. From the illustration in Figure 3.11, the δ of each antecedent is found to be $\delta_{A_1} = -0.171$ and $\delta_{A_2} = -0.1$. The final shift ratio δ_A is determined by taking the average of the individual shift ratios derived from Equation (3.28) across different antecedent, as shown in Equation (3.29).

$$\delta_A = \frac{\sum_{j=1}^J \delta_{A_j}}{J} \quad (3.29)$$

Thurs the final shift ratio for the example in Figure 3.11 is $\delta_A = -0.136$. The intermediate normal point of the output fuzzy set is computed as 14 by Equation (3.25) and is shown in Figure 3.11 as y'_n . The shift ratio derived from Equation (3.29) is

incorporated with Equation (3.25) as shown in Equation (3.30).

$$y_n^* = y_n' + \delta_A(d(y_{kn}, y_{1n})) \quad (3.30)$$

where k indicates the furthest fuzzy set used in the output domain. The shift ratio, δ_A , has to be applied to the range of the chosen output fuzzy sets, $d(y_{kn}, y_{1n})$, in order to determine the amount of shift required. In the example, the final derived consequent normal point, y_n^* , is at 12.64.

Step 2: This step is required for trapezoidal fuzzy sets as described in the previous section. The left normal point y_{n1}^* and right normal point y_{n2}^* of fuzzy set B^* are computed using Equation (3.12) and Equation (3.13), respectively, whereby $d(y_{n1}^*, y_{n2}^*)$ is defined in Equation (3.31).

$$d(y_{n1}^*, y_{n2}^*) = \begin{cases} \left(\sum_{j=1}^J d(x_{jn1}^*, x_{jn2}^*) \right) \times \left(\sum_{\substack{i=1, \exists ij \\ d(x_{ijn1}, x_{ijn2}) > 0}}^I \left[W_{A_i} \times \frac{d(y_{in1}, y_{in2})}{\sum_{j=1}^J d(x_{ijn1}, x_{ijn2})} \right] \right), & \text{if } \exists ij \ d(x_{ijn1}, x_{ijn2}) > 0 \\ \frac{\sum_{j=1}^J d(x_{jn1}^*, x_{jn2}^*)}{J}, & \text{if } \forall ij \ d(x_{ijn1}, x_{ijn2}) = 0 \end{cases} \quad (3.31)$$

where W_{A_i} is derived using Equation (3.24), and $\sum_{i=1}^n W_{A_i} = 1$ has to hold. The top portion of Equation (3.31) will be used if there exists a fuzzy set that has a trapezoidal shape with a top support. If no fuzzy set has a top support, that is they are all triangular fuzzy sets, then the bottom portion of Equation (3.31) will be used. In Figure 3.11, the bottom portion of the equation will hence be used. An example using trapezoidal fuzzy sets is shown in Figure 3.10.

Step 3: Compute the left and right triangular areas of the fuzzy set B^* using Equation (3.32).

$$\alpha_K(B^*) = \begin{cases} \left(\sum_{j=1}^J \alpha_K(A_j^*) \right) \times \left(\sum_{\substack{i=1, \exists ij \\ \alpha_K(A_{ij}) > 0}}^I W_{A_i} \times \frac{\alpha_K(B_i)}{\sum_{j=1}^J \alpha_K(A_{ij})} \right), & \text{if } \exists ij \ \alpha_K(A_{ij}) > 0 \\ \frac{\sum_{j=1}^J \alpha_K(A_j^*)}{J}, & \text{if } \forall ij \ \alpha_K(A_{ij}) = 0 \end{cases} \quad (3.32)$$

where $K \in \{L, R\}$ represents the left and right area of fuzzy set B^* , respectively. The top portion of Equation (3.32) is used when at least one area of the computed side of the antecedent of the fuzzy set is not zero. The bottom portion will be used when all the areas of a specific side are zero, as illustrated in Figure 3.4. With $\alpha_L(B^*)$ and $\alpha_R(B^*)$ being computed by Equation (3.32), the consequent fuzzy set B^* hence is represented as $(y_{n1}^* - 2\alpha_L(B^*), y_{n1}^*, y_{n2}^*, y_{n2}^* + 2\alpha_R(B^*))$.

3.1.6 Fuzzy rule extrapolation

Figure 3.12: Extrapolation with two rules and two antecedent fuzzy interpolation reasoning.

In the previous section, the formulation for handling multiple antecedent rules was shown. The formulation is directly applicable for fuzzy extrapolation. Figure 3.12

shows a simple extrapolation situation whereby all the existing fuzzy rules are only available on one side of the observation. The weightage of each rule and the intermediate output y'_n is computed by Equation (3.24) and Equation (3.25), respectively. The intermediate output y'_n is found to lie at the point 17.24. A shift ratio is required to shift the intermediate point to a more correct position. In order to compute the shift ratio, the intermediate position of the observations in each antecedent is required. The intermediate positions are computed using Equation (3.26) and Equation (3.27), where the intermediate position of the first and second antecedent is 8.57 and 8.33, respectively. The shift ratios of the individual antecedents are then computed using Equation (3.28), and found to be $\delta_{A_1} = -3.57$ and $\delta_{A_2} = -1.67$, which in turn are used to derive the final shift ratio $\delta_A = -2.62$, using Equation (3.29). The final consequent normal point is $y_n^* = 12$ derived using Equation (3.30).

The ability to perform extrapolation is dependant on the shift technique. Without the shift, the intermediate points of the antecedents, x'_{1n} and x'_{2n} , or intermediate normal point of the consequent y'_n are found in between the fuzzy sets of the existing rules, which does not correspond to the actual positions of the observations. The remaining steps (step 2 and step 3) for deriving the final output fuzzy set, are identical to that for interpolation, as described by Equation (3.31) and Equation (3.32), respectively.

Thus the proposed technique is suitable for both fuzzy interpolation and extrapolation.

3.2 Fuzzy Rule Interpolation Results

This section will compare the proposed technique with three established methods proposed as CCL method [29], HS method [50] and CHYC method [33] which were reported capable of handling multiple antecedent rules. The naming of the methods was adopted from initials of the authors. In this in sync with the naming convention used in the literature. These three methods use analogical techniques whereby an intermediate rule is created before deriving the final output.

Table 3.2 shows the attributes of the fuzzy rules and the observation in this ex-

ample. Figure 3.13 shows the plot of the fuzzy rules and resulting fuzzy set B^* of all three implementations.

Table 3.2: Fuzzy interpolation result for the multiple antecedent rules example.

Attribute Values	Methods	Fuzzy Interpolative Reasoning Result B^*
$A_{11} = (0,1,3)$ $A_{21} = (8,9,10)$ $A_{31} = (11,13,14)$	HS Method [50]	(4.76,6.13,7.13)
$A_{12} = (1,2,3)$ $A_{22} = (7,9,10)$ $A_{32} = (11,12,13)$	CCL Method [29]	(6.19,7.65,8.96)
$B_1 = (1,2,3)$ $B_2 = (9,10,11)$ $B_3 = (12,13,14)$	CHYC Method [34]	(6.56,7.63,8.94)
$A_1^* = (3.5,5,7)$ $A_2^* = (5,6,7)$	Proposed Method	(4.79,6.22,7.54)

Figure 3.13: Fuzzy interpolation for the multiple antecedent rules example.

With Equation (3.23) and Equation (3.24), the values of each rule weight are

$W_{A_1} = 0.36$, $W_{A_2} = 0.38$ and $W_{A_3} = 0.26$ hence the intermediate consequent fuzzy set's normal point is given as $y_n' = 7.9$ by using Equation (3.25). According to Equation (3.27), the normal point of the intermediate fuzzy set of each observation are $x'_{1n} = 7$ and $x'_{2n} = 7.38$. The shift ratio is next computed using Equation (3.28) giving $\delta_{A_1} = -0.167$ and $\delta_{A_2} = -0.138$, and the average shift ratio is $\delta_A = -0.1525$ from Equation (3.29). Lastly, the actual fuzzy consequent normal point is derived using Equation (3.30) resulting as $y_n^* = 6.22$. According to Equation (3.32), the area of the left and right fuzzy consequent set B^* can be derived. The left and right areas are $\alpha_L(B^*) = 0.715$ and $\alpha_R(B^*) = 0.66$, respectively. With all the required information, the fuzzy consequent set can be derived as $B^* = (4.79, 6.22, 7.54)$

As observed from the plot in Figure 3.13 and from Table 3.2, the results of the HS method and the proposed method are fairly similar. However, the CCL method and CHYC method are slightly skewed. This is mainly because of CCL method and CHYC method did not consider the unbalanced rule situation whereby the intermediate fuzzy set derived is slightly offset from its actual observation position. The computation used in the HS method uses the whole spectrum of existing fuzzy rules for weight normalization which considers all existing fuzzy sets across the domain irregardless of whether or not the fuzzy set is being used for interpolation. This will be further discussed in the next section where an example of fuzzy extrapolation is presented.

3.3 Fuzzy Rule Extrapolation Results

Fuzzy extrapolation is required when the existing rules are on a single side of the observations. With reference to Figure 3.11, by removing fuzzy sets A_{11} , A_{12} and B_1 , we will have the situation whereby fuzzy extrapolation is required. The remaining existing fuzzy sets all lie on the right side of the observations.

Table 3.3 shows the attribute values of the fuzzy rules, and the results from the HS method, CHYC method and the proposed method. The CCL method is not compared in this example because an invalid weight will be computed during the computation, therefore this method is not applicable in fuzzy extrapolation operation. Figure 3.14 shows the fuzzy rules and extrapolated results from both methods.

Table 3.3: Fuzzy extrapolation result for the multiple antecedent rules example.

Attribute Values	Methods	Fuzzy Reasoning Result	Interpolative Result B^*
$A_{21} = (8,9,10)$ $A_{31} = (11,13,14)$ $A_{22} = (7,9,10)$ $A_{32} = (11,12,13)$ $B_2 = (9,10,11)$ $B_3 = (12,13,14)$ $A_1^* = (3.5,5,7)$ $A_2^* = (5,6,7)$	HS Method [50] CHYC Method [34] Proposed Method	(5,6,7) (10.563,11.5,13) (6.185,7.01,8.51)

Figure 3.14: Fuzzy extrapolation for the multiple antecedent rules example.

The rule weights $W_{A_2} = 0.67$ and $W_{A_3} = 0.33$ are derived from Equation (3.23) and Equation (3.24). The normal points of the intermediate fuzzy sets are $x'_{1n} = 10.33$ and $x'_{2n} = 10$, from Equation (3.27). The normal point of the intermediate consequent fuzzy set, derived from Equation (3.25), is at $y'_n = 10.99$. The final shift ratio using Equation (3.29) is found to be $\delta_A = -1.33$, and hence the normal point of

the consequent fuzzy set is computed to be $y_n^* = 7.01$. The left and right triangle areas are $\alpha_L(B^*) = 0.4125$ and $\alpha_R(B^*) = 0.75$, respectively. Therefore using the proposed method, the final consequent fuzzy set is represented as $B^* = (6.185, 7.01, 8.51)$.

From the plot in Figure 3.14, it can be observed that both the HS method and the proposed method are able to perform extrapolation. However, The CHYC method derived a consequent fuzzy set within the rules. The resultant fuzzy set is displaced from the expected location. Therefore CHYC method is unable to perform extrapolation. It can also be seen that in terms of closeness, the extrapolated fuzzy set derived using the HS method is a better fit according to the observations given. This is because the HS technique uses a weight normalization method that uses a predefined range, which in this experiment was the range given in the interpolation example in the previous section. This shifts the resulting fuzzy set derived by HS method closer to the observations. In contrast, the proposed method uses the attributes from the existing rules to determine the shift instead of predefined knowledge, and hence the extrapolated result is more conservative. The proposed method can be easily extended to evolving fuzzy neural networks where no prior knowledge of the domain range is given.

3.4 Summary

In this chapter, two changes were made to the inspired method of [29]. Firstly, a change in the weightage formulation to allow higher flexibility in computing the weights of each n -nearest rules without the fear of having the distance between the observation and the rules are greater than the overall distance computed. In [29] the overall distance is computed as the greatest distance of the n -nearest rules. When in the situation of extrapolation, the distance between the observation and the rules may be greater than the overall distance computed, which will result in a computation error of the weights. In [34], a prior check will be performed to identify the location of the observation against all n -nearest rules. If the observation lies beyond the rules, the weightage will be assigned as one. This weight assignment is against the idea where the observation will be more similar to the nearest rule. Therefore this weight assignment is unfair. In [50], the overall distance is predefined. This ensures

that the weight computation will be correct. However, in the intention of extending this technique to evolving fuzzy systems, where no prior knowledge of domain is known, this predefined overall distance is not feasible. In the proposed method, it can handle extrapolation situation where [29] is not able to handle, it gives the respective weightage to the nearest rule instead of assigning equal weightage as shown in [34], and it allows dynamic computation of the distance value which is suitable in the context where no prior knowledge should be known instead of a predefined distance used in [50].

Secondly, a shift process was introduced. This process rectifies the imbalance situation of either when more than two nearest rules were selected for interpolation or when an extrapolation was required. This process is similar to the method used in [50]. However, in their method, the maximum consequent domain distance was predefined. In the proposed method, the maximum consequent domain distance was computed from the existing rules and no prior knowledge of the domain information was required. This allows the proposed method to be extended to an evolving fuzzy system more suitably than [50]. The importance of this shift process is shown in the example discussed in the previous section.

The discussions on fuzzy interpolation and extrapolation till date were mainly focusing on the Mamdani fuzzy model where the main advantage of this model focuses on the interpretability of the fuzzy consequent. In the following chapters, the proposed work in this chapter will be extended to a TSK fuzzy model whereby accuracy will be the models focus.

Chapter 4

Offline Learning Fuzzy Neural Networks

Fuzzy neural networks (FNN) can be broadly classified under two categories of learning; namely: online and offline, as mentioned in Table 2.3 of the literature review section. The ability of the learning mechanism to identify fuzzy labels and rules to describe the data distribution or the trend in a domain has brought about many different usages, ranging from control systems or classifiers to nonlinear time series predictors. Different learning mechanisms have been applied to different problem domains.

In Section 2.1.4, the motivation for implementing fuzzy interpolation was discussed, highlighting the lack of reasoning capability in conventional FLS when a sparse rule base exists which may be due to rule pruning or the sparse nature of the data source. This has motivated the incorporation of fuzzy interpolation and extrapolation (FIE) within different FNN to overcome certain data distribution problems.

In this chapter, a fuzzy neural network will be proposed and it will be known as the fuzzy interpolation and extrapolation (FIE) FITSK model (FIE-FITSK) that uses the batch learning approach for training (offline).

4.1 Fuzzy input Takagi-Sugeno-Kang (FITSK) model

The fuzzy input Takagi-Sugeno-Kang model [96] was proposed by Quah and Quek in 2006 that incorporates TSK FLS in a CMAC structure with evenly distributed resolution. The FITSK framework was chosen due to its suitability in adaptive system learning and nonlinear system estimation. FITSK employs a TSK FLS which aims to achieve precision instead of linguistic result, and the CMAC structure is highly suitable for controller model. Figure 4.1 shows the FITSK structure that has been implemented by [96].

Figure 4.1: FITSK structure.

A generalized type-1 TSK rule can be described by a fuzzy IF-THEN rule which

represents the input-output relations of a system, as shown in Equation (4.1).

$$R_k : \text{ IF } X_{1k} \text{ is } x_1 \text{ and } X_{2k} \text{ is } x_2, \text{ THEN output is } y_k = ax_1 + bx_2 + c \quad (4.1)$$

where:

R_k represents rule number k ;

X_{1k} and X_{2k} represents the first and second antecedent of rule k ;

x_1 and x_2 represents the input states;

$y_k = ax_1 + bx_2 + c$ represents the output function of rule k ; and

a , b , and c are the coefficients of the output function.

The FITSK model consists of the CMAC layer which contains the input and linguistic operations, the localized fuzzy rule computation layer, and finally the output/summation layer.

4.1.1 CMAC Layer

The CMAC layer consist of two operations; namely:

- Input fuzzification.
- Input linguistic.

The given input is being fuzzified and represented as a fuzzy membership function. This fuzzy membership function is then used to compute the fuzzy subethood for each corresponding fuzzy linguistic label. The following will describe the operations in detail.

Input Fuzzification

$$Z_{IF_i} = \mu_i(x_i) \quad (4.2)$$

Equation (4.2) illustrates the fuzzification of the input x_i with $\mu_i(\cdot)$ which is the fuzzy membership function of the input fuzzifier node IF_i . The input x_i is fuzzified by the Gaussian membership function by taking the input as the center of the membership function. This input is the maximum value and it will decrease when moving away from the center. Equation (4.3) is derived by substituting $\mu_i(\cdot)$ with the Gaussian

formula.

$$Z_{IF_i} = \mu_i(x_i) = \exp \frac{-(x-x_i)^2}{2\sigma_i^2} \quad (4.3)$$

where:

x spreads over the universe of possible ranges of x_i ; and

σ_i is the standard deviation of the Gaussian membership function.

The σ_i also reflects the width of the membership function, and is dependant on the accuracy and generalization to be achieved by a specified system.

Input Linguistic

This node takes the fuzzy membership function, described by Equation (4.3), as the input and computes the fuzzy subsethood measure for each corresponding fuzzy linguistic label. Let $s_{(i,j)}$ represents the center of the j^{th} receptive field of the i^{th} input node $IL_{i,j}$. The computations can be simplified from Equation (4.3) into Equation (4.4).

$$Z_{IL_{i,j}} = \mu_{i,j}(Z_{IF_i}) = \exp \frac{-(s_{(i,j)}-x_i)^2}{2\sigma_i^2} \quad (4.4)$$

where $s_{(i,j)}$ is evenly distributed over the value range of the input dimension i .

The resolution of network r_i is computed using Equation (4.5).

$$r_i = \frac{range_i}{nl_i - 1} \quad (4.5)$$

where $range_i$ is the value range of the input dimension i ; and nl_i is the total number of receptive fields in dimension i . The value of r_i is used in determining the spread of the receptive fields across a dimension.

The standard deviation, σ_i , of Equation (4.4) is defined by Equation (4.6)

$$\sigma_i = sr^{ld} \times \frac{range_i}{5} \quad (4.6)$$

where:

sr^{ld} is the shrinking rate: $sr \in (0, 1]$;

ld is the degree of localization; and

$range_i$ is the value range of the input dimension.

4.1.2 Localized Fuzzy Rule Computation Layer

There are three operations within this layer; namely:

- Conjunction operation.
- Normalization operation.
- Rule output computation.

The following sections describe each of the operations within this layer.

Conjunction operation

The conjunction operation computes the conjunction of all incoming signals and generates the corresponding weights. Equation (4.7) shows a minimum inference of the conjunction of all incoming signals from the CMAC layer to compute the output Z_{C_k} of node C_k , where k represents the index of the rule. Product inference can also be used instead.

$$Z_{C_k} = \bigcap_{i=1}^I Z_{IL_{i,j}} \quad (4.7)$$

Normalization operation

The normalization layer normalizes the weights from the conjunction operation. The normalized weight, Z_{N_k} , of node N_k is computed from Equation (4.8)

$$Z_{N_k} = \frac{Z_{C_k}}{\sum_{k=1}^K Z_{C_k}} = w_k \quad (4.8)$$

where w_k is the normalized weight of the node C_k by taking the individual output from Equation (4.7) and normalized with the summation of all the outputs Z_{C_k} .

Rule output computation

The rule output implements a local model of the TSK rule that produces an output with the normalized weights. Equation (4.9) computes the weighted output, Z_{F_k} ,

$$Z_{F_k} = w_k C_{F_k} \quad (4.9)$$

where C_{F_k} is represented as a zero-order function of the local model in the node F_k , and is computed with the normalized weight w_k .

4.1.3 Output Layer

The output layer computes the output by summing the weighted results from the localized fuzzy rule computation layer using Equation (4.10).

$$y_k = Z_S = \sum_{k=1}^K Z_{F_k} \quad (4.10)$$

4.1.4 FITSK Learning Scheme

CMAC learns the input-output relation by changing the function of the cells, C_{F_k} , is determined using Equation (4.11).

$$C_{F_k}(t+1) = C_{F_k}(t) + \lambda \alpha^k e(t) \quad (4.11)$$

where

$$e(t) = y_{desired}(t) - y_{predicted}(t) \quad (4.12)$$

and:

k is the rule index;

λ is the learning parameter;

α^k is the degree of matching of the k -th fuzzy IF-THEN rule; and

$e(t)$ is the error between the desired output and the predicted output at time t .

The weight of the cell, b_j^k , is used in the computation for rule output Z_{F_k} of Equation (4.9).

4.2 Fuzzy Interpolation and Extrapolation FITSK (FIE-FITSK) Model

In the FITSK model, a boundary exists in the form of a predefined range as described in Equation (4.5). This boundary is required by the CMAC structure initialization. Although FITSK is predefined to be uniformly spaced within the boundary with the Gaussian fuzzy sets covering all internal space of the CMAC structure, not all cells within this structure are being trained by the given training data. Therefore a sparse rule base may still exist and a null inference may still result. Additionally, when an observation is found to be beyond the boundary of the CMAC structure, a null inference may again result.

To address this boundary limitation, it is proposed to incorporate Fuzzy Interpolation and Extrapolation (FIE) with FITSK [132]. The FIE method discussed in Chapter 3 is used to derive the proposed FIE-FITSK model. However, the method in Chapter 3 is modeling a Mamdani fuzzy model. Therefore, the extension of this method will be extended to a TSK model in this section. The steps in achieving the fuzzy interpolated or extrapolated result are illustrated in Figure 4.2, which shows a single dimension FIE-FITSK structure. The inference process will be described in the following sections.

4.2.1 CMAC Layer

In this layer, a distance measure between the n -nearest rule and the observation will be computed. This distance measure will be used as the weight that is crucial in deriving the final output.

In Figure 4.2, rule nodes are evenly distributed across the input dimension, with their centers represented as $A_{i,j}$ which represents the center of the j^{th} dimension of the i^{th} rule node. The direct distances between the observation (x_j^*) and the center of the rule nodes ($A_{i,j}$) are computed, represented as $d(x_j^*, A_{i,j})$. Only n -nearest rules distance measure will be computed, where in the figure shows an example of 3-nearest rules ($A_{2,1}$, $A_{3,1}$, and $A_{4,1}$) being computed.

Figure 4.2: Fuzzy interpolation and extrapolation FITSK (FIE-FITSK) structure

CMAC has a lattice-based structure and with the assumption that every dimension j is evenly distributed with the same number of rule nodes i , hence a total of i^j rule nodes will be available. Every intersection of the lattice will represent a rule. As the distance computation is computed at every dimension, $n * j$ number of distance measure will be computed.

4.2.2 Localized Fuzzy Rule Computation Layer

The weightage $w_{A_{n,j}}$ for each n nearest rule on each dimension is computed by Equation (4.13) using the derived distance measure at the CMAC layer.

$$w_{A_{n,j}} = 1 - \left| \frac{d(x_j^*, A_{n,j})}{\sum_{i=1}^n |d(x_j^*, A_{i,j})|} \right| \quad (4.13)$$

The weight $w_{A_{n,j}}$ on each dimension is normalized across all j dimension to derive the individual normalized rule weight W_{A_n} as shown in Equation (4.14).

$$Z_{C_n} = W_{A_n} = \frac{\sum_{j=1}^J w_{A_{n,j}}}{\sum_{i=1}^n \sum_{j=1}^J w_{A_{i,j}}} \quad (4.14)$$

where A_n represents the selected rule. A total of n^j rules will be involved in the selection for the n -nearest rules instead of the entire rule base of i^j , where $n \subset i$. The n -nearest rules with the largest weight will be selected.

Each rule output will be computed with the normalized rule weights as shown in Equation (4.15).

$$Z_{F_n} = W_{A_n} C_{F_k} \quad (4.15)$$

where C_{F_k} is represented as a zero-order function of the local model in the node F_k .

4.2.3 Output Layer

The output Z_S of the model can be derived using Equation (4.16) which simply sums up the n -nearest rules output. However, this output is an intermediate result, which as explained in Section 3.1.5 requires a shift operation in order to generate a plausible result.

$$Z_S = y'_n = \sum_{n=1}^N Z_{F_n} \quad (4.16)$$

In order to derive the shift ratio, intermediate observations have to be computed for each dimension. This intermediate position is to verify the amount of shift that has been incurred due to the possible imbalance distribution of n -nearest rules. An intermediate weight for each dimension, $w'_{A_{n,j}}$, is derived by Equation (4.17) using the individual weight, $w_{A_{n,j}}$ from Equation (4.13).

$$w'_{A_{n,j}} = \frac{w_{A_{n,j}}}{\sum_{i=1}^n w_{A_{n,j}}} \quad (4.17)$$

The intermediate position of the observation is computed using Equation (4.18), and subsequently the individual dimension's shift ratio, δ_{x_j} , is computed using Equation (4.19).

$$x'_{n,j} = \sum_{i=1}^n w'_{A_{n,j}} A_{n,j} \quad (4.18)$$

$$\delta_{x_j} = \frac{d(x_{n,j}^*, x'_{n,j})}{d(A_{mjn}, A_{ljn})} \quad (4.19)$$

where $d(A_{mjn}, A_{ljn})$ represents the distance between the n -nearest rules of the dimension j , and m and l represents the last and first n -nearest rule, respectively. In the example shown in Figure 4.2, m and l will be 4 and 2, respectively.

Using the individual shift ratio δ_{x_j} , the final shift ratio δ_x is computed using Equation (4.20).

$$\delta_x = \frac{\sum_{j=1}^J \delta_{x_j}}{J} \quad (4.20)$$

A distance measure $d(y_{mn}, y_{ln})$, where m and l represents the last and first n -nearest rule respectively, is required to compute the output range required by the shift process. This distance measure can be computed during the process when the output of each rule is being computed in the localized fuzzy rule computation layer by Equation (4.15). The output range is computed as the distance of the maximum output of the n -nearest rules, C_{F_K} , where K will be the indexes of the n -nearest rules.

The intermediate output derived by Equation (4.16) needs to be shifted as shown in Equation (4.21), hence deriving the final output, y_n^* , of the process.

$$y_{predicted}(t) = y_n^* = y'_n + \delta_x(d(y_{mn}, y_{ln})) \quad (4.21)$$

4.2.4 FIE-FITSK Learning Scheme

CMAC learns the input-output relation by changing the function of the cells, C_{F_k} , is determined using Equation (4.22).

$$C_{F_n}(t+1) = C_{F_n}(t) + \lambda W_{A_n} e(t) \quad (4.22)$$

where

$$e(t) = y_{desired}(t) - y_{predicted}(t) \quad (4.23)$$

and:

n is the rule index;

λ is the learning parameter;

W_{A_n} is the degree of matching of the n -th fuzzy IF-THEN rule derived by Equation (4.14); and

$e(t)$ is the error between the desired output and the predicted output at time t .

4.3 Experiments and Results

In this section, the proposed model of FIE-FITSK will be compared against other existing models and its originating model, FITSK. This section will be split into two sub-sections, namely: accuracy testing and sparse handling.

The prediction accuracy of the proposed FIE-FITSK will be tested on two commonly used time series problems, Mackey-Glass time series prediction and non-linear system identification problem, against other existing architectures, which includes evolving Takagi Sugeno (eTS) [11], simplified eTS (simpl_eTS) [10], Self-adaptive Fuzzy Inference Network (SaFIN) [98], Self-reorganising Fuzzy Associative Machine (SeroFAM) [110], Sequential Probabilistic Learning for Adaptive Fuzzy Inference System (SPLAFIS) [90], Generic Evolving NEuro-Fuzzy Inference System (GENEFIS) [95], Meta-cognitive Neuro-Fuzzy Inference System (McFIS) [104], Evolving Fuzzy Neural Network (EFuNN) [60], Dynamic Evolving Neural-Fuzzy Inference System (DENFIS) [61], and Adaptive Neuro-Fuzzy Inference System (ANFIS) [52].

The second part of this section shows the ability of the proposed FIE-FITSK in handling drift/shift against its originating variation of FITSK. This is to show the inference capability of the FIE-FITSK in computing plausible results where FITSK is unable to perform.

The root mean squared error (RMSE) will be the main measurements used for performance evaluation. The mathematical definition is described in Equation (4.24).

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (o_n^d - o_n)^2} \quad (4.24)$$

where:

o_n^d is the expected output;

o_n is the deduced output from the network;

N is the total sample size.

The better performance is evaluated as having a smaller RMSE. The RMSE value is dependant on the output range; hence it is not directly comparable across different problem domain.

4.3.1 Prediction accuracy

In this section, two commonly used time series problem will be used to test the accuracy of the proposed FIE-FITSK model. The proposed model is implemented based on approximation, therefore precise accuracy of the work may not be superior to other models. The main objective is to show that the proposed model will be able to perform fairly well.

Mackey-Glass time series problem

A time series is a sequence of vectors, $\vec{x}(t)$, $t = 0, 1, \dots$, where t represents elapsed time. Time series prediction is an important component relevant to diverse fields of applications ranging from financial and business trend-spotting to electric power consumption. The Mackey-Glass time series is often used in time series prediction and is defined by Equation (4.25).

$$\dot{x}(t) = \frac{ax(t - \tau)}{1 + x^{10}(t - \tau)} - bx(t) \quad (4.25)$$

In this experiment the values of the variables in the time series are $a=0.2$, $b=0.1$, time lag $\tau=17$ and $x(0) = 1.2$, as shown in Equation (4.26), which was originally proposed as a control model in white blood cell production [78].

$$\frac{dx}{dt} = \frac{0.2x(t - 17)}{1 + x^{10}(t - 17)} - 0.1x(t) \quad (4.26)$$

A total of 6000 observations were generated and the selected observation range of

Table 4.1: Mackey-Glass: RMSE result for all models

Model	Learning mode	No. of rules	Test RMSE
eTS [11]	Online	99	0.0806
simpl.eTS [10]	Online	21	0.0851
SaFIS [98]	Online	21	0.0860
SeroFAM [110]	Online	52	0.0781
SPLAFIS [90]	Online	30	0.0632
GENEFIS [95]	Online	18	0.0634
McFIS [104]	Online	6	0.0068
EFuNN [60]	Semi-Online	193	0.0908
DENFIS [61]	Semi-Online	58	0.0625
ANFIS [52]	Offline	81	0.0289
FITSK [96]	Offline	81	0.0602
FIE-FITSK	Offline	81	0.0568

$201 \leq t \leq 3200$ and $5001 \leq t \leq 5500$, uniformly distributed within the range of $[0.4, 1.4]$, were used as the training and testing data set, respectively. The data set used in the prediction is formed up with $[x(t - 18), x(t - 12), x(t - 6), x(t)]$ as the inputs to predict the output of $x(t + 85)$. A list of existing techniques were used to benchmark the results accuracy as shown in Table 4.1.

In the table above, there are three different learning modes, namely: Semi-Online, Online and Offline. A model is referred to as semi-online mode because data normalisation had to be performed before the inferencing and learning process. This requires the entire data to be available at the beginning of the process so that the data can be normalised. Online mode refers to no prior knowledge of the problem domain when data is presented to the model for inferencing and learning. Offline (Batch) mode refers to the availability of the entire data set at the start of the learning, and these data will be repeatedly fed back to the model to refine the rule parameters. In both semi-online and online modes, data are only presented once for both inferencing and learning.

The rule base of the online and semi-online models shows a relatively smaller number of rules as compare to the offline models. This is because most of the online and semi-online models incorporate rule pruning mechanism which removes rules that are not frequently used. Both eTS and EFuNN do not include rule mechanism; hence contributing to a larger number of rules. While in ANFIS, FITSK and FIE-FITSK no rule pruning mechanism was incorporated.

A 100 training cycle was performed by ANFIS, FITSK, and FIE-FITSK with identical number of rules. All three models used a uniform spread of membership function across each input dimension, therefore every intersection across different input dimension represents a rule. The number of rules constructed is based on the number of predefined membership function across each dimension. In this experiment, the number of membership function at each dimension is defined as 3 which will form 81 rules. 2 nearest rule were used by FIE-FITSK to derive a prediction. The RMSE of all three offline models as shown in Table 4.1 are better than most of the other models. FIE-FITSK has attained a better result compared to FITSK but poorer result compared to ANFIS. As the size of the rule base is fixed at 81 for the offline models, it does not mean that a large rule base will produce better results. The rule base for eTS and EFuNN are larger than the offline models, however their results are much poorer.

In terms of accuracy, the best result was produced by McFIS which used a novel self-regulatory learning mechanism that decides what to learn, how to learn, and when to learn. The size of the rule base at the end was also the smallest. This model outperforms all other models significantly with its novel learning mechanism.

Non-linear system identification problem

This experiment investigates the ability to approximate a nonlinear dynamic plant with non-time-varying characteristics as described in [58] and [98]. The plant to be learnt is defined by the difference equation as shown in Equation (4.27).

$$y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t) \quad (4.27)$$

where:

$u(t) = \sin(2\pi t/100)$ is the current input signal; and

$y(t)$ is the current output signal of the system.

Following the description in [115], 50,000 training and 200 testing data pairs are generated with initial conditions of $(u(t), y(t))$ as $(0, 0)$, where $u(t) \in [-1.0, 1.0]$ and

Table 4.2: Non-linear system identification problem: RMSE result for all models

Model	Learning mode	No. of rules	Test RMSE
eTS [11]	Online	19	0.0082
simpl_eTS [10]	Online	18	0.0122
SaFIS [98]	Online	8	0.0116
McFIS [104]	Online	5	0.0023
EFuNN [60]	Semi-Online	18	0.0578
ANFIS [52]	Offline	16	0.0048
FITSK [96]	Offline	16	0.0041
FIE-FITSK	Offline	16	0.0166

$y(t) \in [-1.5, 1.5]$. Table 4.2 shows the result of the proposed FIE-FITSK model compared to other fuzzy neural networks (FNN) that were reported in [115] and [104].

In this experiment, ANFIS, FITSK and FIE-FITSK used a uniform spread of membership function across each input dimension, therefore every intersection across different input dimension represents a rule. The number of membership function at each dimension is defined as 4 which will form 16 rules as shown in Table 4.2. A 100 training cycle was performed in this experiment for these three models.

The accuracy of this experiment is based on the RMSE of the test dataset. The number of rules used are highly comparable other than SaFIS and McFIS. The model that had achieved the lowest RMSE is the McFIS model followed by the FITSK and ANFIS models. The test RMSE of FIE-FITSK is marginally higher as compared to simpl_eTS and SaFIS. However, in Figure 4.3 that shows the predicted test result from FIE-FITSK plotted against the actual test result, the error incurred is minimal.

The results shown in the above experiments has shown that although FIE-FITSK was unable to outperform in some cases, but the accuracy of prediction is still reasonably accepted. The following section will discuss the ability of the FIE-FITSK in handling concept drift and shift as compared to it's originating variation of FITSK.

4.3.2 Handling concept drift/shift

Two experiments will be used in this section to illustrate the drift/shift handling capability of the proposed model; namely: dynamic non-linear system identification problem, and diabetes modelling. The purpose of these experiments is to show that

Figure 4.3: Non-linear system prediction result from FIE-FITSK.

with a fix trained offline model, the proposed model will be able to handle drift or shift in new data presented to the model. The testing data will include data points that was not presented during the training. In a typical offline model, they are not able to produce results that were not presented during the training.

Dynamic non-linear system identification problem

In this section, the experiment will have four levels of varying disturbance added at a specific time range making the original non-linear system defined in Equation (4.27) to behave dynamically. The FITSK will be used in these experiments to show that the proposed model with the drift/shift handling capability is able to produce satisfactory result where the former is unable to. The results from the ANFIS model will also be presented. The properties of the dynamic non-linear system are described in Equation (4.28).

$$y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t) + f(t) \quad (4.28)$$

where:

$u(t) = \sin(2\pi t/100)$ is the current input signal;

$y(t)$ is the current output signal of the system; and

$f(t)$ is the added disturbance signal defined as follows:

Table 4.3: Dynamic non-linear system identification: RMSE results

Model	$\delta_A = 0.0$	$\delta_A = 0.5$	$\delta_A = 1.0$	$\delta_A = 1.5$
ANFIS	0.0011	0.2612	0.5052	0.782
FITSK	0.0522	0.2696	0.4235	0.8245
FIE-FITSK	0.0224	0.2017	0.2973	0.3932

$$f(t) = \begin{cases} \delta_A, & 1001 \leq t \leq 2000 \\ 0.0, & \text{otherwise} \end{cases} \quad (4.29)$$

where $\delta_A = [0.0, 0.5, 1.0, 1.5]$ representing four different levels of varying disturbances in the following test scenarios. $\delta_A = 0.0$ indicates no disturbances included and $\delta_A = 0.5$ to $\delta_A = 1.5$ indicates a gradual drift to a drastic shift in the data distribution.

The training data in this experiment uses $\delta_A = 0.0$ which does not include any disturbance, and tested with all four different level of disturbances. This experiment is to demonstrate that even with a fixed rule base, the proposed model will be able to handle situations when inputs are beyond the trained range. The number of membership functions at each dimension is defined as 5 forming a total of 25 rules, and a 100 training cycle was performed for all the models.

Figure 4.4: Dynamic non-linear system prediction results of FITSK and FIE-FITSK.

From Table 4.3, ANFIS was able to model with the best result as compared to FITSK and FIE-FITSK models when $\delta_A = 0.0$. This testing set is basically the same data used for training, and therefore ANFIS performed with a better accuracy. However, when δ_A starts to increase, the accuracy of all three models degraded as indicated by the increasing RMSE. From the result, it can be observed that FIE-FITSK had achieved much lower RMSE as the δ_A increases.

In Figure 4.4, it shows the plot of predicted results from both FITSK and FIE-FITSK models against the actual results. ANFIS results were similar to FITSK; hence, to improve the clarity of the plots, it was not included in the figures. As δ_A is added between time $1001 \leq t \leq 2000$, it can be observed in the figure, the errors were mainly accumulated during this period. A zoom in between time $901 \leq t \leq 2100$ of the plot with $\delta_A = 1.5$ that shows the greatest disturbance is shown in Figure 4.5. From the figure it can be clearly seen that the result from FITSK does not produce results that follows the trend of the actual result. On the other hand, FIE-FITSK produced a relatively more accurate result but gaps can still be observed.

Figure 4.5: Zoom in of Dynamic non-linear system $\delta_A = 1.5$ prediction results of FITSK and FIE-FITSK.

From the experimental results using different levels of disturbance, better predicted

output from the FIE-FITSK model was achieved as compared to both FITSK and ANFIS models where the data presented were beyond the data used to train the offline models.

The next section will introduce diabetes as a whole, followed by the modelling of a diabetic patient using GlucoSim [41], the glucose-insulin modelling capability of the proposed model and lastly with the insulin scheduler application.

4.3.3 Application of Diabetes Modelling

In this section, the diabetes problem that is plaguing the developed nations will be discussed and an insulin scheduler model is proposed to address the regulation issue in meal announcement for diabetic patients. The fuzzy interpolation and extrapolation FITSK (FIE-FITSK) model has been proposed to serve as the scheduler model. The FITSK model was also used to show the effect without the proposed fuzzy interpolation and extrapolation technique.

Global issue

As reported in [79], human life expectancy has increased significantly over the years. However, there is an increasing trend in noncommunicable diseases (NCDs) [4], with an estimated number of 38 million people dying each year because of NCDs. There are four major categories of NCDs; namely: cardiovascular diseases (including heart diseases and stroke), cancer, chronic respiratory diseases (including chronic obstructive pulmonary disease and asthma), and diabetes. NCDs are of long duration with slow progression, imposing a heavy financial burden on the patient, family, and society. The financial burden of NCDs is already recognised as being prominent in high income or developed countries, however, in recent years this burden has been associated with middle and lower income nations, especially in terms of the cost of medicines and the loss of income due to the inability to work [59].

In 2010, it was noted that in many developing nations NCD cases were generally discovered late, leading to extensive and expensive medical care. These catastrophic expenditures amount to billions of dollars in losses to the national income, and push

millions of people below the poverty line [3]. Death directly caused by diabetes was the lowest among the four major categories of NCDs globally (1.3 million deaths) [24]. However, diabetes is a major risk factor for cardiovascular disease, kidney disease, and blindness which causes co-occurrence of the other major categories of NCDs. An estimated expenditure of USD \$465 billion was spent world-wide solely on the treatment of diabetes in the year 2011. This amount to 11 per cent of the total health care expenditure for adults [6].

Introduction to diabetes

There are two main types of diabetes; namely: Type-1 diabetes (known as insulin-dependent-diabetes-mellitus), and Type-2 diabetes (known as non-insulin-dependent-diabetes-mellitus). Type-1 diabetes is diagnosed when the pancreatic beta cells are unable to secrete insulin, while Type-2 diabetes occurs when the pancreas is able to produce insulin, but at levels which are inadequate in regulating the blood glucose level efficiently, mainly due to prior metabolic deficiency as a result of insulin resistance. Both types of diabetes can lead to hypoglycemia (low blood glucose) and hyperglycemia (high blood glucose) in the short term, and acute chronic complications in the long term.

Hypoglycemia may happen from exogenous or endogenous insulin excess alone. It is a state where blood glucose level falls below a 4mmol/L or 72.07mg/dl. Some signs of hypoglycemia includes pallor and diaphoresis (often prominent), and increased heart rate and systolic blood pressure (often more subtle). The symptoms that a diabetic patient may feel includes sensations of warmth, weakness, fatigue, thinking difficulty, confusion, behavioral changes, and emotional lability. They also include seizures, loss of consciousness, and in severe cases, brain damage or even death [37]. Hyperglycemia may happen when the absence of insulin or insulin immunity produces an uncontrolled surge in glucose absorbtion into the blood stream. Hyperglycemia is considered when an individual experience high blood glucose level of more than 10mmol/L or 180mg/dl for a duration longer than 2 hours after a meal. The chronic hyperglycemia of diabetes is associated with long-term damage, dysfunction, and fail-

ure of various organs, especially the eyes, kidneys, nerves, heart, and blood vessels. The long-term complications include potential loss of vision; peripheral neuropathy with risk of foot ulcer, amputations, and Charcot joints; and autonomic neuropathy causing gastrointestinal, genitourinary, cardiovascular symptoms, and sexual dysfunction [12].

Current medical advancements have yet advanced to the point where a malfunctioning pancreas can be recovered. Hence the practice of administering exogenous insulin as a replacement of endogenous insulin has been in place since the 19th century. The need of exogenous insulin treatment has resulted in research into replacing the malfunctioned pancreas with an external artificial pancreas, also known as closed-looped blood glucose regulatory system. The current means of delivering insulin into the body is still commonly achieved by the diabetic patient, or family members, manually. The amount of insulin is determined by the pre-meal blood glucose measurement, obtained from a blood glucose sensor, and an estimate of the meal intake. Insulin over-dose can result in hypoglycemia while insulin under-dose can result in hyperglycemia with both conditions leading to death or health impairment.

As a result, there have been numerous attempts to develop an automatic closed-loop blood glucose regulatory system for the treatment of diabetes via a programmable insulin pump [19, 68, 92]. This is because continuous insulin infusion is found to be superior to multiple daily insulin injections for the treatment of diabetes [25, 70]. In addition, the trend is towards the development of an implantable insulin pump to facilitate convenience and to improve the quality of life of the diabetic patients [73]. Such a closed-loop setup consists of an integrated glucose sensor-controller-pump system that can ideally infuse the appropriate amount of insulin required by a diabetic patient based on his varying dietary intakes.

A successful closed-loop system has a number of components, including sensors, actuators, and controller. However, a dedicated user is still an important factor in closing the loop of the entire control system. The presence of a “humans in the loop” approach increases the risk of human error. Some promising studies involving simulated patients and actual type 1 diabetic patients, using simulated and actual

closed-loop controllers, have been carried out in [14, 47], respectively. However, in the implementation proposed by Pau Herrero *et al.* [47], multiple controller parameters need to be manually tuned before fitting it to an individual simulated patient. In the implementation proposed by Eran Atlas *et al.* [14], a predefined fuzzy logic rule base was established by the doctors. This tuning and rule base creation still involves significant human intervention with the associated possibility of human error.

The thesis uses a insulin scheduler simulation model that has been implemented using MATLAB Simulink [81] using the GlucoSim model [41]. It employs a method of non meal announcement to regulate the insulin infusion rate for the diabetic person. The proposed insulin scheduler model allows a greater degree of freedom to the diabetic person without meal announcement. Meal announcement would require the manual keying of the amount of insulin to be infused with respect to the estimated amount of food intake, or requiring an expert to either determine the setting of the parameters to fit an individual, or predefine a fuzzy logic rule base.

The proposed model omits these requirements by replacing it with readings of the trend in the blood glucose concentration level detected by a continuous glucose monitor. A range of subcutaneous glucose sensors (SGS) can be found in [1]. The trend of blood glucose concentration depends highly on the amount of food ingested and is delayed due to the time taken for digestion to occur before appearing as blood glucose. The proposed insulin scheduler hence can allow the diabetic person to freely consume any reasonable amount of food without the concern of over or under dosage of insulin.

The GlucoSim model

GlucoSim is an educational simulation package for glucose-insulin levels in the human body [41], based on [120]. There are two models of GlucoSim; namely: Healthy person, and type-1 diabetic patient. These two models have a similar glucose pharmacokinetic as shown in Figure 4.6. Glucose pharmacokinetic is the study of how the body will process the glucose from introduction to elimination.

The insulin pharmacokinetic for the healthy person model and type-1 diabetic

Figure 4.7: Pharmacokinetic diagram of healthy person insulin model.

depending on individual function. The detailed mathematical formulas used in GlucoSim can be found in [120] and will not be discussed in this thesis.

In the simulated type-1 diabetic patient model, the subcutaneous tissue compartment requires the insulin infusion rate as an external input. A trained scheduler can provide the insulin infusion rate according to the glucose reading attained.

Glucose-insulin modelling

The diabetic data used in this experiment is generated using GlucoSim [41]. It is able to generate both healthy and type 1 diabetic person blood glucose and insulin residual readings based on the meal announcements provided. Four different groups of dietary habits of the healthy person model are generated using GlucoSim. The four groups of dietary habits are normal dietary habit (Group A), under-eating dietary habit (Group B), over-eating dietary habit (Group C), and irregular dietary habit (Group D), as

Figure 4.8: Pharmacokinetic diagram of type-1 diabetic patient insulin model.

illustrated in Figure 4.9.

Group A mimics the behaviour of a person sticking strictly to the recommended food intake regime. Group B and Group C mimics a person eating at the recommended time, but with a reduced or increased amount of the recommended food quantity, respectively. Group D mimics the person having food intake at any random time and of any amount without following any medical recommendation. This behaviour is shown in Figure 4.9 whereby each blood glucose rise is in response to a food intake regime.

Table 4.5 shows the upper and lower blood glucose limits found in each of the training sets. This limit is used as a predefined limit specified in the CMAC used in both FITSK and FIE-FITSK models during creation, and is necessary because of

Table 4.4: Notations used in the GlucoSim model.

Variables:	
G	Glucose concentration (mg/dl)
I	Insulin concentration (microU/dl)
r	Metabolic source or sink rate (mg/min)
Subscripts: Physiological Compartments	
B	Blood
K	Kidney
NS	Nervous System
PR	Periphery
PN	Pancreas and Spleen
PV	Portal Vein
GT	Gastrointestinal Tract
H	Heart
L	Liver
SC	Subcutaneous Tissue
Subscripts: Metabolic Rates	
KGU	Kidney Glucose Utilization
NSGU	Nervous System Glucose Utilization
HGU	Heart Glucose Utilization
LGU	Liver Glucose Utilization
PRGU	Periphery Glucose Utilization
LGP	Liver Glucose Production
GA	Glucose Absorbtion
KIR	Kidney Insulin Removal
PRIR	Periphery Insulin Removal
LIR	Liver Insulin Removal
IA	Insulin Absorbtion
PIE	Pancreas Insulin Excretion

the CMAC structure as mentioned in Section 4.1.1. The lower limits of the different dietary habits are similar while the upper limits of each dietary habit varies greatly. The structure of input and output data in this experiment is $X = [G(t - 15), G(t - 10), G(t - 5), G(t)]$ and $Y = IR(t + 5)$, respectively. $G(\cdot)$ represents the blood glucose reading at current time t and the historical reading in time intervals of 5 minutes, and $IG(t + 5)$ is the predicted insulin residual within the blood 5 minutes after time t .

Twenty days of data for each individual dietary habit group have been generated using GlucoSim, with the first day of the data being discarded. This is because the

Figure 4.9: Blood glucose fluctuation from the four groups of different dietary habits.

Table 4.5: Blood glucose maximum and minimum limit of different dietary habit trained models and the respective standard deviation of each dietary habit data.

Dietary Habit Group	Minimum (mg/dl)	Maximum (mg/dl)	Std Deviation
Group A (Normal)	26.08	294.72	42.65
Group B (Under-eating)	26.44	211.52	31.71
Group C (Over-eating)	26.11	348.93	54.41
Group D (Irregular)	26.87	568.26	63.71

initial setting of the GlucoSim simulation will cause slight differences which may result in inaccuracies if used. The data is divided into training and testing sets. The first ten days of the data is used as the training set while the remaining nine days of data will be the testing set used to validate the trained model. The testing sets will be labelled as Group A* (Normal dietary habit), Group B* (Under-eating dietary habit), Group C* (Over-eating dietary habit), and Group D* (Irregular dietary habit).

The objective of this experiment is to demonstrate the modelling ability of the trained FIE-FITSK using a particular dietary habit group to model all four different dietary habit groups. Four FIE-FITSK models will be trained using the training sets

of Group A, B, C and D, and each trained model will then be tested with the testing sets of all the four different dietary habit groups, namely: Group A*, Group B*, Group C*, and Group D*. As shown in Table 4.5, when a model is trained using Group A or Group B, the training data does not cover the upper limit of the data coverage observed in Group C and Group D. Therefore, data drift or shift is present in the Group C* and Group D* testing sets. Three nearest rules are used in the interpolation and extrapolation inference process. The result of the experiments are shown in Figures 4.10, 4.11, 4.12, and 4.13.

The comparisons between FITSK and FIE-FITSK are shown in Table 4.6 and Table 4.7, in terms of RMSE and correlation (R), respectively. Correlation (R) is derived using Equation (4.30) which shows how strongly linked between the predicted outputs and the desired outputs.

$$R = \frac{\sum_{n=1}^N (o_n^d - \bar{o}^d)(o_n - \bar{o})}{\sqrt{\sum_{n=1}^N (o_n^d - \bar{o}^d)^2} \sqrt{\sum_{n=1}^N (o_n - \bar{o})^2}} \quad (4.30)$$

where:

o_n^d is the desired output;

o_n is the predicted output from the network;

$\bar{o}^d$ is the mean of the desired outputs;

$\bar{o}$ is the mean of the predicted outputs; and

N is the total sample size.

From the RMSE comparison, the results show that on average, FIE-FITSK performed better than the FITSK model for Group A, Group B, and Group C trained data, as supported by the average correlation results shown in Table 4.7.

The improvement in modelling accuracy that FIE-FITSK exhibits over FITSK was shown by the results highlighted for the test cases using the Group C* and Group D* test sets. As shown in Figure 4.14, the inability to produce any output had caused the low correlation and high RMSE of the FITSK trained models. Higher correlation values were recorded by FIE-FITSK in the experiments where FITSK performed poorly as highlighted in Table 4.7. Higher accuracy in terms of lower RMSE by FIE-FITSK trained models can also be observed, as highlighted in Table 4.6.

Table 4.6: Overall comparison of RMSE between FITSK and FIE-FITSK in diabetes experiment.

	Testing Dietary									
	Group A*		Group B*		Group C*		Group D*		Average	
	FITSK	FIE-FITSK	FITSK	FIE-FITSK	FITSK	FIE-FITSK	FITSK	FIE-FITSK	FITSK	FIE-FITSK
Training Dietary										
Group A	0.9903	2.0896	0.7700	1.6409	5.4015	8.3973	45.6876	36.8369	13.2123	12.2412
Group B	7.2259	11.2379	1.1399	2.9856	37.6908	33.9964	66.6924	57.7807	28.1872	26.5001
Group C	1.5110	2.4366	1.3415	2.0198	1.8283	2.9909	35.2225	23.7681	9.9758	7.8039
Group D	1.7460	3.4623	1.6576	3.2510	2.0183	3.9790	2.4772	5.1958	1.9748	3.9720

Table 4.7: Overall comparison of Correlation (R) between FITSK and FIE-FITSK in diabetes experiment.

	Testing Dietary									
	Group A*		Group B*		Group C*		Group D*		Average	
	FITSK	FIE-FITSK	FITSK	FIE-FITSK	FITSK	FIE-FITSK	FITSK	FIE-FITSK	FITSK	FIE-FITSK
Training Dietary										
Group A	0.9997	0.9989	0.9997	0.9986	0.9962	0.9908	0.7900	0.8808	0.9464	0.9673
Group B	0.9882	0.9753	0.9993	0.9958	0.7909	0.8571	0.4833	0.6804	0.8154	0.8771
Group C	0.9994	0.9985	0.9991	0.9980	0.9995	0.9988	0.8803	0.9524	0.9696	0.9869
Group D	0.9992	0.9972	0.9987	0.9950	0.9994	0.9978	0.9994	0.9976	0.9992	0.9969

Figure 4.10: Group A trained network result plots.

Figure 4.11: Group B trained network result plots.

Figure 4.12: Group C trained network result plots.

Figure 4.13: Group D trained network result plots.

The higher correlation from the FIE-FITSK trained models are observed due to the ability to interpolate or extrapolate to cover the missing regions thus partially filling in the gaps, as shown in the figure. This is because fuzzy interpolation and extrapolation allows the trained model to handle and extrapolate a result beyond the boundaries. When compared to the non-highlighted results of Table 4.6, the RMSE from FIE-FITSK is higher than the RMSE from FITSK. FIE-FITSK uses an approximation method to derive a result instead of the precise computation of FITSK, as discussed in Chapter 3.

Figure 4.14: Zoom in on all Group A trained Group D* tested, Group B trained Group D* tested, and Group C trained Group D* tested.

Figure 4.15 shows two example of an evenly distributed single antecedent FCMAC rule base consisting of four fuzzy rules. Figure 4.15(a) is a simplified example of the experiment to illustrate the non-highlighted situations of Table 4.6 explaining the underlying reason why better accuracy is achieved by FITSK than FIE-FITSK. As

mentioned in the computation of FITSK in Section 4.1, τ is a threshold value defined in determining whether a rule is triggered to be fired. If a fuzzy subethood is above τ , the rule will be fired. In the figure, two levels of τ are shown. With the input x_1 and when τ_2 is used, rules R_1 and R_2 will be fired, and when τ_1 is used, only rule R_1 will be fired. This ensured that each fired rule is of a significant relevance to the inputs. However, FIE-FITSK fires all n -nearest rules with reference to the input regardless of the rule relevancy. In the figure, $n=3$, the 3 nearest rules are R_1 , R_2 , and R_3 at the distance of d_1 , d_2 , and d_3 , respectively. Thus the results from the n -nearest rules may not be precise.

Figure 4.15(b) is a simplified example of the experiment to illustrate the highlighted situations of Table 4.6. As seen in the figure, the input x_1 falls on the extreme right. When using FITSK technique, no rule will be fired as the only intersection of the input and the extreme right rule, R_4 , has the fuzzy subethood below the threshold τ . However, FIE-FITSK is able to deduce a plausible result from the n -nearest available rules.

Figure 4.15: Rules involved in inferencing by FITSK and FIE-FITSK techniques.

Insulin Scheduler Model

In 2007, Phee et. al. proposed an insulin regulatory system to regulate the amount of insulin administered to a diabetic patient [94] without any knowledge of the amount of carbohydrates going to be consumed. A pseudo-outer product based fuzzy neural network using the Yager rule of inference (POP-Yager) was used to predict the insulin level after a certain time step based on the glucose reading from the simulated patient model. However, this work focuses only on the ideal situation whereby regular and well controlled meals based on dietician's recommendation were consumed.

In 2009, a similar model to regulate the amount of insulin to be administered was proposed by Ting and Quek [113]. This work used two independent zero-ordered Takagi-Sugeno-Kang fuzzy CMAC (TSK0-FCMAC) models to act as a healthy reference model and to predict the insulin infusion rate used by the insulin pump. The healthy reference TSK0-FCMAC model was trained using the irregular dietary habit data to model all the different types of dietary habits of a simulated diabetic patient. However, to use irregular dietary habit data to train a model is impractical. The collection of real diabetic patient with irregular dietary habit poses high risk of causing irreversible effects on the health of the diabetic patient. Therefore the collection of such data in real life is impossible.

Teddy et. al. proposed a glucose-insulin regulation control model where a pseudo self-evolving CMAC (PSECMAC) model was used to predict the insulin requirement of the diabetic patient [112]. A PSECMAC model was trained using the normal dietary habit and tested against all four different dietary habits. The focus of comparison in this work was against a basic CMAC network because this is the predecessor of the proposed PSECMAC and CMAC network is also widely used in control applications. Results of this work had shown that PSECMAC performed better in both normal and undereat dietary habit, but perform poorer in both overeat and irregular dietary habit as compared to CMAC network. The author of the work had analysed that due to the non-uniformly distributed memory cells allocated during the training phase using the normal dietary habit, more memory cells were allocated to the significant regions of the inputs of the training data. The uniformly distributed CMAC

structure although perform poorer than that of PSECMAC in the normal and under-eat dietary habit, but the averaging effect allows more robust capability in handling overeat and irregular dietary habit.

In this chapter, the underlying learning structure of both FITSK and FIE-FITSK uses a uniform cell distribution CMAC structure. This is to avoid assumption of data distribution based solely on the training data. In the previous section of 4.3.3, the FITSK and FIE-FITSK models were trained using different dietary groups to determine the best model for the scheduler in the proposed insulin scheduler. From the results as shown in Table 4.6 and Table 4.7, the selected schedulers are trained using data based on the Irregular dietary habit, Group D. This result coincides with the work in[113] where their work also used the irregular dietary habit to train their healthy reference model.

This simulation is done using GlucoSim’s healthy reference and diabetic models, which model a healthy person with a functioning pancreas, and a diabetic person without any pancreas activation, respectively, as mentioned in Section 4.3.3. A two day simulation with the food intake schedule shown in Table 4.8 is performed. Both models are based on a 60kg body weight, with an initial blood glucose concentration level of 85mg/dl and residual insulin at 1500 microU/ml.

Table 4.8: Two days meal schedule.

Day	Meal Timing	Carbohydrates(g)
One	1300hrs	200
	1530hrs	350
	1800hrs	100
	2030hr	75
	2300hr	20
Two	0900hrs	210
	1000hrs	30
	1200hrs	300
	1500hrs	100
	1900hrs	120
	2200hrs	20

From Table 4.8, it can be observed that the GlucoSim patient/person consumed a different amount of food at different timings over the two days. This setting aims to show that with variable meal intake, in terms of time and amount, by the diabetic

patient, the scheduler is able to maintain and control the blood glucose concentration level within the acceptable range to prevent hyperglycemic or hypoglycemic conditions from occurring. This setting also shows that the diabetic patient does not need to adhere to strict meal consumption timing or the amount of food to be consumed. This minimizes the disruption to the normal daily lifestyle of the diabetic patient.

The following section will discuss the results generated from the GlucoSim simulation by using the two proposed fuzzy neural networks implemented as the schedule of a insulin scheduler model.

Insulin Scheduler Results

In this section, the FIE-FITSK model proposed in Section 4.3.3 will be used as a insulin scheduler to predict the insulin residual that will be converted into the rate of insulin infusion into the subcutaneous tissue to be announced to the pump, represented as the subcutaneous tissue shown in Figure 4.8. The FITSK model will be compared to so as to show the effect of having the fuzzy interpolation and extrapolation technique of the proposed model.

From Table 4.6 and Table 4.7, the irregular dietary habit trained model (Group D) achieved the lowest average RMSE and highest average correlation across all four different dietary habits. This shows that with a Group D trained model, it is able to model different types of dietary habits. Therefore in this application, both FITSK and FIE-FITSK will be trained using the Group D dietary habits so that the trained model will be able to handle the unknown dietary habit of the simulated diabetic patient.

The main mechanical components involved are the subcutaneous glucose sensor (SGS), scheduler, and insulin pump. A SGS is now commonly used to replace the traditional finger prick methods to measure the blood glucose level in the body. With an SGS, the blood glucose level can be sampled in intervals of minutes rather than the much coarser resolution determined by the frequency of the finger prick test of the traditional method. Some of the common features of a SGS is listed as follow.

- System includes a receiver, transmitter, and sensor.

- Waterproof transmitter and sensor.
- Requires periodic manual calibration.
- Mechanical device to insert electrodes.
- Warm up period required at every start of sensor session for initial calibration.
- Minimal skin irritation at site of electrodes insertion.

A comparison of the technical aspects comparisons have been made in [2] in terms of size, weight, sensor life, cannula size, etc. The performance of different SGS has been examined in [42]. The selection of SGS is not within the scope of this thesis.

The insulin scheduler system has been implemented in MATLAB Simulink as shown in Figure 4.16. Block A shows the insulin scheduler model, which will implement either the FITSK or FIE-FITSK models. Block B (insulin infusion model) computes the rate of insulin infusion that was proposed in [111]. This insulin infusion model reads in the predicted insulin residual value at time $t + 5$ from Block A to determine the rate of insulin infusion by the insulin pump. Block C is the GlucoSim type-1 diabetic model that is able to provide a blood glucose reading at per minute intervals.

Two objectives for the system were set out, namely:

1. To mimic the blood glucose fluctuation, the motion of rise and fall, as compared to the healthy GlucoSim reference model.
2. To maintain the blood glucose level within the healthy range.

For a fair comparison between the models' behaviour, the healthy reference model in this experiment shares the same metabolic behaviour as the GlucoSim type-1 diabetic model. Hypoglycaemia is indicated when blood glucose level falls below 4mmol/L or 72.07mg/dl, and hyperglycaemia is indicated when blood glucose level stays above 10mmol/L or 180mg/dl for a duration longer than 2 hours after a meal.

The simulation for two days, based on the meals in Table 4.8, was conducted. Figure 4.17 shows the simulated results from both trained models, FITSK and FIE-FITSK, against the healthy reference model with the same meal schedule.

The results from both schedulers used with a simulated diabetic person are shown in Figure 4.17. Both schedulers were trained using the Irregular dietary habit data

Figure 4.16: Insulin scheduler system.

Figure 4.17: Result of simulation with Irregular dietary habit trained models.

(Group D) as discussed in Section 4.3.3. Deviation between the simulated diabetic person and the healthy reference can be observed. The healthy reference model response immediately when there is food intake, however in the diabetic model, insulin will only be released upon a series of blood glucose rise as predicted by the schedulers. A drift in the resulting blood glucose reading was observed as compared to the healthy reference model from both diabetic models. From the figure, it could be observed that hyperglycaemia did not occur as there was no indication of prolonged hyperglycaemic condition for more than two consecutive hours.

Hypoglycaemia could be observed from the results generated from both scheduler models and also the healthy reference model. This issue lies with the modelling ability of GlucoSim, which requires mathematical modifications of the GlucoSim model in

Figure 4.18: Result of simulation with Normal dietary (Group A) trained models.

[41]. Therefore, the hypoglycaemia issue could not be effectively addressed.

There were three indicated regions where there was an increase in blood glucose level of the healthy model without any tagged meal intake (shown in carbohydrates along the time axis). This was likely to be release of glucose from the liver component, as shown in Figure 4.6. The two glucose release at around 4am of both days were likely to be an inhibited mechanism to prevent the hypoglycaemia situation. The additional increase in blood glucose level was not observed in both diabetic models; hence it was not modelled.

The objective of comparing the results from the models with the healthy reference model was to show that the models were able to have the similar behaviour of blood glucose fluctuation. This tells us that the insulin scheduler was able to control the amount of necessary release at each time point so as to control the patient to stay out of the hyper and hypoglycaemic conditions. However, it will be ideal that a model will be able to mimic the healthy model's behaviour.

With GlucoSim, data from different dietary groups can be easily generated. However, in reality, this type of data collected from humans is very limited. The risk of collecting variable dietary habit data from a diabetic person is high. Any mistakes

made in the amount of insulin to be administered may cause adverse immediate or long term effects as discussed in Section 4.3.3. Therefore the safest set of data that can be collected is the normal dietary habit which does not put high risks to the diabetic person. Hence, the Normal dietary habit (Group A) trained FITSK and FIE-FITSK models will be used as the scheduler in the following experiment.

Figure 4.19: Result of FIE-FITSK trained models.

Figure 4.18 shows the results of the control system using the Normal dietary (Group A) trained FITSK and FIE-FITSK models as the scheduler. Same meal announcement as the former experiment was used. Alignment between the healthy reference model and predicted output from FITSK was poorer as compared to the FIE-FITSK. Large gaps in terms of the peaks and troughs between the healthy reference model and the diabetic model using the FITSK trained scheduler were observed. The result from the FIE-FITSK model had shown higher similarity with the healthy reference model. Although the FITSK model stays within the healthy blood glucose level, however, the hypoglycaemia condition was more consistently observed during the night of both days from the FITSK model as compared to the FIE-FITSK model. This is due to the prediction error causing an increase in the dosage of insulin to be released hence causing the hypoglycaemic condition to occur. This is a more critical condition to avoid for a diabetic patient as adverse immediate effects commonly occurs

during hypoglycaemic situation.

The standard deviation of the glucose reading obtained from the healthy reference model was measured at 68.11 which is similar to the Irregular dietary habit training data as shown in Table 4.5. This signifies a high irregularity of the dietary habit being modelled. The inability of the conventional offline models such as FITSK does not efficiently handle situations when new data are found beyond the limits from the training data. Producing zero-output from the inference affects directly the amount of insulin to be released into the diabetic patient. This may cause health risk to the patient which may be irreversible. However, with the fuzzy interpolation and extrapolation technique, FIE-FITSK was able to produce better result when the model was trained with data that does not cover unseen data.

To further discuss, in medical context, keeping the person within the healthy range is of top priority. In this experiment, we are assuming that the GlucoSim is an accurate simulator in simulating the blood glucose of a person with the respective carbohydrate consumption. Therefore the FIE-FITSK results shown in Figure 4.18 is considered as a better match as compared to the FITSK model in terms of mimicing the blood fluctuation behaviour.

The two objectives in this application were to mimic the blood glucose fluctuation of the healthy reference model and also maintaining the blood glucose level within the healthy range. The blood glucose fluctuation as shown by both dietary habit trained FIE-FITSK models in Figure 4.19 shows the well controlled rise and fall of the blood glucose. A complete matching of the diabetic model and the healthy reference model is near impossible as the amount of insulin released from an insulin pump will never be able to be similar to an actual pancreas behaviour.

4.4 Summary

This chapter had proposed a fuzzy neural network, named as FIE-FITSK, extended from the CMAC structured FITSK. FIE-FITSK was implemented to allow more flexibility to infer more reasonable results from the restrictions of the boundaries of the rigid CMAC structure. The first half of the experiment discussed in Section 4.3 showed

that FIE-FITSK could produce similar accuracy as compared to existing models for both Mackey-Glass and non-linear system identification time series problems. The second half of the experiment, where the testing data distribution differs from the training data, the proposed FIE-FITSK was able to produce better results as compared to other offline models. The fuzzy interpolation and extrapolation inferencing technique enables a bounded offline trained model to have the flexibility to produce beyond its bound.

The FITSK and FIE-FITSK trained diabetic models from Section 4.3.3 were used as the scheduler for managing the blood glucose level of a diabetic person. From Table 4.6 and Table 4.7, the irregular dietary habit trained model (Group D) was the best option to be used as the scheduler that would be able to model all four different dietary habits. The results in Figure 4.17 had shown that both trained schedulers were able to fairly control the underlying blood glucose fluctuation by the amount of insulin to be released from the insulin pump. However, high risk would be involved to gather the actual irregular dietary habit data from a diabetic patient. Therefore, the normal dietary habit data would be the only possible data that could be collected from a real diabetic patient.

In Figure 4.18, the results of the models trained with normal dietary habit (Group A) were shown. The FITSK model was unable to produce the behaviour of blood glucose fluctuation as shown by the healthy reference model. On the other hand, FIE-FITSK model trained with Group A data was able to produce similar blood glucose fluctuation.

The modelling result of FIE-FITSK model using both Normal and Irregular dietary habit trained scheduler can be seen in Figure 4.19. Both results have fairly similar behaviour, therefore indicating that by using a Normal dietary habit data that can be obtained safely from an actual diabetic patient, an insulin scheduler can be sufficiently trained.

However, the fuzzy neural network presented are based on a batch learning structure. Batch-learning is unable to model data with a changing distribution. Therefore in Chapter 5, an evolving fuzzy neural network will be introduced, to handle data

where the distribution is continuously changing.

Chapter 5

Evolving Fuzzy Neural Networks

In Chapter 4, FIE-FITSK was proposed which uses the CMAC learning structure. This learning structure is able to generalize with localized learning due to its local weight adjustment that exhibits fast learning speed. However, FIE-FITSK was not incorporated with online learning capability, therefore the rules learnt will not be able to adaptively change when data trend changes. As the data trend changes drastically from the data used for training, practically the offline trained FIE-FITSK will not be effective. The stock trading data has the ever changing data trend, which does not have any distribution bound or consistent pricing fluctuation pattern. Therefore, to cope with these fluctuations, we propose an evolving FNN.

An evolving FNN adopts an incremental learning structure where knowledge creation is performed at each time instance and outdated rules will be pruned. This allows the fuzzy rules to be created based on the current data distribution and thus maintain the rule base recency. The generic self-evolving Takagi-Sugeno-Kang (GSETSK) is an evolving FNN that has incorporated learning and unlearning capabilities [89]. It uses a clustering technique known as multidimensional-scaling growing clustering (MSGC) for structural learning and recursive least square (RLS) for consequent parameter learning. GSETSK possesses rule learning and rule pruning capabilities which enables the knowledge base to be kept current. This allows the modelling of the concept drift situation, as discussed in Chapter 1. However, its effectiveness in modelling concept shift is not obvious.

The fuzzy interpolation and extrapolation (FIE) inference method may be able

to handle the concept shift situation. FIE is capable of extrapolating fuzzy inference results from the existing rule base where knowledge beyond the current rule base is unknown. This motivated the implementation of FIE with a rule pruning evolving FNN in order to handle both concept shift and concept drift situations simultaneously.

In this chapter a new FNN known as the evolving fuzzy interpolation and extrapolation Takagi-Sugeno-Kang fuzzy model (eFIE-TSK) is proposed. The proposed FNN uses a similar structure, known as generic self-evolving Takagi-Sugeno-Kang (GSETSK) model proposed in [88, 89]. The details of the GSETSK model is included in the following section for completeness. The sections that follows will provide the details of the proposed eFIE-TSK model and comparison results.

5.1 Generic Self-Evolving Takagi-Sugeno-Kang (GSETSK) Model

In this section, the structure and learning mechanism of GSETSK [89] is reviewed. GSETSK starts with an empty rule base, and new rules are sequentially discovered by MSGC, which is completely data-driven. The GSETSK model is basically an FNN that consists of six layers of computing nodes, as shown in Figure 5.1. They are: Layer 1 (the input layer), Layer 2 (the input linguistic layer), Layer 3 (the rule layer), Layer 4 (the normalization layer), Layer 5 (the consequent layer) and Layer 6 (the output layer). The structure of the GSETSK model defines a set of TSK-type IF-THEN fuzzy rules. The fuzzy rules are incrementally constructed by presenting the training observations $\dots (X(t), d(t)) \dots$ sequentially, where $X(t)$ and $d(t)$ denote the vectors containing the inputs and the corresponding desired outputs, respectively, at any time t . Each fuzzy rule R_k in GSETSK has the form shown in Equation (5.1).

$$\begin{aligned}
 R_k : & \text{ IF } (x_1 \text{ is } IL_{1,j_1^k}) \text{ AND } \dots (x_i \text{ is } IL_{i,j_i^k}) \text{ AND } \dots (x_n \text{ is } IL_{n,j_n^k}) \\
 & \text{ THEN } y_k = b_{0k} + b_{1k}x_1 + \dots + b_{ik}x_i + \dots + b_{nk}x_n \quad (5.1)
 \end{aligned}$$

where:

$X = [x_1, \dots, x_i, \dots, x_n]^T$ represents the numeric inputs of GSETSK;

$IL_{i,j_i^k}(j_i = 1, \dots, J_i(t), k = 1, \dots, K(t))$ denotes the j_i th linguistic label of the input x_i that is part of the antecedent of rule R_k ;

$J_i(t)$ is the number of fuzzy sets of x_i ;

$K(t)$ is the number of fuzzy rules at time t ;

y_k is the crisp output of rule R_k ;

n is the number of inputs; and

$[b_{0k}, \dots, b_{nk}]$ represents a set of consequent parameters of rule R_k .

Figure 5.1: Structure of GSETSK network, adapted from [89].

Each input node I_i may connect to a different number of input linguistic nodes $J_i(t)$. Hence the total number of nodes in layer 2 at each time t is $\sum_{i=1}^n J_i(t)$. Also, at each time t , layer 3 consists of $K(t)$ rule nodes R_k . It should be noted that $K(t)$ and $J_i(t)$ change over time, increasing to accommodate new data or decreasing to maintain a compact fuzzy rule base. Each rule node R_k is directly connected to a normalization node N_k in layer 4. Subsequently, each normalization node N_k is directly connected to a consequent node C_k in layer 5. Hence, the number of nodes in layer 3, layer 4

and layer 5 are the same. For clarity in subsequent discussion, the variables i and j are used to refer to arbitrary nodes in layers 1 and 2, and the variable k for nodes in layers 3, 4, and 5. The output node at layer 6 is a summation node which is connected to all the nodes in layer 5.

GSETSK performs two operations; namely: forward reasoning to derive a result, and backward computations of the rule potential. The rule potential is essentially used to determine whether that specific rule is out-dated and suitable for removal from the rule base.

Detailed mathematical functions of each layer of GSETSK are shown in the following section.

5.1.1 Forward Reasoning

Layer 1: Input Layer

$$Z_i^1 = x_i, \quad i = 1, \dots, n \quad (5.2)$$

where:

Z_i^1 represents the input node x_i ; and

n represents the number of inputs.

Layer 1 nodes are termed the linguistic nodes. Each node receives only one input as one dimension of the input vector, and outputs to several nodes in the next layer.

Layer 2: Input Linguistic Layer

$$Z_{i,j_i}^2 = \mu_{i,j}(Z_i^1) = \mu_{i,j_i}(x_i), \quad i = 1, \dots, n, \quad j_i = 1, \dots, J_i(t) \quad (5.3)$$

where μ_{i,j_i} is a fuzzy membership function of the fuzzy linguistic node IL_{i,j_i} .

Layer 2 nodes are known as the input-label nodes. They are the antecedent of the fuzzy rules. The label IL_{i,j_i} denotes the j th linguistic label of the i th linguistic input variable. The matching degree of the input (x_i) is measured with its corresponding linguistic nodes in this layer using a Gaussian membership function having its center and width dynamically computed during the structural learning phase. Equation (5.3)

can be expressed as Equation (5.4).

$$Z_{i,j_i}^2 = \mu_{i,j}(x_i) = e^{-\frac{(x_i - m_{i,j_i})^2}{\sigma_{i,j_i}^2}}, \quad i = 1, \dots, n, \quad j_i = 1, \dots, J_i(t) \quad (5.4)$$

where:

m_{i,j_i} the center of the Gaussian membership function of the j th linguistic label of the i th linguistic variable input x_i ; and

σ_{i,j_i} the width of the Gaussian membership function of the j th linguistic label of the i th linguistic variable input x_i .

Layer 3: Rule Layer

Each rule node in the rule-base layer represents a single Sugeno-type fuzzy rule. The firing strength of a rule node R_k is computed based on the activation of its antecedents as shown in Equation (5.5).

$$r_k = Z_k^3 = \min(Z_{1,j_1^k}^2, \dots, Z_{i,j_i^k}^2, \dots, Z_{n,j_n^k}^2), \quad k = 1, \dots, K(t) \quad (5.5)$$

where:

$Z_{i,j_i^k}^2$ is the output of the j th linguistic label of the i th linguistic variable input x_i that connects to the k th rule;

k representing the number of rules; and

r_k is the forward firing strength of R_k .

Layer 4: Normalization layer

$$w_k = Z_k^4 = \frac{Z_k^3}{\sum_{k=1}^{K(t)} Z_k^3}, \quad k = 1, \dots, K(t) \quad (5.6)$$

where w_k is the normalized firing strength.

Layer 5: Consequence Layer

$$Z_k^5 = Z_k^4 \cdot f_k(X), \quad k = 1, \dots, K(t) \quad (5.7)$$

where $f_k(X)$ is a linear function of consequent node C'_k .

Layer 6: Summation Layer

$$Z^6 = \sum_{k=1}^{K(t)} Z_k^5 \quad (5.8)$$

The output node in this layer corresponds to the output of the GSETSK model.

The GSETSK model employs a Hebbian-based rule pruning algorithm which takes into consideration the backward connections from layer 6 to layer 3 via layer 5. This is described in the following section.

5.1.2 Backward Computations of GSETSK

The backward connections from layer 6 to layer 3 via layer 5 in GSETSK serve the purpose of computing the potentials of the fuzzy rules in GSETSK. These fuzzy rule potentials are subsequently used to determine if the rules will be pruned or kept. The fuzzy rule potentials are based on the observed training data pair $(X(t), d(t))$. At each rule node R_k , the forward firing strength r_k is described in Equation (5.5); and the backward firing strength b_k^{back} is computed in the following steps.

1. Compute the Output Error of each Fuzzy Rule:

Layer 5 (Backward Operation): At time t , the desired output $d(t)$ is directly transmitted to each consequent node C_k in layer 5. The output of the linear function of the consequent node C_k in response to the input $X(t)$ is a crisp value given in Equation (5.9).

$$y_k(t) = b_{0k}(t) + b_{1k}(t)x_1(t) + \dots + b_{ik}(t)x_i(t) + \dots + b_{nk}(t)x_n(t), \quad k = 1, \dots, K(t) \quad (5.9)$$

where $[b_{0k}(t), \dots, b_{nk}(t)]$ represents a set of consequent parameters of rule R_k at time t . Note that y_k is the output of the fuzzy rule R_k . It is different from Z_k^V , which is the output of the consequent node C_k . For each rule R_k , the difference between the computed output $y_k(t)$ and the desired output $d(t)$ is computed in

Equation (5.10).

$$e_k(t) = |d(t) - y_k(t)|, \quad k = 1, \dots, K(t). \quad (5.10)$$

where $e_k(t)$ is the output error of rule R_k at time t .

2. Determine the Backward Firing Strength of each Fuzzy Rule:

Layer 5 (Backward Operation): The values $e_1(t), \dots, e_k(t), \dots, e_{K(t)}(t)$ will then be used to form a Gaussian membership function with the mean of 0 and the width at time t , see Equation (5.11).

$$\sigma_{back}(t) = \frac{\sum_{k=1}^{K(t)} e_k(t)}{K(t)\sqrt{\pi}} \quad (5.11)$$

This membership function measures how closely the computed output $y_k(t)$ can approximate the desired output $d(t)$. Let $\mu(0, \sigma_{back}(t))$ denote the Gaussian membership function with center 0 and width $\sigma_{back}(t)$. Figure 5.2 shows such a Gaussian membership function, which can be approximated by an isosceles triangle with unity height and the length of its bottom edge equal to $2\sigma_{back}(t)\sqrt{\pi}$.

Figure 5.2: The Gaussian membership function with mean=0 and standard deviation represented as $\sigma_{back}(t)$.

The backward firing strength of b_k^{back} of rule R_k at time t is then determined

using Equation (5.12).

$$b_k^{back}(t) = \mu(0, \sigma_{back}(t), e_k(t)) = \exp(-e_k(t)^2 / \sigma_{back}(t)^2) \quad (5.12)$$

At layer 5, the Gaussian function $\mu(0, \sigma_{back}(t))$ is formulated to measure the degree of closeness between the desired output $d(t)$ and the computed output $y_k(t)$. When $\mu(0, \sigma_k(t), e_k(t)) = 1$, $e_k(t) = 0$, and $y_k(t) = d(t)$. The smaller the value of $e_k(t)$, the greater the value of $\mu(0, \sigma_k(t), e_k(t))$. That also means the closer the computed output $y_k(t)$ is to the desired output $d(t)$, the greater the backward firing strength of rule R_k . It can be observed from Equation (5.11) that the width $\sigma_{back}(t)$ is constructed using the average of the errors of all the rules at time t . This approach was built on the idea that the existing fuzzy rules in GSETSK at time t should be compared against each other in terms of how well they were able to approximate the desired output.

The rule-pruning algorithm of GSETSK relies on the fuzzy rule potentials. The potential P_k of a fuzzy rule R_k in GSETSK indicates its importance in the entire rule base of the system. At any time t , the potential P_k of a fuzzy rule R_k can be recursively calculated based on the current training data $(X(t), d(t))$ as shown in Equation (5.13).

$$P_k(t) = P_k(t-1) + \overbrace{r_k(X(t))}^{\text{Forward}} \times \overbrace{b_k^{back}(d(t))}^{\text{Backward}} \quad k = 1, \dots, K(t) \quad (5.13)$$

where:

$P_k(t-1)$ as the potential of rule R_k at time $(t-1)$;

$r_k(X(t))$ is the forward firing strength of rule R_k as given in (5.5); and

$b_k^{back}(d(t))$ is the backward firing strength of rule R_k as given in (5.12).

Equation (5.13) shows that the importance of a fuzzy rule R_k in GSETSK is reinforced if its input antecedents and computed output can closely mimic the information expressed in the training pair $(X(t), d(t))$. This complies with the Hebbian learning mechanism behind the long-term potentiation phenomenon [122] in the brain. The mechanism is based on the Hebb theory which states that the synaptic connections

of the associative memories formed in the brain are strengthened when coincident pre-synaptic and post-synaptic activities occur.

A forgetting mechanism has been implemented in GSETSK to allow the pruning of rules in the rule base to maintain an up-to-date rule representation. This updated rule base consist of a set of up-to-date fuzzy rules that best describes the current characteristics of the incoming data. Furthermore, the rule base will be more compact and can be better interpreted by human experts. This is done by adding a forgetting factor to the original formulation described in Equation (5.13), defined as in Equation (5.14).

$$P_k(t) = \lambda P_k(t-1) + r_k(X(t)) \times r_k^{back}(d(t)) \quad \lambda \in (0, 1] \quad k = 1, \dots, K(t) \quad (5.14)$$

where λ is the forgetting factor.

The smaller λ is, the faster the effects of old rules decay. The rule R_k will be pruned if $P_k(t)$ falls below the predefined parameter $thresP$.

5.1.3 Parameter Learning of GSETSK

In this phase, only the consequent parameters in the consequent nodes at layer 5 will be tuned. In GSETSK, the output node at layer 6, based on the observed data pair (X, D) , is shown in Equation (5.15).

$$y = \sum_{k=1}^{K(t)} Z_k^5 = \sum_{k=1}^{K(t)} w_k f_k(X) = \sum_{k=1}^{K(t)} w_k [b_{0k} + b_{1k}x_1 + \dots + b_{ik}x_i + \dots + b_{nk}x_n] \quad (5.15)$$

where $B_k = [b_{0k}, \dots, b_{ik}, \dots, b_{nk}]^T$ is the parameter vector of the consequent node C_k ; and w_k is the normalized firing strength at the normalization node N_k .

GSETSK adapts a localized version of the recursive linear-least-squares (RLS) algorithm [45] to reduce the space complexity and the computation cost as well as to enhance the training speed. Assuming a rule R_k stays in the fuzzy rule base after T training samples, and assuming that GSETSK has only two inputs x_1 and x_2 , a local approximation that can represent the input-output relationships at the consequent

node C_k is shown in Equation (5.16).

$$\begin{bmatrix} w_k(1) & w_k(1)x_1(1) & w_k(1)x_2(1) \\ \vdots & \vdots & \vdots \\ w_k(t) & w_k(t)x_1(t) & w_k(t)x_2(t) \\ \vdots & \vdots & \vdots \\ w_k(T) & w_k(T)x_1(T) & w_k(T)x_2(T) \end{bmatrix} \times \begin{bmatrix} b_{0k} \\ b_{1k} \\ b_{2k} \end{bmatrix} = \begin{bmatrix} w_k(1)d(1) \\ \vdots \\ w_k(t)d(t) \\ \vdots \\ w_k(T)d(T) \end{bmatrix} \quad (5.16)$$

Equation (5.16) can be represented in the form in Equation (5.17).

$$A \times B = D \quad (5.17)$$

If a_p is the p^{th} row of the matrix A and using RLS, B can be iteratively estimated, as shown in Equation (5.18).

$$\begin{aligned} B_{(p+1)} &= B_p + C_{(p+1)}a_{(p+1)}^T(d^{(p+1)} - a_{(p+1)}B_p) \\ C_{(p+1)} &= \frac{1}{\lambda} \left[C_p - \frac{C_p a_{(p+1)}^T a_{(p+1)} C_p}{\lambda + a_{(p+1)} C_p a_{(p+1)}^T} \right] \end{aligned} \quad (5.18)$$

with initial condition $C_0 = \theta \mathbf{I}$ where θ is a large positive number and $\mathbf{I}$ is the identity matrix of dimension $\beta \times \beta$, where β is the number of consequent parameters of one rule, and λ is the forgetting factor. For each newly created rule $R_{K(t+1)}$, its parameters are determined by the weighted average of the parameters of the other rules [11]. The weights are the normalized firing strengths of the existing rules. More specifically, the parameters for the rule $R_{K(t+1)}$ are initialized as shown in Equation (5.19).

$$b_{i,K(t+1)} = \sum_{k=1}^{K(t)} w_k b_{i,k} \quad i = 1, \dots, n \quad (5.19)$$

where:

$[b_{0k}, \dots, b_{ik}, \dots, b_{nk}]^T$ is the parameter vector of the rule R_k ; and

w_k is the normalized firing strength of the rule R_k .

5.2 Evolving Fuzzy Interpolation and Extrapolation Takagi-Sugeno-Kang (eFIE-TSK) Model

Based on the defined structure of the GSETSK model [89], reviewed in Section 5.1, the evolving fuzzy interpolation and extrapolation Takagi-Sugeno-Kang (eFIE-TSK) model is proposed and implemented, extended from GSETSK model, in this section. The extension is a version of GSETSK with fuzzy interpolation and extrapolation (FIE) incorporated instead of the conventional fuzzy inference method. With input data being found beyond the coverage of existing rule base, the proposed FIE method will allow higher possibility in inferencing instead of producing a null output. The eFIE-TSK model can be represented as follows.

5.2.1 Forward Reasoning

Layer 1: Input Layer

The input layer consists of i nodes where i represents the number of input antecedents. The input x_i represents the observation given at the i th antecedent with Z_i^1 representing the input node of x_i and n represents the number of inputs, as shown in (5.20).

$$Z_i^1 = x_i, \quad i = 1, \dots, n \quad (5.20)$$

Layer 2: Input Linguistic Layer

In GSETSK, the matching degree of the input (x_i) is measured with its corresponding linguistic node using a Gaussian membership function. In this implementation, the matching degree is measured in terms of the distance between the fuzzy set's normal point to the given input (x_i).

The normal point, represented as $A_{i,j}$, of each fuzzy sets in the rule base are represented by the mean of each Gaussian membership function where j represents the fuzzy set of the antecedent. Computations of the linear distances between the observation x_i^* and the fuzzy sets j are performed. The matching degree is referred

to as individual weight $w_{A_{i,j}}$ and is derived as follows:

$$Z_{i,j_i}^2 = w_{A_{i,j}} = 1 - \left| \frac{d(x_i^*, A_{i,j})}{\sum_{j=1}^{m_i} |d(x_i^*, A_{i,j})|} \right| \quad (5.21)$$

where m_i represents the number of fuzzy sets in dimension i .

Layer 3 Rule Layer

Each rule k consist of n number fuzzy sets in the antecedent and each rule weight is computed as the average of individual weights of these fuzzy sets.

$$Z_k^3 = r_k = \frac{\sum_{i=1}^n Z_{i,j_i^k}^2}{n} \quad (5.22)$$

The κ -nearest rules are selected based on the highest rule weights r_k .

Layer 4: Normalization Layer

After the selection of κ -nearest rules, the normalised weight of each rule, W_κ , will be computed as follows:

$$Z_\kappa^4 = W_\kappa = \frac{r_\kappa}{\sum_{j=1}^\kappa r_j} \quad (5.23)$$

where W_κ represents the weight of the selected rule κ .

Layer 5: Consequence Layer

With the normalized rule weight W_κ , it is then applied in Equation (5.24) to compute the individual output from each rule k .

$$Z_\kappa^5 = W_\kappa \cdot f_k(X), \quad k = 1, \dots, \kappa \quad (5.24)$$

where $f_k(X)$ is a linear function of each rule consequent node.

In order to derive the consequent output range required in the final step in layer 6, the κ -nearest rules will be fired using Equation (5.24) with $W_\kappa = 1$. The output range, $d(y_{kn}, y_{1n})$, is then computed using the linear difference between the largest and the smallest results of these outputs.

Layer 6: Summation Layer

$$Z^{6'} = y' = \sum_{j=1}^{\kappa} Z_j^5 \quad (5.25)$$

The intermediate consequent output y' is computed using Equation (5.25). However, this output is an intermediate result, which as explained in Section 3.1.5 requires a shift operation in order to generate a plausible result.

In order to deduce the shift ratio, intermediate observations have to be computed for each dimension. This intermediate position is to verify the amount of shift that has been incurred due to the possible imbalance distribution of κ -nearest rules. An intermediate weight, $w'_{A_{i,\kappa}}$, for each fuzzy set in each dimension i of the κ -nearest rules is computed by Equation (5.26) using the individual weight, $w_{A_{i,\kappa}}$, computed from Equation (5.21).

$$w'_{A_{i,\kappa}} = \frac{w_{A_{i,\kappa}}}{\sum_{j=1}^{\kappa} w_{A_{i,j}}} \quad (5.26)$$

The intermediate position, $x'_{i,\kappa}$, of the observation in each dimension i using the κ -nearest rules is computed using Equation (5.27),

$$x'_{i,\kappa} = \sum_{j=1}^{\kappa} w'_{A_{i,j}} A_{i,j} \quad (5.27)$$

where $A_{i,j}$ represents the κ -nearest fuzzy sets. Subsequently, the individual dimension's shift ratio, δ_{x_i} , is computed using Equation (5.28).

$$\delta_{x_i} = \frac{d(x_i, x'_{i,\kappa})}{d(A_{il\kappa}, A_{im\kappa})} \quad (5.28)$$

where $d(A_{il\kappa}, A_{im\kappa})$ represents the distance between the κ -nearest rules of the dimension i , with l and m representing the largest and minimum representation value of the fuzzy sets from κ -nearest rule, respectively.

The final shift ratio δ_x is computed by Equation (5.29) using the individual shift ratio δ_{x_i} across n -dimension.

$$\delta_x = \frac{\sum_{i=1}^n \delta_{x_i}}{n} \quad (5.29)$$

The intermediate output derived from Equation (5.25) needs to be shifted as shown in Equation (5.30) with the output range $d(y_{kn}, y_{1n})$ computed in Layer 5. Therefore, the final output, Z^6 , is computed as follows:

$$Z^6 = Z^{6'} + \delta_x(d(y_{kn}, y_{1n})) \quad (5.30)$$

The backward computation described in Section 5.1.2 and parameter learning in Section 5.1.3 are used in this implementation, and hence not repeated in this section.

The next section will put the proposed model to test in terms of accuracy and handling concept drift and shift using different experiments settings.

5.3 Experiments and Results

In this section, the proposed eFIE-TSK model will be tested on three different experiments. They are the Mackey-Glass time series prediction, non-linear system identification problem and glucose-insulin prediction.

5.3.1 Mackey-Glass time series prediction

As mentioned in Section 4.3.1, the Mackey-Glass time series is often used in time series prediction and is defined by Equation (5.31).

$$\dot{x}(t) = \frac{ax(t - \tau)}{1 + x^{10}(t - \tau)} - bx(t) \quad (5.31)$$

In this experiment the values of the variables in the time series are $a=0.2$, $b=0.1$, time lag $\tau=17$ and $x(0) = 1.2$, as shown in Equation (5.32), which was originally proposed in [78].

$$\frac{dx}{dt} = \frac{0.2x(t - 17)}{1 + x^{10}(t - 17)} - 0.1x(t) \quad (5.32)$$

A total of 6000 observations were generated and the selected observation range of $201 \leq t \leq 3200$ and $5001 \leq t \leq 5500$, uniformly distributed within the range of $[0.4, 1.4]$, were used as the training and testing data set, respectively. The data set used in

Table 5.1: Mackey-Glass: RMSE result for all models

Model	Learning mode	No. of rules	Test RMSE
McFIS [104]	Online	6	0.0068
SPLAFIS [90]	Online	30	0.0632
GENEFIS [95]	Online	18	0.0634
SeroFAM [110]	Online	52	0.0781
eTS [11]	Online	99	0.0806
simpl_eTS [10]	Online	21	0.0851
SaFIS [98]	Online	21	0.0860
GSETSK [89]	Online	22	0.0614
eFIE-TSK	Online	26	0.0679

the prediction is formed up with $[x(t - 18), x(t - 12), x(t - 6), x(t)]$ as the inputs to predict the output of $x(t + 85)$. A list of existing techniques were used to benchmark the results accuracy as shown in Table 5.1.

As shown in the table, 26 rules were recorded for eFIE-TSK at the end of the testing and this is comparable to most of the models like simpl_eTS, SaFIS, GENEFIS and SPLAFIS. Lower RMSE was observed from eFIE-TSK as compared to eTS, simpl_eTS, SaFIS and SeroFAM models, but it was slightly higher than GENEFIS and SPLAFIS.

The best result was produced by McFIS which used a novel self-regulatory learning mechanism that decides what to learn, how to learn, and when to learn. The size of the rule base was also reported to be the smallest. This model outperforms all other models significantly with its novel learning mechanism.

Figure 5.3: Test result of eFIE-TSK on Mackey-Glass time series experiment.

Figure 5.3 shows the result plot from the 500 testing samples of eFIE-TSK prediction against the actual output. The predicted output can be observed to follow the

general trend of the actual output, however, it is not a perfect match at all sample points.

The objective of this experiment is basically to show that the prediction accuracy of the proposed eFIE-TSK model is comparable to other recent works despite using an approximate reasoning method based on existing nearest rules. The following experiment will first compare the basic accuracy of the proposed model and additionally compare the accuracy when greater shift in the data distribution is observed.

5.3.2 Non-linear system identification problem

This experiment has been setup under two different settings to test the proposed eFIE-TSK model in terms of accuracy and the capability of handling greater change in data distribution. This experiment investigates the ability to approximate a non-linear dynamic plant with non-time-varying characteristics as described in [58] and [98]. The plant to be learnt is defined by the difference equation as shown in Equation (5.33).

$$y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t) \quad (5.33)$$

where:

$u(t) = \sin(2\pi t/100)$ is the current input signal; and

$y(t)$ is the current output signal of the system.

Following the description in [98], 50,000 training and 200 testing data pairs are generated with initial conditions of $(u(t), y(t))$ as $(0, 0)$, where $u(t) \in [-1.0, 1.0]$ and $y(t) \in [-1.5, 1.5]$. Table 5.2 shows the result of the proposed eFIE-TSK model compared to other models.

The proposed eFIE-TSK has achieved an accuracy that is better than all Mamdani models mainly because TSK type fuzzy systems are generally more expressive and accurate compared to Mamdani fuzzy systems. eFIE-TSK has comparable RMSE and number of rules compared to the other TSK models. This is because eFIE-TSK uses approximation from the nearest rules instead of conventional fuzzy systems that is used by the other models which is more precise.

Table 5.2: Results of online identification of a non-linear system.

Model	Type	No. of Rules	RMSE
FALCON-ART [72]	Mamdani	289	0.1379
eFSM [116]	Mamdani	14	0.073
EFuNN [60]	Mamdani	18	0.0578
DENFIS [61]	TSK	32	0.021
GSETSK [89]	TSK	11	0.0119
SONFIN [58]	TSK	10	0.013
SAFIS [98]	TSK	8	0.0116
Simpl_eTS [10]	TSK	18	0.0122
eTS [11]	TSK	19	0.0082
McFIS [104]	TSK	5	0.0023
eFIE-TSK	TSK	11	0.0218

Figure 5.4: Test result of eFIE-TSK on online identification of a non-linear system experiment.

Figure 5.4 shows the modelling results of the eFIE-TSK model. From the plot, the match between predicted output against the actual output is closely overlapping with the RMSE of 0.0218.

In the following, the experiment conducted will have four levels of varying disturbance (δ_A) added at a specific time range. This additional disturbance will be able to make the original non-linear system defined in Equation (5.33) to be time-varying. Four existing evolving models, namely: SPLAFIS [90], GENEFIS [95], McFIS [104], and GSETSK [89], will be used in this section to compare the result of the proposed eFIE-TSK model. The properties of the time-varying non-linear system are described in Equation (5.34).

$$y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t) + f(t) \quad (5.34)$$

where:

$u(t) = \sin(2\pi t/100)$ is the current input signal;

$y(t)$ is the current output signal of the system; and

$f(t)$ is the added disturbance signal defined as follows:

$$f(t) = \begin{cases} \delta_A, & 1001 \leq t \leq 2000 \\ 0.0, & \text{otherwise} \end{cases} \quad (5.35)$$

where $\delta_A = [0.0, 0.5, 1.0, 1.5]$ representing four different level of disturbances added between $1001 \leq t \leq 2000$ in the following test scenarios as shown in Figure 5.5.

Figure 5.5: Behaviour of system with four different levels of δ_A included between $1001 \leq t \leq 2000$.

To minimise the error incurred, the common practice is to find the optimal model parameters by using the existing data. However, in reality, future data distributions are unknown. Therefore, it is common that the collected data may not include varying disturbances or distribution. In this section, the four existing evolving models will use the time-varying non-linear system with $\delta_A = 0$ to find the set of optimal parameters. This set of data with $\delta_A = 0$ is similar to a non-linear system without any disturbance,

hence possibly the type of data that may be commonly collected. With the chosen optimal parameters, the test is extended other δ_A values. This experiment is to test the handling capability of the models when different level of disturbance is present in the online stream of data. Since the proposed model is extended from GSETSK, it will hence adopt the same parameters that was chosen by GSETSK.

Table 5.3: Results of time-varying online identification of a non-linear system.

Model	$\delta_A = 0.0$		$\delta_A = 0.5$		$\delta_A = 1.0$		$\delta_A = 1.5$	
	No. of Rules	RMSE	No. of Rules	RMSE	No. of Rules	RMSE	No. of Rules	RMSE
SPLAFIS	45	0.0661	32	0.2516	18	0.3224	23	0.4530
GENEFIS	10	0.0827	9	0.2597	8	0.4434	13	0.5669
McFIS	7	0.0162	15	0.0841	15	0.0949	15	0.1145
GSETSK	5	0.0466	7	0.0659	7	0.0943	3	0.1247
eFIE-TSK	5	0.0583	7	0.0730	11	0.0678	8	0.0802

Table 5.3 shows the RMSE of each model under different δ_A setting. Under the column with $\delta_A = 0.0$, it can be observed that McFIS has achieved the best accuracy and a considerably small rule size. GSETSK has the next best accuracy with the smallest rule size. The proposed eFIE-TSK model has also contain the smallest rule size but with a slightly higher RMSE as compared to GSETSK. Both GENEFS and SPLAFIS achieved a considerably low RMSE, however SPLAFIS had created a larger rule size. As shown in the close up to $501 \leq t \leq 1001$ in Figure 5.6, the results of from all the models are highly identical. This is because, parameters used in this setting was specifically generated using the non-linear data with $\delta_A = 0.0$.

As the disturbance δ_A increases, the accuracy achieved by all models can be observed to be poorer. As the models' parameters used in this experiment remains the same, which is specifically chosen for δ_A to be 0.0, the modelling capability will be affected. As mentioned in the beginning of this experiment, in order to be realistic with respect to the restrictions of available data, the best parameters can only be tuned based on these data. Modelling capability of the models will hence be reflective with changing δ_A on the data. As eFIE-TSK uses the fuzzy interpolation and extrapolation technique, in cases when new data are found to be different from the current rules, an approximation of result can be achieved, whereas models with conventional fuzzy techniques may face poor or even null results in extreme cases.

Time varying nonlinear system with $\delta_A = 0.0$

Figure 5.6: Prediction results between $501 \leq t \leq 1001$ where $\delta_A = 0.0$.

Figure 5.7: Prediction results between $801 \leq t \leq 1800$ where $\delta_A = 1.5$.

In the four different level of disturbances, the largest inserted disturbance was $\delta_A = 1.5$. From Figure 5.5, it can be observed that during the time period when disturbances were added to the system, the entire data distribution actually shifts out of the original distribution. This is the extreme case whereby sudden data shift occurs. Therefore the results from the experiment where $\delta_A = 1.5$ have the highest overall RMSE from all the models. The eFIE-TSK model had achieved the lowest RMSE in this setting with a relatively small rule size.

Figure 5.7 shows the results of McFIS, GSETSK and eFIE-TSK models at the points when disturbance of $\delta_A = 1.5$ was introduced at $t = 1001$ to the system. From the plots, when disturbance was introduced, McFIS was unable to immediately deal with such drastic changes. Hence contributing to the prolong gaps between the actual and predicted output. However, due to the evolving structure of this model, McFIS was able to rapidly adapt to the current data distribution to produce fitting results from $t = 1201$ onwards.

The plots showing the results of GSETSK and eFIE-TSK did not have prolong gaps when disturbance of $\delta_A = 1.5$ was introduced at $t = 1001$ to the system. However, instances of instant wide gaps could be observed instead. These gaps were found to be wider in the GSETSK model as compared to the eFIE-TSK model. The wide gaps in predicted result of GSETSK model was because of the conventional fuzzy inference technique which required a well coverage rule base. In the situation when the data shifts too drastically out of the rule base coverage, a poor inference will be resulted. On the other hand, eFIE-TSK model used the fuzzy interpolation and extrapolation method that was based on the nearest neighbouring rules to extrapolate a plausible result. Hence the gaps produced by the eFIE-TSK model were slightly narrower. It should also be noted that due to the learning capability of GSETSK, the modelling of the current distribution was subsequently adapted from $t = 1100$ onwards.

From Table 5.3, eFIE-TSK model may not perform better when data distribution was not largely shifted ($\delta_A = 0.0$ and $\delta_A = 0.5$). However, when data shifts greatly ($\delta_A = 1.0$ and $\delta_A = 1.5$), the handling capability of the model can be significantly better than the other models. In the next experiment, the glucose-insulin prediction

experiment will be discussed to investigate the fuzzy interpolation and extrapolation capability.

5.3.3 Glucose-insulin prediction

Four groups of glucose-insulin data had been generated from GlucoSim [41] and used in the experiments in Section 4.3.3, and these data will be used in this section. In this section, the models are initially trained with ten days of glucose-insulin data of Group A (Normal dietary habit) and tested against all the four groups of outsampled (testing) data sets representing the nine days of unseen data for Normal dietary habit, Under-eating dietary habit, Over-eating dietary habit, and Irregular dietary habit.

The initial training is basically to provide the predictive models with a basic knowledge before being tested using the testing sets. The data vectors used for both training and testing is given as $[G(t - 15), G(t - 10), G(t - 5), G(t); IR(t + 5)]$ where $G(\cdot)$ represents blood glucose readings and $IR(t + 5)$ is the 5 minutes ahead insulin residual output from the model in response to meal intakes.

In this section three models were used to benchmark the performances of the proposed eFIE-TSK model, namely: SPLAFIS [90], McFIS [104], and GSETSK [89]. The parameters used in each model were chosen based on the best accuracy of the normal dietary habit testing data. As eFIE-TSK model was extended from GSETSK, it will adopt the same parameters chosen by GSETSK.

Table 5.4: RMSE of the models in predicting insulin residual under different dietary habits and ranking of the models for having the smallest rule size and lowest RMSE as shown in the parenthesis.

Dietary Habit:	Normal		Under-eating		Over-eating		Irregular		
Model	Rule size	RMSE	Rule size	RMSE	Rule size	RMSE	Rule size	RMSE	Avg. Rank
SPLAFIS	103(4)	1.356(2)	118(4)	1.554(2)	130(4)	3.658(4)	91(4)	6.866(2)	3.25(4)
McFIS	35(3)	0.784(1)	35(2)	0.935(1)	35(3)	1.955(1)	35(3)	6.951(3)	2.125(1)
GSETSK	22(1)	1.71(4)	23(1)	2.786(4)	22(1)	3.278(3)	22(1)	20.98(4)	2.37(3)
eFIE-TSK	31(2)	1.539(3)	35(2)	1.795(3)	34(2)	3.074(2)	34(2)	6.601(1)	2.125(1)

Table 5.4 shows the performance of the testing results from different dietary habits in terms of RMSE and rule size. SPLAFIS had achieved comparable RMSE as com-

pared to the other three models across all the different models. However, a relatively larger rule base was required. McFIS on the other hand achieved the best accuracy for normal, undereat and overeat dietary habits and required a much smaller rule base as compared to SPLAFIS. GSETSK used the smallest rule base size but the accuracy achieved was not the best in any testing situation. eFIE-TSK used similar rule size as McFIS but with a poorer accuracy for all testing data except the irregular dietary habit.

From the table, rankings are shown within the parenthesis beside the actual size and RMSE of each model for each testing set. The models were ranked in terms of rule size and RMSE. The lowest average ranked is SPLAFIS model. In terms of RMSE, SPLAFIS had actually fared well, however, large rule size was required to produce the low RMSE. Both McFIS and eFIE-TSK had been ranked first by having a tie in the average rank.

As discussed previously in Section 4.3.3, irregular dietary habit displayed the extreme dietary behaviour which was beyond the normal dietary habit. In the irregular dietary habit testing set, the fluctuation of the blood glucose level can be much greater than the normal dietary habit. From Table 4.5, the data used for the initial training (normal dietary habit) has the blood glucose inputs with an upper limit of near 300mg/dl. The irregular dietary habit has the blood glucose inputs with the upper limit of nearly 600mg/dl, which is nearly twice the coverage of the normal dietary habit. The rules learnt during the initial training does not include any possible rules that will be able to handle such inputs.

Figure 5.8 illustrates the predicted results of all nine days from the models against the actual data. From the figure it can be observed that the predicted results during the first day looks erroneous. Figure 5.9 shows the zoom in to the first day of the predictions showing the entire Day 1 result and Figure 5.10 shows the zoom in to the first three peaks of the results.

The initial inputs of the training data has a blood glucose upper limit of nearly 300mg/dl but the current data seen at the first peak was up to 350mg/dl. Therefore it can be observed that McFIS model starts to have a kind of learning behaviour,

trying to adapt to the current output. On the other hand, both GSETSK and eFIE-TSK had managed to produce a close mapping to the expected output for the first peak when the inputs were not too far off from the limits. This shows that to a certain extend, both GSETSK and eFIE-TSK were still able to handle concept drift where data distribution does not shift away too drastically.

The second peak as illustrated in Figure 5.10 shows a higher blood glucose peak (450mg/dl) as compared to the first peak. McFIS model had behave similarly as shown in the first peak trying to adapt to the change in the output. GSETSK and eFIE-TSK behaved differently as compared to the first peak. A higher degree of change in the data distribution caused a concept shift where data had actually shifted away further than a concept drift. This put our proposed model to test the capability to handle concept shift with direct comparison with GSETSK that used the same learning methodology.

Figure 5.8: Predicted insulin residual by the models for the span of 9 days.

From Figure 5.10, the results of GSETSK at the second peak fluctuates drastically. With conventional fuzzy inferencing method, the predicted output of the model was unable to produce and learn the actual output effectively. However, eFIE-TSK had shown otherwise. Although an output gap could be visibly seen between the prediction from eFIE-TSK and the actual result, but the gap was narrower as compared to the gap produced by GSETSK model. With a closer prediction output, the model was able to learn more effectively and hence at the next prediction the output gap closed up.

Figure 5.9: Predicted insulin residual by the models for Day 1.

Figure 5.10: First three peaks of predicted insulin residual by the models for Day 1.

The third peak was plotted in this figure for the purpose of showing that with the initial training and after the learning of the two peaks, the accuracy of the three models were still highly accurate. However, in Day 2, another higher peak (nearly 500mg/dl) was observed as shown in Figure 5.11, and a closer look at the first peak of Day 2 was illustrated in Figure 5.12.

The result from McFIS shows that at this moment, the model was still unable to produce any close prediction even with the learning from the previous peaks. This was because the data distribution of this new peak was never seen and the model had to gradually learn which was similarly observed in the first two peaks of Day 1. The result from GSETSK was still erroneous. This may be because during the first two peaks of Day the model did not managed to learn adequately due to the large errors incurred. On the other hand, eFIE-TSK had managed to produce fairly better results

Figure 5.11: Predicted insulin residual by the models for Day 2.

Figure 5.12: First peak of predicted insulin residual by the models for Day 2.

than the other two models. With the rules created during the first two peaks, the model was able to reasonably produce an output, although not perfect, that was close to the actual output. As the prediction error was reduced, the learning process of the eFIE-TSK model would be more effective.

From Figure 5.8, the next highest peak is found happening at Day 8. The result of Day 8 is illustrated in Figure 5.13 showing the predicted and actual output. After being exposed to the extreme data distribution of irregular dietary habit from the first two days, when similar excursion of data distribution was observed in Day 8, all the models were able to handle such situation and produces mapping results. This was mainly because the rule base had already created the necessary rules during the first two days when these extreme data were experienced.

Figure 5.13: Predicted insulin residual by the models for Day 8.

5.4 Summary

This chapter proposed the eFIE-TSK model which used the learning mechanism from GSETSK and the proposed fuzzy interpolation and extrapolation (FIE) inferencing method proposed in Chapter 3 by extending it from the Mamdani fuzzy model to the TSK fuzzy model. The Mackey-Glass time series prediction, non-linear identification system and glucose-insulin prediction experiments were used to benchmark the performance of the proposed model against other established models.

The Mackey-Glass experiment was used to show the performance of eFIE-TSK model with other existing models. From the result shown in Table 5.1, the performance of the proposed model was highly comparable to other established models in terms of RMSE and rule size.

Comparable performance by eFIE-TSK was also shown in the non-linear identification system experiment. This experiment was further extended to test the concept drift and shift handling capability of the proposed model and some existing techniques. Four different levels of disturbances δ_A were introduced to the data stream between $1001 \leq t \leq 2000$, and each model selected their best parameter by using the data with $\delta_A = 0.0$ or basically no disturbance. With the increase of δ_A , the results produced by all the models degrades. Although eFIE-TSK had a moderately good performance when δ_A was 0.0 and 0.5, it had shown to be the least degrading as compared to the other models when δ_A increases. The result plot as illustrated in Figure 5.7 shows that eFIE-TSK had actually been able to approximate a better

result as compare to the GSETSK which basically used the same model parameters and learning mechanism. With a smaller error, the eFIE-TSK model was able to learn more efficiently.

The performance of eFIE-TSK was next tested using the glucose-insulin experiment. As per previous experiment, the parameters of each model were selected from the testing experiments (normal dietary habit). It can be observed from Table 5.4 showing that under the normal dietary habit, the RMSE recorded was the lowest among all four different dietary habits for all models. Both McFIS and eFIE-TSK had been ranked to be in the first place. A more detailed discussion was made using the irregular dietary habit to show the capability in handling concept drift and shift by the proposed model as compared to conventional fuzzy method. It was observed in Figure 5.10 and Figure 5.12 that eFIE-TSK model had tried to extrapolate an output that was closer to the actual output as compared to the output from the GSETSK model. In return with a smaller error, eFIE-TSK was able to learn a new rule with a better output that was shown in the next instance of the prediction.

In conclusion to this chapter, with the fuzzy interpolation and extrapolation technique incorporated to a neural network or learning model, the accuracy of the output may not be better than other models but still maintained a highly comparable result. However, when in view of handling concept drift and shift, the proposed eFIE-TSK model was able to produce better result compared to the GSETSK model that uses conventional fuzzy inference method but identical learning methodology.

Chapter 6

Conclusions and Future works

This chapter provides some concluding discussions on the topics presented in this thesis. A summary of contributions will be presented in Section 6.1, followed by directions for possible future works in Section 6.2.

6.1 Conclusions

In the thesis, two objectives were set in the introduction chapter which included addressing issues arising from sparse fuzzy rule systems and changes in the data distribution.

The issues to be addressed in the thesis, which were identified in Chapter 1 and are reproduced here, as follows:

A - Sparse fuzzy rule systems:

1. Propose a new fuzzy interpolation technique that is able to handle multiple antecedents and multiple rules.
2. Extend the technique with interpolation and extrapolation capability.
3. Implement fuzzy neural networks that incorporate the proposed fuzzy interpolation and extrapolation techniques using different learning mechanisms.

B - Concept drift/shift:

1. Handling concept drift/shift with proposed fuzzy extrapolation technique.

In addressing the sparse fuzzy rule issue, an interpolation and extrapolation technique (addressing Part A (1) and (2)) was proposed in Chapter 3. The proposed

method was extended from [29] where they had claimed to be able to handle multiple antecedents and multiple rules. However, from their presented result, this claim was not valid. In our proposed method, two changes were made to the original work to make the handling valid. The two changes were the weightage formulation and the introduction of the shift process. The closest work that had the same capability was presented in [50], but the method requires predefined domain boundaries which was not feasible to be extended to non-stationary domain like the presented diabetes problem.

The proposed fuzzy interpolation and extrapolation technique was incorporated into as the FIE-FITSK model in Section 4.2 and the eFIE-TSK model in Section 5.2, thus addressing (Part A (3)). Each of these implementations were tested and the accuracy and concept drift/shift handling capabilities were examined, addressing (Part B (1)).

In the implementation of FIE-FITSK model, the accuracy of the model was tested against commonly used regression problems of Mackey-Glass time series problem and the non-linear system identification problem. Under normal circumstances, although the proposed model did not performed with the best result, it was still able to produce comparable results in terms of accuracy and rule size. A modified non-linear system identification which includes disturbances of varying levels were then used to test the capability of handling concept drift/shift by offline models. The popular ANFIS [52] was used to benchmark against the proposed model and the FITSK model [96]. From Table 4.3, it was observed that as the disturbance δ_A increases, the RMSE of each model will increase. This was because, the models were trained with no disturbance and tested on the data that are of different distribution. The results were plotted in Figure 4.4 with the zoom in to the largest disturbance of $\delta_A = 1.5$ in Figure 4.5. The ability of FIE-FITSK to produce a closer result as compared to the FITSK was shown. This explained why FIE-FITSK had achieved a lower RMSE.

The model was next tested on the glucose-insulin prediction experiment. The advantage of the FIE-FITSK model over the FITSK model was mainly observed when the model was trained using Group A, Group B and Group C dietary data and tested

on the Group D* dietary data. FIE-FITSK was able to predict out of the trained boundaries of the model to produce a plausible result. The worst result obtained by both models was when the models were trained with Group B dietary and tested on Group D* dietary. This was because the data boundary of Group B dietary was the smallest while the data boundary of Group D* dietary was the largest. Even with the extrapolation capability in FIE-FITSK, when the given inputs were too far off from the trained model, it will not be able to produce any result. The training data used with the highest testing accuracy was the Group D dietary and this model was used in the closed-loop insulin scheduler application.

Figure 4.17 shows the results from both FITSK and FIE-FITSK using the Group D trained models of the closed-loop insulin scheduler application. Deviation between the simulated diabetic person and the healthy reference can be observed. The objective of comparing the results from the models with the healthy reference model was to show that the models were able to have the similar behaviour of blood glucose fluctuation. This tells us that the insulin scheduler was able to control the amount of necessary release at each time point so as to control the patient to stay out of the hyper and hypoglycaemic conditions. However, in reality, the possibility of gathering the type of Group D data is near impossible. The risk involved in getting a person or an actual diabetic patient to collect irregular dietary habit was too high. Irreversible medical condition that are life threatening may occur.

Therefore another test was conducted which uses the Group A training data that represents a normal dietary habit. Both models were trained using Group A data and similar experiment was conducted. Figure 4.18 shows the result of the experiment. The blood glucose fluctuation behaviour observed from the FIE-FITSK was still similar to the healthy reference model. However, the result from the FITSK model could be observed to be totally different. With Group A training set, that was safer to be collected in reality, the proposed FIE-FITSK model had shown that it could produce better modelling ability than the FITSK model.

Evolving models have the online learning capability where new rules can be created when current data is not well represented by the current rule base. However, creation

of rules are rightfully after the current prediction was made, which means that the current prediction may not be good enough if the current inputs are significantly different from the existing rule base (concept drift/shift). The ongoing creation of rules in an evolving model may cause a huge rule base which will slows down prediction and waste memory space. Pruning techniques were incorporated by most models to remove redundant rules that were not useful. Most pruning mechanism selects rules to be pruned based on the rule potential. This rule potential was reinforced when the rules were used, or decayed with a forgetting factor or half-life generation. The pruning did not consider the rule coverage issue. Therefore, pruning of rules may also cause the rule base to be sparse.

The proposed eFIE-TSK model, incorporated with the fuzzy interpolation and extrapolation (FIE) capability, was developed in order to address the issues and placed into test. This model made use of the same learning mechanism of the GSETSK model [89] and hence this model was used in all experiments to show the direct implication of the FIE ability. The Mackey-Glass time series problem was used to benchmark the proposed eFIE-TSK model against other existing models. The result as shown in Table 5.1 had shown that the proposed model did not achieved the best result but fares comparably with other models in terms of RMSE and rule size.

The next benchmarked experiment was the non-linear system identification problem. Similarly to the result from the previous experiment, eFIE-TSK did not achieve the best accuracy but the RMSE was comparable considering the small rule size to all other models as shown in Table 5.2. This result was not unexpected as FIE itself was an approximation method instead of the precise computation like the TSK implementation but it was better than all the other Mamdani models.

The next experiment was identical to the non-linear system with disturbance that was used in the FIE-FITSK model. The experimental results were shown in Table 5.3 where each model used the best parameter that could be found to model the data without any disturbance on all the four different data sets. It could be observed from the table that as the increase in the level of disturbances, the accuracy of the models degrades. On the other hand, the level of degrading as shown by the proposed model

was not as great as the rest of the models. The main objective of this experiment setting was to investigate how the proposed model handles concept drift/shift. The zoom in as shown in Figure 5.7 shows that as compared to McFIS, eFIE-TSK had managed to produce a feasible result when input data were found far beyond the previously seen data. McFIS on the other hand had shown that once the input data was found to be far from the previously seen data, a poor result was computed. GSETSK had been observed to behave similarly to McFIS where at the first contact with the new data, poor result was produced. However due to the learning capability of the models, McFIS and GSETSK had gradually learnt to adapt to the new data distribution. This behaviour was also observed in eFIE-TSK as it had the same learning mechanism as GSETSK.

In the next experiment, the same diabetic data used in FIE-FITSK model was used to investigate the concept drift/shift handling capability of the proposed model. Using the same logic that normal dietary habit data can be safely collected, the best model parameters of all the models was derived from this training data. From Table 5.4, the accuracy was once again shown that the normal dietary testing set was the best. The irregular dietary habit data is the set with the greatest distribution difference as compared to the normal dietary habit and hence concept drift/shift was most highly possible to occur in this testing data set.

As illustrated in Figure 5.10 and Figure 5.12, when input data (blood glucose reading) was observed to escalate beyond the normal dietary range upper limit of 300mg/dl, McFIS prediction degrades and it could be seen that the model progressively tries to learn the new data trend. However when input data was beyond 350mg/dl, it could be observed that GSETSK behaved erratically. Huge drop in prediction was observed. This drop behaviour could be observed in eFIE-TSK as well but with a smaller difference as compared to the actual result. With the extrapolation capability, extrapolated results could be deduced based on the observation and the nearest rules involved.

6.2 Future Works

The solution to handle data sparsity is challenging but required. This thesis had examined the fusion of fuzzy interpolation and extrapolation technique with different types of neural network to handle sparsity and concept drift/shift. Due to the novelty of the models, many areas are yet to be fully examined, and there are some limitations in the current model.

Possible future research can consider the following directions: 1) an examination of the uncertainty handling capability of the fuzzy interpolation and extrapolation technique; 2) the complexity analysis of the fuzzy interpolation and extrapolation technique; 3) imprecision as shown in the accuracy from the experiments performed; 4) the formation of the rule base; and 5) range of domain used in computing shift.

In short term, examination of uncertainty handling can be investigated. In a situation when two nearest rules are used to interpolate a result, it has the similar operational behaviour of how type-2 fuzzy logic defuzzify itself to inference a result. Test can be carried out by using FIE-FITSK and eFIE-TSK models with other established type-2 FLS that have been shown to exhibit handling capability of higher degrees of uncertainties.

The complexity analysis of the fuzzy interpolation and extrapolation technique can be analysed in short term. This analysis work will be able to justify the amount of time required for completing a cycle of prediction in a sparse rule base as compared to a cycle perform in a dense rule base. This complexity study was not done in this thesis because optimisation was not a consideration when developing this model. There may be redundant cycles that may be optimised. Further work to refine this method has to be looked into.

However in long term aspect, the other directions can be explored which requires more research studies as these directions affects the learning model directly. Imprecision represented as the accuracy from all the experiments under normal situations were shown to be worse off compared to most existing TSK models. The reason that can be deduced is that the model, although using TSK representation, uses an approximation method to interpolate to produced results even when the rule base is not

sparse. A selection function can be imposed to choose when to interpolate based on the detection of sparsity. When the rule base is not sparse, the model can select to inference using the conventional way.

In this thesis, the generation of the rule base was determined by the original model. In FIE-FITSK, an evenly distributed CMAC structure was used where all interleaving nodes represent individual rules, while eFIE-TSK uses multidimensional scaling growing clustering (MSGC) [87] to determine the rule-base structure of the model. Clustering methods were used in determining the fuzzy sets centroid which represents the data distribution. Some commonly used clustering methods are k-means clustering [71], discrete incremental clustering [118], learning vector quantization [64], modified learning vector quantization [8], fuzzy C-means (FCM) [20], fuzzy Kohonen partitioning (FKP) [21], adaptive resonance theory (ART) [28], etc.

Figure 6.1: Suggested rule creation by converting the derived consequent as a rule.

With the use of FIE, a new approach in expanding the rule base can be suggested which was mentioned in [86] as the dynamic rule creation. From the FIE method discussed in Chapter 3, an intermediate consequent fuzzy set from the rule base was created using the interpolation or extrapolation comparison of the observation to the nearest fuzzy sets. Figure 6.1 shows a simple two rule example, where Rule 1: IF x is A_1 THEN y is B_1 and Rule 2: IF x is A_2 THEN y is B_2 . If the observation is x is A^* with the derived consequent as B^* , a new rule can be simply created in the rule base

by taking the observation and derived consequent as a new rule, such as, Rule 3: IF x is A_3 THEN y is B_3 , where $A_3 = A^*$ and $B_3 = B^*$. However, a carefully selected threshold needs to be set and a strict criteria should be met before the creation of new rule using this approach. Overlapping fuzzy sets may be created in the rule base if the threshold and criteria are not carefully defined.

In the work from [50], a predefined domain range was made based on the overall data for normalising the nearest distance measure. This method was identified as not being directly applicable in problem domains where boundary of data can be defined. In this thesis, this domain range is defined to be the the distance computed between the n -nearest rules instead of the entire domain range. This allows a more conservative normalisation and highly based on the used knowledge. However, this range definition can be further analysed to determine the sensitivity of this parameter which will affect the overall performance. In an evolving model, domain range of each dimension can be updated based on the existing rules. When rules are pruned and fuzzy sets are no longer used, the domain range can be adjusted accordingly. Such a measure can be used to replace the predefined domain range that was used in [50].

Another possible direction is to examine hybrid networks. For example, the ensemble method, originally proposed by Hansen and Salamon in 1990 [44], made use of different neural networks for a single experiment and combined the results to solve a single problem. The set of data is separated into two different portions, with the first portion used to train the neural networks in finding the best parameter setting, and the second portion used to validate the effectiveness of the ensemble result. The results from individual neural networks are then used in a consensus scheme to decide the final output. The ensemble method was extended to evolving neural networks, where learning is incremental rather than batch mode, in [27, 36]. However these ensemble methods still face the same constraint when dealing with the concept shift issue. Therefore a hybrid of the ensemble method and a FIE based neural network could be incorporated in order to improve the accuracy during normal operation, while still providing the ability to infer when irregularity occurs.

Additionally, the fuzzy neural network techniques proposed in this thesis could be

applied to other applications where noise is prevalent. For example, the electroencephalogram (EEG) has been widely explored in recent years, ranging from physical robotic control [67, 93, 108] to emotional measurement of empathic understanding [91]. EEG is a non-invasive and relatively inexpensive method for collecting neurophysiological function from the brain.

Noise exists in two forms in an EEG application, namely: neuronal noise [39], and equipment noise [106, 107]. Two main sources of neuronal noise, known as thermal noise [103] and channel noise, are directly or indirectly linked to other sources of noise in the brain. At the microscopic level, a considerable amount of noise is present due to molecular agitation and collisions. The thermal noise has many consequences for the function of neuronal ion channels, where ion channels are defined as macromolecules which are embedded into a constantly fluctuating medium. The thermal fluctuation, within the channel and the molecule collisions, trigger spontaneous conformation changes which may cause the opening or closing of the channel. This phenomenon of vague opening and closing is known as channel noise.

Equipment noise is generated by the sensors that is attached to the skin of the scalp. The major noise contribution is from the conducting gel used to facilitate attachment, which may dry up or be unevenly spread on the surface which causes connection problems, hence contributing the noise. While the use of dry [107] or non-contact [106] EEG sensors has been proposed to reduce the noise, these methods only partially reduce the amount of noise. Thus, noise can be minimized but not eliminated. Hence, the proposed FNN in this thesis could possibly be applied to this application.

Bibliography

- [1] Continuous glucose monitors: Consumer guide 2016, . URL <http://main.diabetes.org/dforg/pdfs/2016/2016-cg-cgm-chart.pdf>.
- [2] Comparison of current continuous glucose monitors (CGMs), . URL <http://www.diabetesnet.com/diabetes-technology/meters-monitors/continuous-monitors/compare-current-monitors>. Last visited 29th December 2015.
- [3] *Global status report on noncommunicable diseases 2010*. WHO Press, 2010. ISBN 978-92-4-156422-8.
- [4] *Noncommunicable Diseases country profiles 2014*. WHO Press, 2014. ISBN 978-92-4-150750-8.
- [5] M. Anand, K. Piyush, and S. MG. Prediction Models for Global Solar Radiations, Diffused Radiations and Direct Solar Radiations Using ANFIS. In *Proceeding of International Conference on Intelligent Communication, Control and Devices*, pages 27–37. Springer, 2017.
- [6] I. Anderson and E. Jarawan. The Economic Costs of Noncommunicable Diseases in the Pacific Islands. Technical report, The World Bank, 2012.
- [7] K Ang and C Quek. RSPOP: Rough Set Based Pseudo Outer-Product Fuzzy Rule Identification Algorithm. *Neural Computation*, 17(1):205–243, 2005. doi: 10.1162/0899766052530857.
- [8] K. K. Ang and C. Quek. MLVQ: A modified learning vector quantization algorithm for identifying centroids of fuzzy membership functions. In Boris Igel'nik,

- editor, *Computational Modeling and Simulation of Intellect: Current State and Future Perspectives*, pages 485–509. Hershey: IGI Global, 2011.
- [9] K.K. Ang, Chai Quek, and M. Pasquier. POPFNN-CRI(S): pseudo outer product based fuzzy neural network using the compositional rule of inference and singleton fuzzifier. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 33(6):838–849, 2003. doi: 10.1109/TSMCB.2003.812850.
 - [10] P. Angelov and D. Filev. Simpl.eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models. In *Fuzzy Systems, 2005. FUZZ '05. The 14th IEEE International Conference on*, pages 1068–1073, 2005. doi: 10.1109/FUZZY.2005.1452543.
 - [11] P. Angelov, Victor J., Dourado D., and D. Filev. On-line evolution of Takagi-Sugeno fuzzy models. In *in Proc. 2nd IFAC Workshop on Advanced Fuzzy/Neural Control, Oulu, Finland*, pages 67–72, 2004.
 - [12] American Diabetes Association. Diagnosis and classification of diabetes mellitus. *Diabetes Care*, 27(Supplement 1):S5–S10, 2004.
 - [13] A. Atashi, N. Nazeri, E. Abbasi, S. Dorri, and M. Alijani_Z. Breast Cancer Risk Assessment Using Adaptive Neuro-Fuzzy Inference System (ANFIS) and Subtractive Clustering Algorithm. *Multidisciplinary Cancer Investigation*, 1(2): 20–26, 2017.
 - [14] E. Atlas, E. A. Grunberg, R. Nimri, M. Phillip, and S. Miller. MD-Logic Artificial Pancreas System - A pilot study in adults with type 1 diabetes. *Diabetes Care*, 33(5):1072 – 1076, 2010.
 - [15] P. Baranyi, T. D. Gedeon, and L. T. Kóczy. A general interpolation technique in fuzzy rule bases with arbitrary membership functions. In *Systems, Man, and Cybernetics, 1996., IEEE International Conference on*, volume 1, pages 510–515, 1996. doi: 10.1109/ICSMC.1996.569844.
 - [16] P. Baranyi, T. Domonkos, Y. Yeung, L. T. Kóczy, and L. Nadai. A new method for avoiding abnormal conclusion for α -cut based rule interpolation. In

- Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE '99. 1999 IEEE International*, volume 1, pages 383–388, 1999. doi: 10.1109/FUZZY.1999.793270.
- [17] P. Baranyi, L. T. Kóczy, and T. D. Gedeon. A generalized concept for fuzzy rule interpolation. *Fuzzy Systems, IEEE Transactions on*, 12(6):820–837, 2004. doi: 10.1109/TFUZZ.2004.836085.
 - [18] E R Bareiss, B W Porter, and C C Wier. PROTOS: an exemplar-based learning apprentice. In *Proceedings of the Fourth International Machine Learning Workshop*, pages 12–23, 1987.
 - [19] R. Bellazzi, G. Nucci, and C. Cobelli. The subcutaneous route to insulin-dependent diabetes therapy. *IEEE Engineering Medicine and Biology Magazine*, 20(1):54–64, 2001.
 - [20] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981. ISBN 0306406713.
 - [21] J.C. Bezdek, E.C.-K. Tsao, and N.R. Pal. Fuzzy kohonen clustering networks. In *Fuzzy Systems, 1992., IEEE International Conference on*, pages 1035–1043, 1992. doi: 10.1109/FUZZY.1992.258797.
 - [22] EL Bienenstock, LN Cooper, and PW Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2(1):32–48, 1982. ISSN 0270-6474.
 - [23] Taner Bilgi and I.Burhan Trksen. Measurement of membership functions: Theoretical and empirical work. In Didier Dubois and Henri Prade, editors, *Fundamentals of Fuzzy Sets*, volume 7 of *The Handbooks of Fuzzy Sets Series*, pages 195–227. Springer US, 2000. ISBN 978-1-4613-6994-3. doi: 10.1007/978-1-4615-4429-6_4.
 - [24] D. E. Bloom, E. T. Cafiero, E. Jan-Llopis, S. Abrahams-Gessel, L. R. Bloom, S. Fathima, A. B. Feigl, T. Gaziano, M. Mowafi, A. Pandya, K. Prettnner,

- L. Rosenberg, B. S. A. Z. Stein, and C. Weinstein. The Global Economic Burden of Noncommunicable Diseases, 2011.
- [25] E. A. Boland, M. Grey, A. Oesterle, L. Fredrickson, and W. V. Tamborlane. Continuous subcutaneous insulin infusion. a new way to lower risk of severe hypoglycemia, improve metabolic control, and enhance coping in adolescents with type 1 diabetes. *Diabetes Care*, 22(11):1779–1784, 1999.
- [26] M Botta, A Giordana, and L Saitta. Learning fuzzy concept definitions. In *Proc. 2nd IEEE International Conference on Fuzzy Systems*, pages 18–22, 1993.
- [27] L. Bueno, P. Costa, I. Mendes, E. Cruz, and D. Leite. Evolving ensemble of fuzzy models for multivariate time series prediction. In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, pages 1–6, 2015. doi: 10.1109/FUZZ-IEEE.2015.7338002.
- [28] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *Neural Networks, IEEE Transactions on*, 3(5):698–713, 1992. doi: 10.1109/72.159059.
- [29] Y. Chang, S.-M. Chen, and C.-J. Liau. Fuzzy interpolative reasoning for sparse fuzzy-rule-based systems based on the areas of fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 16(5):1285–1301, 2008.
- [30] C. Chen, C. Quek, and Q. Shen. Scale and move transformation-based fuzzy rule interpolation with interval type-2 fuzzy sets. In *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, pages 1–8, 2013. doi: 10.1109/FUZZ-IEEE.2013.6622351.
- [31] C. Chen, N. M. Parthalàin, Y. Li, C. Price, C. Quek, and Q. Shen. Rough-fuzzy rule interpolation. *Information Sciences*, 351:1 – 17, 2016. ISSN 0020-0255. doi: <https://doi.org/10.1016/j.ins.2016.02.036>.
- [32] S.-M. Chen and Y.-C. Chang. Weighted fuzzy interpolative reasoning for sparse

- fuzzy rule-based systems. *Expert Systems with Applications*, 38(8):9564 – 9572, 2011. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2011.01.138>.
- [33] S.-M. Chen and Y.-K. Ko. Fuzzy interpolative reasoning for sparse fuzzy rule-based systems based on alpha-cuts and transformations techniques. *Fuzzy Systems, IEEE Transactions on*, 16(6):1626–1648, 2008. doi: 10.1109/TFUZZ.2008.2008412.
 - [34] S.-M. Chen, W.-C. Hsin, S.-W. Yang, and Y.-C. Chang. Fuzzy interpolative reasoning for sparse fuzzy rule-based systems based on the slopes of fuzzy sets. *Expert Systems with Applications*, 39(15):11961 – 11969, 2012. doi: 10.1016/j.eswa.2012.03.065.
 - [35] S.-M. Chen, L.-W. Lee, and V. R. L. Shen. Weighted fuzzy interpolative reasoning systems based on interval type-2 fuzzy sets. *Information Sciences*, 248: 15 – 30, 2013. doi: 10.1016/j.ins.2013.05.002.
 - [36] E. Y. Cheu, C. Quek, and S. S. Ng. Evolving ensemble of fuzzy models. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 2668–2675, 2011. doi: 10.1109/FUZZY.2011.6007358.
 - [37] P. E. Cryer, S. N. Davis, and H. Shamoon. Hypoglycemia in diabetes. *Diabetes Care*, 26(6):1902–1912, 2003.
 - [38] J. De Jesus Rubio. SOFMLS: Online Self-Organizing Fuzzy Modified Least-Squares Network. *Fuzzy Systems, IEEE Transactions on*, 17(6):1296–1309, 2009. doi: 10.1109/TFUZZ.2009.2029569.
 - [39] Alain Destexhe and Michelle Rudolph-Lilith. In *Neuronal Noise*, volume 8 of *Springer Series in Computational Neuroscience*. Springer US, 2012. ISBN 978-0-387-79019-0. doi: 10.1007/978-0-387-79020-6_1.
 - [40] T. Domonkos and P. Baranyi. Comprehensive analysis of a new fuzzy rule interpolation method. *Fuzzy Systems, IEEE Transactions on*, 8(3):281–296, 2000. doi: 10.1109/91.855917.

- [41] F.C. Erzen, G. Birol, and Ali Cinar. Glucosim: a simulator for education on the dynamics of diabetes mellitus. In *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, volume 4, pages 3163–3166 vol.4, 2001. doi: 10.1109/IEMBS.2001.1019493.
- [42] G. Freckmann, S. Pleus, M. Link, E. Zschornack, H.-M. Kltzer, and C. Haug. Performance evaluation of three continuous glucose monitoring systems: Comparison of six sensors per subject in parallel. *Journal of Diabetes Science and Technology*, 7(4):842–853, 2013.
- [43] H. Goh, J.-H. Lim, and C. Quek. Fuzzy associative conjuncted maps network. *Neural Networks, IEEE Transactions on*, 20(8):1302–1319, 2009. doi: 10.1109/TNN.2009.2023213.
- [44] L.K. Hansen and P. Salamon. Neural network ensembles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(10):993–1001, 1990. doi: 10.1109/34.58871.
- [45] L.K. Hansen, C. Liisberg, and P. Salamon. Ensemble methods for handwritten digit recognition. In *Neural Networks for Signal Processing [1992] II., Proceedings of the 1992 IEEE-SP Workshop*, pages 333–342, 1992. doi: 10.1109/NNSP.1992.253679.
- [46] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.
- [47] P. Herrero, P. Georgiou, N. Oliver, D. G Johnston, and C. Toumazou. A bio-inspired glucose controller based on pancreatic beta-cell physiology. *Journal of Diabetes Science and Technology*, 6(3):606–616, 2012.
- [48] W.-H. Hsiao, S.-M. Chen, and C.-H. Lee. A new interpolative reasoning method in sparse rule-based systems. *Fuzzy Sets and Systems*, 93(1):17 – 22, 1998. doi: 10.1016/S0165-0114(96)00190-X.

- [49] Z. Huang and Q. Shen. Fuzzy interpolative reasoning via scale and move transformations. *Fuzzy Systems, IEEE Transactions on*, 14(2):340–359, 2006. doi: 10.1109/TFUZZ.2005.859324.
- [50] Z. Huang and Q. Shen. Fuzzy interpolationg and extrapolation: A practical approach. *IEEE Transactions on Fuzzy Systems*, 16(1):13–28, 2008.
- [51] B. J. Jacob, E. Y. Cheu, J. Tan, and C. Quek. Self-reorganizing TSK fuzzy inference system with BCM theory of meta-plasticity. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8, 2012. doi: 10.1109/IJCNN.2012.6252527.
- [52] J. S. R. Jang. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, 1993.
- [53] S. Jenei. Interpolation and extrapolation of fuzzy quantities revisited an axiomatic approach. *Soft Computing*, 5(3):179–193, 2001. doi: 10.1007/s005000100080.
- [54] S. Jenei, E. P. Klement, and R. Konzel. Interpolation and extrapolation of fuzzy quantities the multiple-dimensional case. *Soft Computing*, 6(3-4):258–270, 2002. doi: 10.1007/s005000100152.
- [55] S. Jin, R. Diao, C. Quek, and Q. Shen. Backward fuzzy rule interpolation. *Fuzzy Systems, IEEE Transactions on*, 22(6):1682–1698, 2014. doi: 10.1109/TFUZZ.2014.2303474.
- [56] S. Jin, R. Diao, and Q. Shen. Alpha-cut-based backward fuzzy interpolation. In *Cognitive Informatics Cognitive Computing (ICCI*CC), 2014 IEEE 13th International Conference on*, pages 211–218, 2014. doi: 10.1109/ICCI-CC.2014.6921462.
- [57] Z. Johanyák and S. Kovács. Fuzzy rule interpolation based on polar cuts. In Bernd Reusch, editor, *Computational Intelligence, Theory and Applications*, volume 38, pages 499–511. Springer Berlin Heidelberg, 2006. doi: 10.1007/3-540-34783-6_49.

- [58] C. F. Juang and C. T. Lin. An online self-constructing neural fuzzy inference network and its applications. *IEEE Transaction Fuzzy Systems*, 6:12–32, 1998.
- [59] Hyacinthe Tchewonpi Kankeu, Priyanka Saksena, Ke Xu, and David B Evans. The financial burden from non-communicable diseases in low- and middle-income countries: a literature review. *Health Research Policy and Systems*, 11(31), 2013. doi:10.1186/1478-4505-11-31.
- [60] N. Kasabov. Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 31(6):902–918, 2001. doi: 10.1109/3477.969494.
- [61] N.K. Kasabov and Qun Song. DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *Fuzzy Systems, IEEE Transactions on*, 10(2):144–154, 2002. doi: 10.1109/91.995117.
- [62] L. Kóczy and K. Hirota. Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, 9(3):197 – 225, 1993. doi: 10.1016/0888-613X(93)90010-B.
- [63] L. Kóczy and K. Hirota. Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases. *Information Sciences*, 71(12):169 – 201, 1993. doi: 10.1016/0020-0255(93)90070-3.
- [64] Teuvo Kohonen. Learning vector quantization. In *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*, pages 245–261. Springer Berlin Heidelberg, 2001. ISBN 978-3-540-67921-9. doi: 10.1007/978-3-642-56927-2_6.
- [65] S. Kovács and L. T. Kóczy. Application of the approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rulebase in the fuzzy logic controlled path tracking strategy of differential steered AGVs. In Bernd Reusch, editor, *Computational Intelligence Theory and Applications*, volume 1226 of *Lecture Notes in Computer Science*, pages 456–467. Springer Berlin Heidelberg, 1997. ISBN 978-3-540-62868-2.

- [66] Szilveszter Kovács. Extending the fuzzy rule interpolation FIVE by fuzzy observation.
- [67] T.D. Lalitharatne, A. Yoshino, Y. Hayashi, K. Teramoto, and K. Kiguchi. Toward EEG control of upper limb power-assist exoskeletons: A preliminary study of decoding elbow joint velocities using EEG signals. In *Micro-NanoMechatronics and Human Science (MHS), 2012 International Symposium on*, pages 421–424, 2012. doi: 10.1109/MHS.2012.6492482.
- [68] H. Lee, B. A. Buckingham, D. M. Wilson, and B. W. Bequette. A close-loop artificial pancreas using model predictive control and a sliding meal size estimator. *Journal of Diabetes Science and Technology*, 3(5):1082–1090, 2009.
- [69] G. Leng, G. Prasad, and T. M. McGinnity. An on-line algorithm for creating self-organizing fuzzy neural networks. *Neural Networks*, 17(10):1477 – 1493, 2004. doi: 10.1016/j.neunet.2004.07.009.
- [70] G. Lepore, A. Corsi, A. R. Dodesini, I. Nosari, and R. Trevisan. Continuous subcutaneous insulin infusion is better than multiple daily insulin injections in reducing glucose variability only in type 1 diabetes with good metabolic control. *Diabetes Care*, 33(6):28–47, 2010.
- [71] A. Likas, N. Vlassis, and J. J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451 – 461, 2003. doi: [http://dx.doi.org/10.1016/S0031-3203\(02\)00060-2](http://dx.doi.org/10.1016/S0031-3203(02)00060-2).
- [72] C.-J. Lin and C.-T. Lin. An ART-based fuzzy adaptive learning control network. *Fuzzy Systems, IEEE Transactions on*, 5(4):477–496, 1997. doi: 10.1109/91.649900.
- [73] R. Linkeschova, M. Raoul, U. Bott, M. Berger, and M. Spraul.
- [74] F Liu, C Quek, and G Ng. A novel generic Hebbian ordering-based fuzzy rule base reduction approach to Mamdani neuro-fuzzy system. *Neural Computation*, 19(6):1656–1680, 2007. doi: 10.1162/neco.2007.19.6.1656.

- [75] Chi-Huang Lu. Wavelet fuzzy neural networks for identification and predictive control of dynamic systems. *Industrial Electronics, IEEE Transactions on*, 58(7):3046–3058, 2011. doi: 10.1109/TIE.2010.2076415.
- [76] E. D. Lughofer. FLEXFIS: A Robust Incremental Learning Approach for Evolving Takagi Sugeno Fuzzy Models. *Fuzzy Systems, IEEE Transactions on*, 16(6):1393–1410, 2008. doi: 10.1109/TFUZZ.2008.925908.
- [77] E. D. Lughofer and P. Angelov. Handling drifts and shifts in on-line data streams with evolving fuzzy systems. *Applied Soft Computing*, 11(2):2057 – 2068, 2011. doi: 10.1016/j.asoc.2010.07.003.
- [78] M. C. Mackey and L. Glass. Oscillation and chaos in physiological control systems. In *Science*, volume 197, pages 287–289, 1977.
- [79] K. Malik, E. Jespersen, and M. Kugler. *Human Development Report 2014*. PBM Graphics, an RR Donnelley Company, 2014. ISBN 978-92-1-126368-8.
- [80] Ebrahim H. Mamdani. Application of fuzzy logic to approximate reasoning using linguistic synthesis. *Computers, IEEE Transactions on*, C-26(12):1182–1191, 1977.
- [81] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.
- [82] J. M. Mendel, R. I. John, and F. Liu. Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems*, 14(6):808–821, 2006.
- [83] Jerry M. Mendel. *Uncertain rule-based fuzzy logic system: Introduction and new direction*.
- [84] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In *In Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 1041–1045, 1986.

- [85] L. Naderloo, H. Javadikia, and M. Mostafaei. Modeling the energy ratio and productivity of biodiesel with different reactor dimensions and ultrasonic power using ANFIS. *Renewable and Sustainable Energy Reviews*, 70:56–64, 2017.
- [86] N. Naik, R. Diao, and Q. Shen. Genetic algorithm-aided dynamic fuzzy rule interpolation. In *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 2198–2205, July 2014. doi: 10.1109/FUZZ-IEEE.2014.6891816.
- [87] N. N. Nguyen and C. Quek. Stock price prediction using generic self-evolving Takagi-Sugeno-Kang (GSETSK) fuzzy neural network. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8, 2010. doi: 10.1109/IJCNN.2010.5596348.
- [88] N. N. Nguyen, C. Quek, and E. Y. Cheu. Traffic prediction using a generic self-evolving Takagi-Sugeno-Kang (GSETSK) fuzzy neural network. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2012. doi: 10.1109/IJCNN.2012.6252409.
- [89] N. N. Nguyen, W. J. Zhou, and C. Quek. GSETSK: a generic self-evolving {TSK} fuzzy neural network with a novel Hebbian-based rule reduction approach. *Applied Soft Computing*, 35:29 – 42, 2015. doi: 10.1016/j.asoc.2015.06.008.
- [90] R. J. Oentaryo, M. J. Er, S. Linn, and X. Li. Online probabilistic learning for fuzzy inference system. *Expert Systems with Applications*, 41(11): 5082 – 5096, 2014. ISSN 0957-4174. doi: <http://dx.doi.org/10.1016/j.eswa.2014.01.034>. URL <http://www.sciencedirect.com/science/article/pii/S095741741400058X>.
- [91] K. Osaka, T. Tanioka, R.C. Locsin, S.-i. Ueno, K. Matsumoto, C. Kawanishi, S. Kuroiwa, S. Tsuchiya, and F. Ren. Electroencephalograph estimation method of measuring ”empathic understanding”. In *Natural Language Processing and Knowledge Engineering, 2007. NLP-KE 2007. International Conference on*, pages 514–519, 2007. doi: 10.1109/NLPKE.2007.4368080.

- [92] R. S. Parker, F. J. Doyle III, and N. A. Peppas. The intravenous route to insulin-dependent diabetes therapy. *IEEE Engineering Medicine and Biology Magazine*, 20(1):65–73, 2001.
- [93] A. Perez Garcia, I. Schjlborg, and S. Gale. EEG control of an industrial robot manipulator. In *Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on*, pages 39–44, 2013. doi: 10.1109/CogInfoCom.2013.6719280.
- [94] H. K. Phee, W. L. Tung, and C. Quek. A personalized approach to insulin regulation using brain-inspired neural semantic memory in diabetic glucose control. In *2007 IEEE Congress on Evolutionary Computation*, pages 2644–2651, Sept 2007. doi: 10.1109/CEC.2007.4424804.
- [95] M. Pratama, S. G. Anavatti, and E. Lughofer. Genefis: Toward an effective localist network. *IEEE Transactions on Fuzzy Systems*, 22(3):547–562, June 2014. ISSN 1063-6706. doi: 10.1109/TFUZZ.2013.2264938.
- [96] K. H. Quah and C. Quek. FITSK: online local learning with generic fuzzy input Takagi-Sugeno-Kang fuzzy framework for nonlinear system estimation. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36(1):166–178, 2006. doi: 10.1109/TSMCB.2005.856715.
- [97] J Ross Quinlan. Combining instance-based and model-based learning. In *ICML*, page 236, 1993.
- [98] H.-J. Rong, N. Sundararajan, G.-B. Huang, and P. Saratchandran. Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction. *Fuzzy Sets and Systems*, 157(9):1260 – 1275, 2006. doi: 10.1016/j.fss.2005.12.011.
- [99] Jeffrey C. Schlimmer and Richard H. Granger Jr. Incremental learning from noisy data. *Machine Learning*, 1:317–354, 1986.

- [100] Y. Shi and M. Mizumoto. Reasoning conditions on Kóczy's interpolative reasoning method in sparse fuzzy rule bases. Part {II} . *Fuzzy Sets and Systems*, 87(1):47 – 56, 1997. doi: 10.1016/S0165-0114(96)00031-0.
- [101] H. Song, C. Miao, Z. Shen, Y. Miao, and B.-S. Lee. A fuzzy neural network with fuzzy impact grades. *Neurocomputing*, 72(1315):3098 – 3122, 2009. doi: 10.1016/j.neucom.2009.03.009.
- [102] K. O. Stanley. Learning concept drift with a committee of decision trees. Tech. report utai-tr-03-302, 2003.
- [103] Charles F. Stevens. Inferences about membrane properties from electrical noise measurements. *Biophysical Journal*, 12(8):1028–1047, 1972. doi: 10.1016/S0006-3495(72)86141-1.
- [104] K. Subramanian and S. Suresh. A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system. *Applied Soft Computing*, 12(11): 3603 – 3614, 2012. ISSN 1568-4946. doi: <http://dx.doi.org/10.1016/j.asoc.2012.06.012>. URL <http://www.sciencedirect.com/science/article/pii/S1568494612002888>.
- [105] M. Sugeno and G. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1):15–33, 1988.
- [106] T.J. Sullivan, S.R. Deiss, and G. Cauwenberghs. A low-noise, non-contact EEG/ECG sensor. In *Biomedical Circuits and Systems Conference, 2007. BIO-CAS 2007. IEEE*, pages 154–157, 2007. doi: 10.1109/BIOCAS.2007.4463332.
- [107] T.J. Sullivan, S.R. Deiss, T.-P. Jung, and G. Cauwenberghs. A brain-machine interface using dry-contact, low-noise EEG sensors. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 1986–1989, 2008. doi: 10.1109/ISCAS.2008.4541835.
- [108] A. Sun, B. Fan, and C. Jia. Motor imagery EEG-based online control system for upper artificial limb. In *Transportation, Mechanical, and Electrical Engineering*

- (*TMEE*), *2011 International Conference on*, pages 1646–1649, 2011. doi: 10.1109/TMEE.2011.6199526.
- [109] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *Systems, Man, and Cybernetics, IEEE Transactions on*, 15(1):116–132, 1985.
- [110] J. Tan and Chai Quek. A BCM Theory of Meta-Plasticity for Online Self-Reorganizing Fuzzy-Associative Learning. *Neural Networks, IEEE Transactions on*, 21(6):985–1003, 2010. doi: 10.1109/TNN.2010.2046747.
- [111] S. D. Teddy. *A Novel Neurophysiologically-Inspired Self-Organizing Cerebellar Memory Framework*. PhD thesis, School of Computer Engineering, Nanyang Technological University, Singapore, 2007.
- [112] S. D. Teddy, C. Quek, E. M. K. Lai, and A. Cinar. PSECMAC Intelligent insulin schedule for diabetic blood glucose management under nonmeal announcement. *IEEE Transactions on Neural Networks*, 21(3):361–380, March 2010. ISSN 1045-9227. doi: 10.1109/TNN.2009.2036726.
- [113] C. W. Ting and C. Quek. A Novel Blood Glucose Regulation Using TSK0-FCMAC: A Fuzzy CMAC Based on the Zero-Ordered TSK Fuzzy Inference Scheme. *Neural Networks, IEEE Transactions on*, 20(5):856–871, 2009. doi: 10.1109/TNN.2008.2011735.
- [114] A. Tsanas, M.A. Little, P.E. McSharry, and L.O. Ramig. Accurate tele-monitoring of Parkinson’s disease progression by noninvasive speech tests. *Biomedical Engineering, IEEE Transactions on*, 57(4):884–893, 2010. doi: 10.1109/TBME.2009.2036000.
- [115] S.W. Tung, C. Quek, and C. Guan. eT2FIS: An Evolving Type-2 Neural Fuzzy Inference System. *Information Sciences*, 220:124 – 148, 2013. doi: 10.1016/j.ins.2012.02.031.
- [116] W. L. Tung and C. Quek. eFSM - A Novel Online Neural-Fuzzy Semantic

- Memory Model. *Neural Networks, IEEE Transactions on*, 21(1):136–157, 2010. doi: 10.1109/TNN.2009.2035116.
- [117] W.L. Tung and C. Quek. GenSoFNN: a generic self-organizing fuzzy neural network. *Neural Networks, IEEE Transactions on*, 13(5):1075–1086, 2002. doi: 10.1109/TNN.2002.1031940.
 - [118] W.L. Tung and C. Quek. DIC: A Novel Discrete Incremental Clustering Technique for the Derivation of Fuzzy Membership Functions. In Mitsuru Ishizuka and Abdul Sattar, editors, *PRICAI 2002: Trends in Artificial Intelligence*, volume 2417 of *Lecture Notes in Computer Science*, pages 178–187. Springer Berlin Heidelberg, 2002. doi: 10.1007/3-540-45683-X_21.
 - [119] L. Ughetto, D. Dubois, and H. Prade. Fuzzy interpolation by convex completion of sparse rule bases. In *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*, volume 1, pages 465–470, 2000. doi: 10.1109/FUZZY.2000.838704.
 - [120] Puckett Wanda Rae. *Dynamic modelling of diabetes mellitus*. PhD thesis, The University of Wisconsin - Madison, 1992.
 - [121] J.-S. Wang and C.S.G. Lee. Self-adaptive neuro-fuzzy inference systems for classification applications. *Fuzzy Systems, IEEE Transactions on*, 10(6):790–802, 2002. doi: 10.1109/TFUZZ.2002.805880.
 - [122] J. R. Whitlock, A. J. Heynen, M. G. Shuler, and M. F. Bear. Learning induces long-term potentiation in the hippocampus. *Science*, 313:1093–1097, 2006.
 - [123] W Wolberg and O Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology,. In *Proceedings of the National Academy of Sciences*, pages 9193–9196, 1990.
 - [124] Z. Q. Wu, M. Masaharu, and Y. Shi. An improvement to Kóczy and Hirota’s interpolative reasoning in sparse fuzzy rule bases . *International Journal of Approximate Reasoning*, 15(3):185 – 201, 1996. doi: 10.1016/S0888-613X(96)00054-0.

- [125] L. Yang and Q. Shen. Adaptive fuzzy interpolation. *Fuzzy Systems, IEEE Transactions on*, 19(6):1107–1126, 2011. doi: 10.1109/TFUZZ.2011.2161584.
- [126] L. Yang and Q. Shen. Closed form fuzzy interpolation. *Fuzzy Sets and Systems*, 225:1 – 22, 2013. doi: 10.1016/j.fss.2013.04.001.
- [127] L. Yang, F. Chao, and Q. Shen. Generalised adaptive fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, PP(99):1–15, 2016. ISSN 1063-6706. doi: 10.1109/TFUZZ.2016.2582526.
- [128] Y. Yueng and L. T. Kóczy. Cartesian representation for fuzzy interpolation. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 3, pages 2936–2937, 1998. doi: 10.1109/CDC.1998.757926.
- [129] L. A. Zadeh. The concepts of a linguistic variable and its application to approximate reasoning. *Information Sciences*, 8:199–149, 1975.
- [130] J. Zhang, G. G. Walter, Y. Miao, and N. W. L. Wan. Wavelet neural networks for function learning. *Signal Processing, IEEE Transactions on*, 43(6):1485–1497, 1995. doi: 10.1109/78.388860.
- [131] W. J. Zhou, D. L. Maskell, and C. Quek. Fuzzy interpolation and extrapolation using shift ratio and overall weight measurement based on areas of fuzzy sets. In *Computational Intelligence (UKCI), 2013 13th UK Workshop on*, pages 46–53, Sept 2013. doi: 10.1109/UKCI.2013.6651286.
- [132] W. J. Zhou, D. L. Maskell, and C. Quek. FIE-FCMAC: A novel fuzzy cerebellum model articulation controller (FCMAC) using fuzzy interpolation and extrapolation technique. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2015. doi: 10.1109/IJCNN.2015.7280456.
- [133] M. Zikeba, J. M. Tomczak, M. Lubicz, and J. 'Swikatek. Boosted SVM for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients. *Applied Soft Computing*, 2013.