# Investigation into Rule-based Inferential Modelling and Prediction with Application in Healthcare

**A Thesis Submitted to The University of Manchester for the degree of Doctor of Philosophy in the Faculty of Humanities**

**2018**

**Shuaiyu Yao**

**Alliance Manchester Business School**

# List of Contents

# List of Tables

# List of Figures

# Abstract

Sepsis is a serious disease that can cause death. It is important to evaluate patients' sepsis risk during diagnostic decisions within the early stages after the detection of the presence of symptoms that suggest sepsis. The conventional approach to sepsis diagnosis is blood culture, which may takes several days. The approaches based on statistics and machine learning for sepsis diagnosis can be cheap, fast, and non-invasive. There are a wide variety of approaches based on statistics and machine learning that can be used for sepsis diagnosis, but these approaches have some issues, e.g. interpretability and overfitting, which may affect their performance in sepsis diagnosis.

To address some of the issues in the popular approaches to disease diagnosis, we proposed a new approach, i.e., the rule-based inferential modelling and prediction. This approach integrates statistical analysis, belief rule-base inference, and maximum likelihood prediction, and machine learning. The referential-value-based data discretisation technique used in this approach is closer to reality and better at reducing information loss and distortion, as well as better at presenting the characteristics of the data, compared to other data-processing techniques. We can use the belief rule-base inference to clearly analyse the relationship between system inputs and outputs. An interdependence index is used in this approach to quantify the interdependence between input variables. An adapted genetic algorithm is used in this approach for the bilevel optimisation of models. The stopping criteria for the training process of the models used in this approach help us find the optimal structure of the models, which generally achieves balance between accuracy and complexity.

Compared to the complex classifiers for disease diagnosis, e.g., ensemble, ANN, and random forest, the classifier based on the maximum likelihood evidential reasoning (MAKER) framework established by the rule-based inferential modelling and prediction approach is more interpretable. The performance of the MAKER-based classifiers constructed by this approach for sepsis diagnosis is generally better than the majority of alternative models for sepsis diagnosis, and similar to the performance of ensemble: bagged trees, which is a complex model. The MAKER-based classifier is an outstanding classifier for classical data sets: the Banana data set, Haberman's survival data set, and the Iris data set, and it generally performs better than other interpretable classifiers, e.g., complex tree, logistic regression, and naïve Bayes.

Keywords: Evidential Reasoning, Data Discretization, Statistical Analysis, Probabilistic Inference, Machine Learning, Prediction, Decision Making.

# Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

# Copyright Statement

The following four notes on copyright and the ownership of intellectual property rights must be included as written below:

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.
ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance

with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.
iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=2442 0), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see http://www.library.manchester.ac.uk/about/regulations/) and in The University's policy on Presentation of Theses

# Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisors Prof. Jian-Bo Yang and Prof. Dong-Ling Xu for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

My sincere thanks also goes to Prof. Paul Dark who provided me original data from hospitals and medical knowledge for research. Without his precious support it would not be possible to conduct this research.

I thank my colleagues in the Decision and Cognitive Sciences Research Centre for the stimulating discussions, for the constructive comments, for the friendly criticisms, and for all the fun we have had in the last four years.

Last but not the least, I would like to thank my beloved parents and fiancée for supporting me spiritually throughout writing this thesis and my life in general.

# Chapter 1 Introduction

## 1.1 Background

Sepsis is a type of clinical syndrome associated with infection and inflammation. It generally results from the host's systemic inflammatory response to different external factors, e.g. infection, trauma, etc. Daniels and Nutbeam (2010) argue that Sepsis is a continuum of different terms, e.g. systemic inflammatory response syndrome (SIRS), infection, sepsis, severe sepsis, septic shock, multiple organ dysfunction syndrome (MODS). Table 1.1 provides the definitions for these terms (Levy et al., 2003), which originated from a consensus conference headed by the Society of Critical Care Medicine (SCCM) and the American College of Chest Physicians (ACCP). Figure 1.1 and Figure 1.2 describe the intrinsic relationships between the different sepsis terms.

**Table 1.1 Nomenclature of sepsis**

| Terms | Definitions[1] |
|---|---|
| SIRS | Systemic inflammatory response syndrome is a non-specific term used to describe the inflammatory response triggered by infection, trauma, burns, pancreatitis, etc. The identification of SIRS requires the presence of at least two of the following criteria: temperature <36°C or >38.3°C; heart rate >90/min; respiratory rate >20/min; white cells <4 or >12 × 109/l; acutely altered mental status; hyperglycaemia (glucose >6.6 mmol/l) (unless diabetic). |
| Infection | Infection refers to the presence of or the response to the microorganisms in a sterile body cavity or fluid. |
| Sepsis | Sepsis is defined as the presence of SIRS and a confirmed |

[1] These definitions are obtained from 'CHAPTER 2: Defining the Spectrum of Disease' (Daniels, 2010).

| | |
|---|---|
| | infection. |
| Severe sepsis | Severe sepsis is defined as the presence of sepsis and signs of organ dysfunction. |
| Septic shock | Septic shock refers to the persistent evidence of hypoperfusion in spite of adequate fluid resuscitation. |
| MODS | Multi-organ dysfunction syndrome is defined as the presence of altered organ dysfunction in critically ill patients. |

From Figure 1.1, it can be seen that both infection and SIRS are not specific to sepsis (Daniels and Nutbeam, 2010). Infection can be triggered by bacteria, virus, fungi, parasite, etc., while SIRS may be caused by pancreatitis, trauma, burns, etc. As mentioned in Table 1.1, the prerequisites of sepsis are infection and SIRS. Hence, the intersection of infection and SIRS is sepsis, as shown in Figure 1.1. Similarly, according to the definitions in Table 1.1, severe sepsis is a subset of sepsis and MODS[2] is part of severe sepsis, as shown in Figure 1.1.

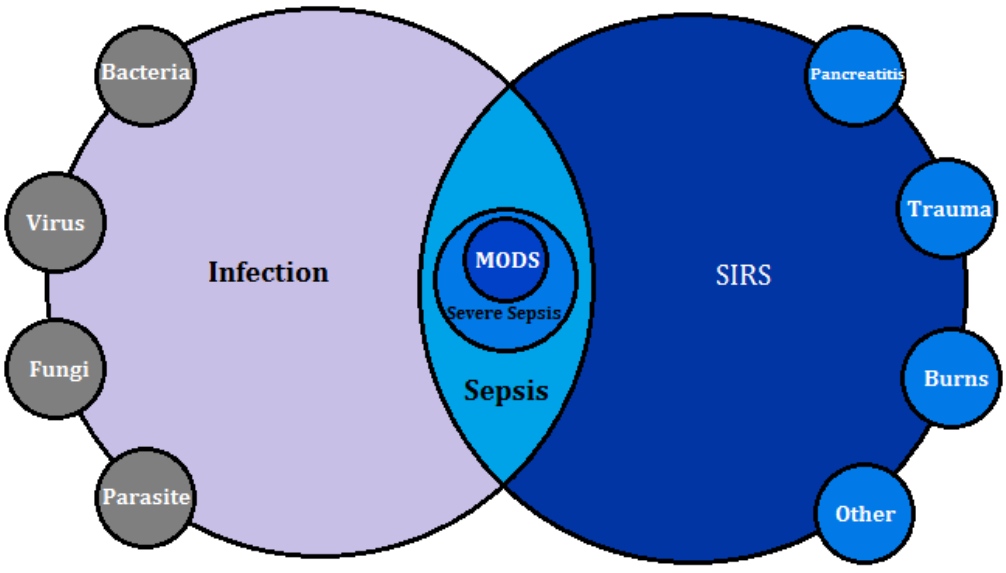



**Figure 1.1 Schematic of the interrelationships between different sepsis terms**

[2] In this context, MODS refers to a specific part of the sepsis continuum.

Figure 1.2 describes the spectrum or the continuum of sepsis in which different sepsis terms, e.g. sepsis, severe sepsis, septic shock, etc., are on a scale of different degrees of characteristics. From left to right in Figure 1.2, the mortalities caused by different sepsis terms are higher and higher (Daniels and Nutbeam, 2010), which is also shown in Figure 1.3 (Levy, 2010).


**Figure 1.2 Continuum of sepsis**

Sepsis is one of the most serious diseases in the world. Each year, approximately 13 million people suffer from sepsis and around 4 million of these die (Levy, 2010). In the US, there are approximately 750,000 cases of sepsis per year, resulting in around 215,000 deaths (Levy, 2010). The financial burden of sepsis on healthcare is significant and it costs the US government approximately 16.7 billion dollars per year (Levy, 2010). In Figure 1.4, it is clear that European incidence of severe sepsis is significantly higher than that of cancers and AIDS. In the UK, it has been estimated that sepsis causes approximately 37,000 deaths per year (Levy, 2010), which is higher than the number of deaths caused by cancers (Figure 1.5).

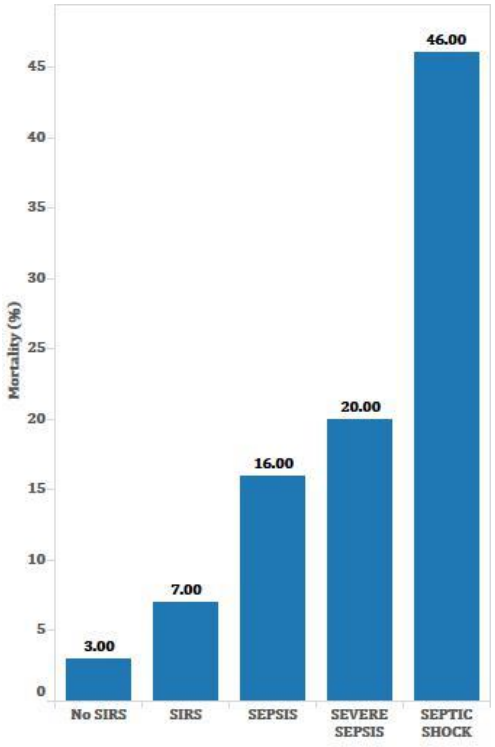



**Figure 1.3 Survival spectrum of sepsis continuum**

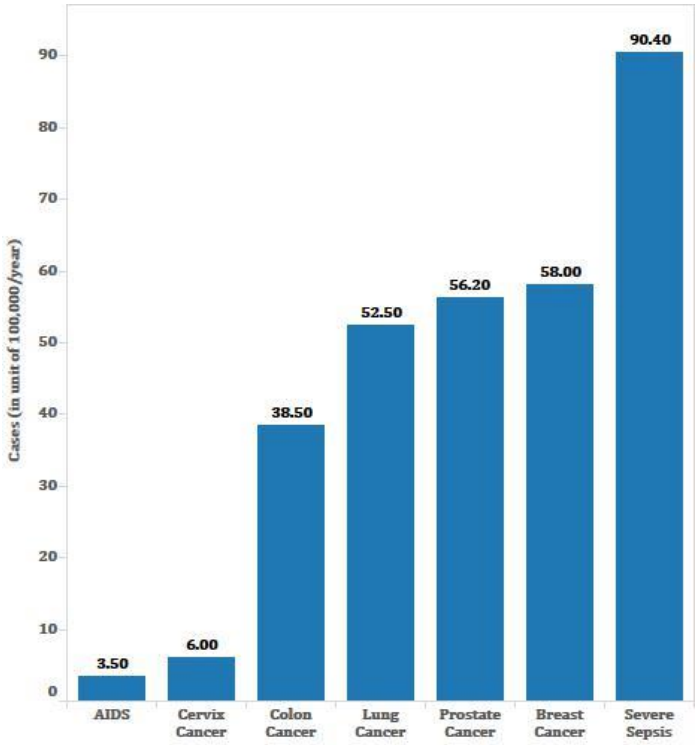



**Figure 1.4 Incidence of severe sepsis and other diseases in Europe**

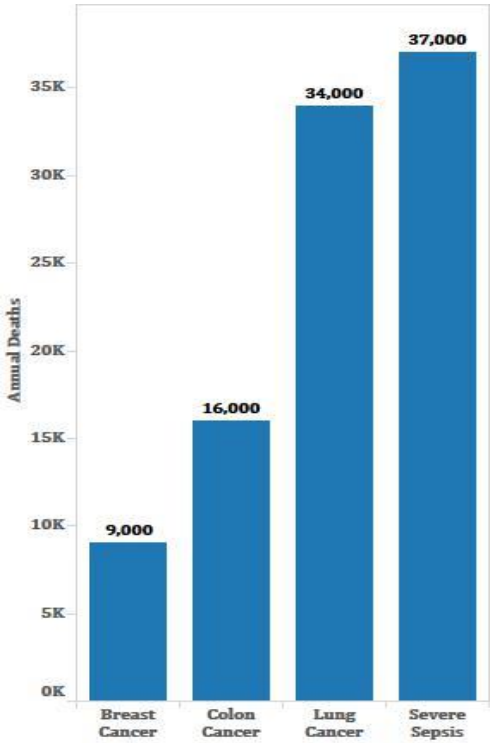



**Figure 1.5 Annual mortality rates for cancers and severe sepsis**

As mentioned in Table 1.1, sepsis is confirmed by the presence of SIRS and evidence of infection. To confirm the presence of infection, common practice is to identify the live pathogens of sepsis from blood samples using culture techniques (Dark et al., 2014).

In general, it takes several days for blood cultures (Warhurst et al., 2014) to determine whether the specimen is culture positive or negative (at least two days are required before a negative result is available). Due to the relatively lengthy process of blood culture and the positive correlation between the delay in antimicrobial therapy and increased mortality, healthcare practitioners generally adopt a 'safety first' clinical strategy to provide early and persistent delivery of potent broad-spectrum antibiotics, which are used against probable pathogens for patients presenting with systemic inflammation (Dark et al., 2014). However, the safety first strategy inevitably results in unreasonable antibiotic prescription, accompanied by a series of adverse effects, e.g. clostridium difficile infection, antimicrobial resistance development, increased acquisition costs, etc., as systemic inflammation is very common in critical care and may be caused by pancreatitis, blood transfusion, trauma, etc., in addition to infection (Warhurst et al., 2014).

Therefore, it is important to develop a decision support system to predict patients' sepsis risk during diagnostic decisions within the early stages after the detection of the presence of symptoms that may suggest sepsis. With the assessment of a patient's sepsis risk, targeted antibiotic therapy can be used in the early stages to effectively improve the patient's prospects of survival.

## 1.2 Research Questions

The research presented in this thesis aims to develop a rule-based inferential modelling and prediction approach for sepsis diagnosis that could be further expanded to analyse and infer complex systems in other domains. A series of

research questions are raised to facilitate the research in this thesis. The research questions comprise two levels: the fundamental level of questions to search for an appropriate research methodology and the advanced level for the selected research methodology.

To find an appropriate research methodology, the author provides a comprehensive analysis of the comparison between the present methodologies of data-driven modelling. The following research questions are designed to guide the analysis:

Q1. What are the most frequently used approaches for disease diagnosis?

Q2. What are the advantages and disadvantages of alternative approaches for disease diagnosis?

Q3. Can we understand the interrelationship between the inputs and outputs of a complex numerical system through the alternative approaches to disease diagnosis?

Q4. How do the alternative approaches perform in disease diagnosis?

Following the analysis of the comparison between the present methodologies for disease diagnosis, the most appropriate approach will be selected for sepsis diagnosis. The following research questions are designed to address this selected approach:

Q5. How can we process continuous input data for a complex numerical system for modelling?

Q6. How can we measure the interdependence between input variables in a complex numerical system?

Q7. How can we perform bilevel optimisation for the models?

Q8. How can we identify the optimal structure of the models?

## 1.3 Research Objectives

To address the research questions formulated in section 1.2, which will facilitate the research, we have designed a number of research objectives. Corresponding with the research questions, the research objectives are classified into two groups: the objectives of identifying the most appropriate approaches for sepsis diagnosis and the objectives of the selected approach for sepsis diagnosis.

The objectives of identifying the most appropriate approach for sepsis diagnosis are listed below.

Obj.1. We will find out the popular approaches to disease diagnosis.

Obj.2. We will compare the weaknesses and strengths of popular approaches to disease diagnosis.

Obj.3. We will understand whether the alternative approaches are interpretable.

Obj.4. We will find out which alternative approaches perform well in the classification of the datasets.

The objectives of the selected approach to sepsis diagnosis are as follows:

Obj.5. We will transform continuous input data for modelling.

Obj.6. We will measure the interdependence between the input variables in a complex numerical system.

Obj.7. We will apply an algorithm for bilevel optimisation.

Obj.8. We will find the optimal structures for the training of the models.

## 1.4 Research Contributions

The main contributions of this research are summarised as follows:

- A referential-value-based data discretisation technique is applied to transform continuous data for modelling. This is one of the innovations in this research. Compared to other data-processing techniques, this technique is closer to reality and better at reducing information loss and distortion, as well as better at presenting the characteristics of the data.

- Stopping criteria are proposed for the training process of the models based on the maximum likelihood evidential reasoning (MAKER) framework, which is another innovation in this research. These stopping criteria help us to find the optimal structure of the models based on the MAKER framework, which generally achieves balance between accuracy and complexity.

- An adapted single-level genetic algorithm is proposed for the problems of bilevel optimization of the MAKER-based models. This is an innovative solution to the bilevel optimization. The function approximation and the classification experiments show that this adapted genetic algorithm can work effectively to find the optimised solutions for the referential values and weights (reliabilities) of the MAKER-based models.

- The statistical analysis, belief rule-base inference, and prediction and machine learning are integrated in the approach of rule-based inferential modelling and prediction, which makes the inference process based on this approach totally transparent and interpretable. It is an innovative approach of modelling and prediction.

- A rule-based inferential modelling and prediction approach is applied to establish the MAKER-based models to identify patients at risk according to the five patient biomarkers in the sepsis dataset. The performance of the MAKER-based models is better than the performance of the alternative models, including complex tree, fine Gaussian support vector machine (SVM), fine k-nearest neighbour (KNN), weighted KNN, ensemble: subspace KNN, naïve Bayes, and artificial neural networks (ANN): feed-forward backpropagation. Among these alternative models, complex tree and naïve Bayes are interpretable. In addition, the performance of the MAKER-based models is similar to the performance of the ensemble: bagged trees, which is a complex model. Compared to the ensemble: bagged trees, the MAKER-based model is totally transparent and interpretable. It is essentially a white-box model in which the relationship between system inputs and outputs can be analysed clearly.

## 1.5. Research Significance

The theoretical and practical significance of this research is summarised in the follow sub-sections.

**Theoretical significance:**

This research contributes to the analysis, modelling, and prediction of complex numerical systems by proposing a rule-based inferential modelling and prediction

approach that provides a new way of effectively addressing some of the issues in the analysis, modelling, and prediction of complex numerical systems. Specifically, the referential-value-based data discretisation technique of this approach reduces the information loss and distortion to which other data discretisation techniques lead. The stopping criteria for the training of the models used in this approach effectively avoid the overfitting of models, which is a common issue in decision tree models. The belief rule-base inference of this approach is totally transparent and interpretable, from which we can analyse the relationship between system inputs and outputs. As a comparison, complex models, e.g. ensembles and artificial neural networks, are generally difficult to understand. The adapted genetic algorithm used in this approach provides an effective way of performing bilevel optimisation for complex numerical systems, while classical algorithms may not work effectively for bilevel optimisation. Overall, the rule-based inferential modelling and prediction approach integrates statistical analysis, belief rule-base inference, and maximum likelihood prediction and machine learning, which enriches the approaches for the analysis, modelling, and prediction of complex numerical systems.

**Practical significance:**

This research will benefit healthcare professionals involved in sepsis diagnosis and diagnosis of other diseases. Healthcare professionals can use the model based on the MAKER framework established by the selected approach: the rule-based inferential modelling and prediction approach to improve the efficiency and accuracy of disease diagnosis.

Specifically, sepsis is a serious disease that may cause death. It is important to evaluate patients' sepsis risk during diagnostic decisions within the early stages after the detection of the presence of symptoms that suggest sepsis. The conventional approach to sepsis diagnosis is blood culture, which may take several days. We can use the rule-based inferential modelling and prediction approach to

establish a MAKER-based model to evaluate patients' sepsis risk in the early stages after the detection of the presence of suspicious symptoms of sepsis, so that targeted antibiotic therapy can be used in the early stages to prevent patients' sepsis from becoming worse and to effectively improve patients' prospects of survival.

As the MAKER-based model is essentially a white-box model from which the relationship between system inputs and outputs can be analysed clearly, healthcare professionals may extract useful patterns from the data on patients' features to provide new perspectives in order to provide timely treatment to patients with suspicious symptoms of a disease to prevent their disease from becoming worse.

## 1.6 Thesis Structure

The remainder of this thesis is structured in six further chapters. These chapters are outlined in Figure 1.6. Each chapter is designed for a specific group of research questions and their relevant research objectives.

Chapter 2 provides a systematic literature review of the popular approaches to disease diagnosis. Based on the systematic literature review, we perform a critical analysis of these popular approaches, which shows that there is a need to develop a new approach to disease diagnosis. Chapter 2 accomplishes research objectives 1 and 2 and addresses research questions 1 and 2, as shown in Figure 1.6. The systematic literature review in Chapter 2 forms a theoretical basis from which to address the research questions formulated in this thesis.

Chapter 3 presents the research methodologies used in this thesis. By achieving Obj.5, Obj.6, and Obj.7, Chapter 3 addresses research questions Q5, Q6, and Q7, as shown in Figure 1.6. Specifically, we introduce the general research methods, data collection, evidential rule, and the research methods for the rule-based

inferential modelling and prediction approach, which is developed to address some of the issues with the popular approaches to disease diagnosis.

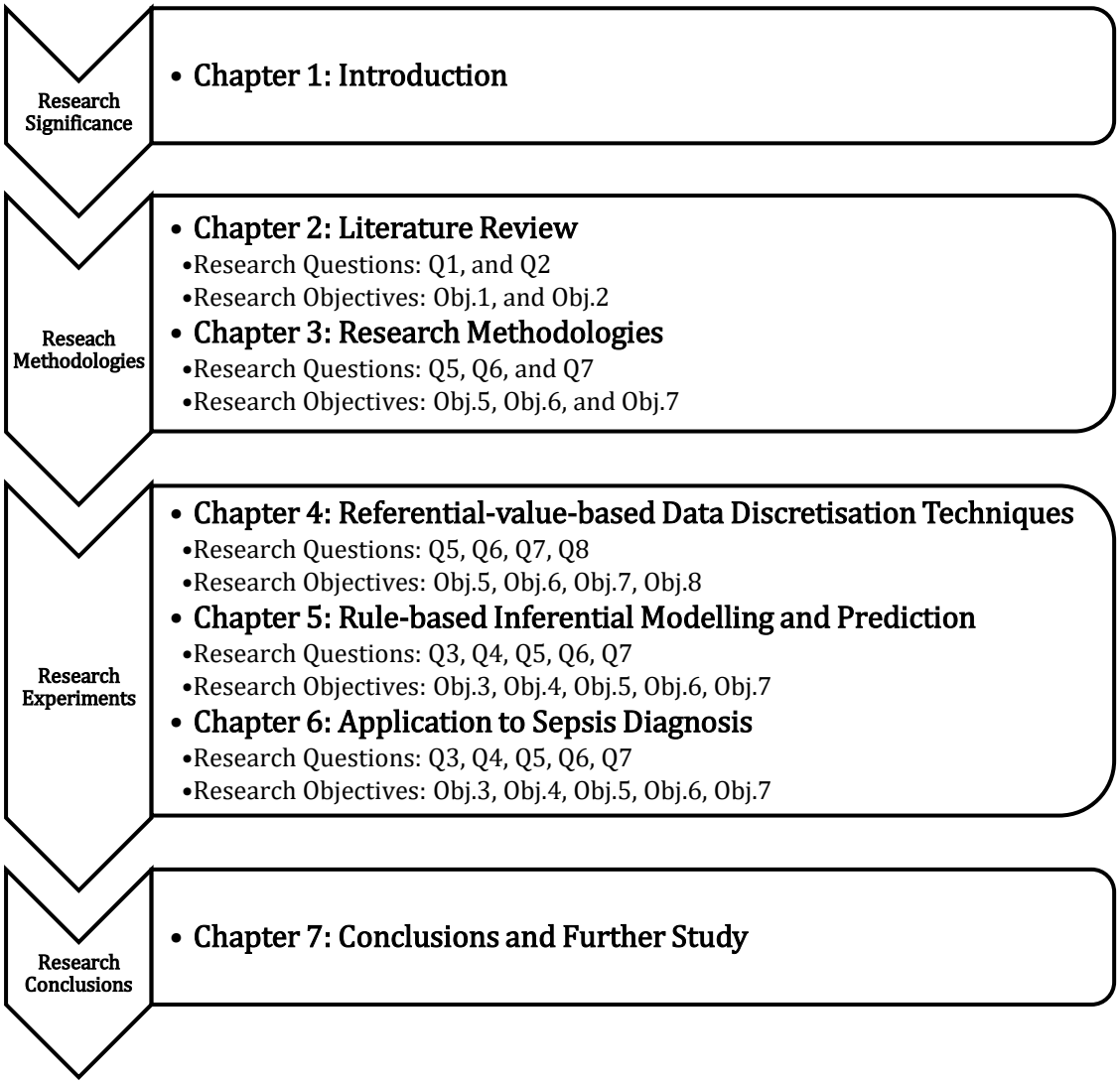



**Figure 1.6 Structure of the thesis**

Chapter 4 is focused on the function approximations. As indicated in Figure 1.6, research questions 5, 6, 7, and 8 are addressed in Chapter 4 by fulfilling research objectives 5, 6, 7, and 8. These function approximations can be divided into univariate function approximations and bivariate function approximation. The univariate function approximations take the monotonic power function, monotonic logarithmic function, monotonic power function, unimodal power function, and bimodal exponential function as examples to validate the capability of the approximation of the rule-based inferential modelling and prediction approach. The

bivariate function approximation takes a benchmark function-Himmelblau function as an example to validate the approximation capability of the rule-based inferential modelling and prediction approach. Based on these approximations, stopping criteria are proposed to find the optimal model structure for the models to achieve balance between accuracy and complexity.

Chapter 5 is dedicated to the rule-based inferential modelling and prediction approach from the perspectives of fundamental knowledge, theoretical comparative analysis, case study, and performance comparative analysis. Research questions 3, 4, 5, 6, and 7 are addressed in Chapter 5 by completing objectives 3, 4, 5, 6, and 7, as shown in Fig 1.6. In this chapter, we first introduce the fundamental knowledge of the rule-based inferential modelling and prediction approach from the perspectives of statistical analysis, belief rule-base inference, and maximum likelihood prediction and machine learning. Then, we perform a comparative analysis to emphasise the limitations of the popular modelling and prediction approaches, and highlight the advantages of the rule-based inferential modelling and prediction approach. Subsequently, we present a case study on how to use the rule-based inferential modelling and prediction approach to build a MAKER-based classifier with which to perform classification experiments on classical datasets, including the Banana dataset, Haberman's survival dataset, and the Iris dataset. Finally, we compare the classification results of the MAKER-based classifiers with those of alternative classifiers.

Chapter 6 focuses mainly on the application of the rule-based inferential modelling and prediction approach to sepsis diagnosis. Figure 1.6 shows that research questions 3, 4, 5, 6, and 7 are addressed in Chapter 6 by fulfilling objectives 3, 4, 5, 6, and 7. In this chapter, we first present the data preparation for the application of the rule-based inferential and modelling approach to sepsis diagnosis. Then, we describe how the classifier based on the system of the MAKER framework is built by the rule-based inferential modelling and prediction approach. Finally, we conduct

a performance comparative analysis between the classification results of the MAKER-based classifier and those of alternative classifiers.

Chapter 7 concludes the findings of this research and suggests directions for further research.

# Chapter 2 Literature Review

## 2.1 Introduction

From the existing literature, it can be seen that a wide variety of approaches have been employed for disease diagnosis. This chapter aims to provide a comprehensive analysis of the popular approaches to disease diagnosis. The remainder of this chapter is organised as follows. Section 2.2 defines disease diagnosis and statistical classification, and the relationship between these two concepts. Section 2.3 defines machine learning, supervised learning, and the relationship between machine learning, supervised learning, and statistical classification. Section 2.4 identifies the popular approaches to disease diagnosis. Section 2.5 critically analyses the identified popular approaches to disease diagnosis.

## 2.2 Disease Diagnosis and Statistical Classification

The aim of disease diagnosis is to identify the disease of a sick patient on the basis of their characteristics (Hand, 1992). Classification means to arrange things into groups of shared characteristics (Cooper and Sartorius, 2013). Therefore, diseases diagnosis is essentially classification. In the context of statistics and machine learning, classification is the identification of the set of existing categories to which a new observation belongs on the basis of training data on observations that have known category memberships (Michie, Spiegelhalter and Taylor, 1994).

Different types of disease, observations of patient characteristics, and characteristics of patients in the context of diagnosis can be considered as classes of output variables, observations, and input variables, respectively, in the context of statistics and machine learning. Based the methods of statistics and machine

learning, we can build models from the data containing the different types of disease and the characteristics of patients for disease diagnosis.

The experience of diagnostic decision was collected and collated subjectively in early days, and objective methods of statistics and machine learning have been applied in diagnosis in recent decades (Hand, 1992). Apparently, disease diagnosis based on methods of statistics and machine learning have some superiority over traditional disease diagnosis. Hand (1992) suggests that traditional disease diagnosis relies on methods that are expensive (e.g. surgical investigations), slow (e.g. bacterial culture), or invasive (bone density measurement in osteoporosis). Compared with traditional disease diagnosis, diagnosis based on methods of statistics and machine learning is cheap, rapid, and non-invasive. With methods of statistics and machine learning, medical diagnostic knowledge can be automatically learned from patient records, and the classifiers built from patient records can help healthcare professionals to improve diagnostic speed, accuracy, and reliability (Kononenko, 2001).

## 2.3 Machine Learning, Supervised Learning, and Statistical Classification

Machine learning is the study of how to build computer programs that improve computer performance through experience in relation to certain tasks (Zhang and Tsai, 2007). It uses algorithms that can be employed to extract patterns from data in order to make inferences or predictions (Alpaydin, 2010). Machine learning techniques are divided into supervised and unsupervised (Zimmermann et al., 2002). Karim and Kaysar (2016) suggest that the tasks in machine learning can be divided into three broad categories: supervised learning, unsupervised learning, and reinforcement learning, which depends on whether or not a learning signal or feedback is available to a learning system. Mohri, Rostamizadeh, and Talwalkar (2012) classify common scenarios of machine learning into supervised learning,

unsupervised learning, semi-supervised learning, reinforcement learning, active learning, etc., on the basis of the types of training data available to the learner, and the order and method by which the training data is received and the test data used to evaluate the learning algorithm. Table 2.1 summarises the different types of machine learning task based on the description presented by Mohri, Rostamizadeh, and Talwalkar (2012).

**Table 2.1 Classification of machine learning tasks**

| Types of machine learning task | Description |
|---|---|
| Supervised learning | The learner receives a set of labelled examples as training data and makes predictions for all unseen points (Mohri, Rostamizadeh, and Talwalkar, 2012). |
| Unsupervised learning | The learner exclusively receives unlabelled training data and makes predictions for all unseen points (Mohri, Rostamizadeh, and Talwalkar, 2012). |
| Semi-supervised learning | The learner receives a training sample consisting of both labelled and unlabelled data and makes predictions for all unseen points (Mohri, Rostamizadeh, and Talwalkar, 2012). |
| Reinforcement learning | To collect information, the learner actively interacts with the environment, and affects the environment in some cases, and receives an immediate reward for each action. The objective of the learner is to maximize their reward throughout a course of actions and iterations with |

| | |
|---|---|
| | the environment (Mohri, Rostamizadeh, and Talwalkar, 2012). |
| Active learning | The learner adaptively or interactively collects training examples, typically by querying a database to request labels for new points. The goal of active learning is to achieve a performance comparable to the standard supervised learning scenario, but with fewer labelled examples (Mohri, Rostamizadeh, and Talwalkar, 2012). |

From the description of supervised learning given by Mohri, Rostamizadeh, and Talwalkar (2012), it can be found that supervised learning is a learning task of deducing a function from labelled training data. Cord and Cunningham (2008) point out that supervised learning entails learning a mapping between input variables and output variables and applying this mapping to predict the outputs for unseen data. Supervised learning is analogous to human learning from past experience to gain new knowledge in order to improve the ability to perform real-world tasks (Liu, 2007). Supervised learning is considered to be one of the most important areas of knowledge discovery (Arikawa and Motoda, 1998) and is one of the most commonly used and successful types of machine learning (Müller and Guido, 2016).

According to Karimi (2014), supervised learning is primarily concerned with classification, interpolation, and prediction. Suthaharan (2016) suggests that supervised learning models can be grouped into predictive models, i.e. regression models and classification models. Müller and Guido (2016) conclude that there are generally two major supervised machine learning problems, i.e. regression and classification. Hackeling (2014) suggests that classification and regression are two of the most common supervised machine learning tasks. As mentioned in section

2.2, disease diagnosis is essentially classification. Therefore, disease diagnosis needs a classification algorithm or a supervised learning algorithm.

## 2.4 Popular Approaches to Disease Diagnosis

To identify popular approaches to diseases diagnosis employed in the existing literatures, it was necessary to conduct a literature search. The search was performed using Web of Science (2017). Only papers published in journals from this source were taken into consideration. Papers from journals below a certain quality standard are not included in Web of Science. Hence, low-quality papers were excluded from consideration in the search. Table 2.2 lists the papers relating to disease diagnosis using machine learning approaches.

**Table 2.2 Papers on disease diagnosis using machine learning approaches**

| No. | Approaches employed in the paper | Author(s) | Title of paper |
|---|---|---|---|
| 1 | Linear Discriminant Analysis, **Support Vector Machine** | Alam, S., G. R. Kwon, et al. (2017) | Alzheimer disease classification using KPCA, LDA, and multi-kernel learning SVM |
| 2 | **Support Vector Machine**, Artificial Neural Network | Alyami, R., J. Alhajjaj, et al. (2017) | Investigating the effect of Correlation based Feature Selection on breast cancer diagnosis using Artificial Neural Network and Support Vector Machines |
| 3 | **Support Vector Machine** | Beheshti, I., H. Demirel, et al. (2017) | Classification of Alzheimer's disease and prediction of mild cognitive impairment- |

| | | | |
|---|---|---|---|
| | | | to-Alzheimer's conversion from structural magnetic resource imaging using feature ranking and a genetic algorithm |
| 4 | **Support Vector Machine** | Chen, H. L., L. F. Hu, et al. (2017) | An Effective Machine Learning Approach for Prognosis of Paraquat Poisoning Patients Using Blood Routine Indexes |
| 5 | **Support Vector Machine,** Naive Bayes, Random Forest, and K-Nearest Neighbours | Chen, Y., Y. Luo, et al. (2017) | Machine-learning-based classification of real-time tissue elastography for hepatic fibrosis in patients with chronic hepatitis B |
| 6 | **Support Vector Machine** | Chen, Y. Y., M. A. Sha, et al. (2017) | Automated detection of pathologic white matter alterations in Alzheimer's disease using combined diffusivity and kurtosis method |
| 7 | **Support Vector Machine**-Quadratic Support Vector Machine | Dakappa, P. H., K. Prasad, et al. (2017) | A Predictive Model to Classify Undifferentiated Fever Cases Based on Twenty-Four-Hour Continuous Tympanic Temperature Recording |
| 8 | Disjunctive Normal Form Rule Based | Deng, C. and M. Perkowski (2017) | A General Data Mining Methodology Based on a |

| | | | |
|---|---|---|---|
| | Method, Decision Trees, Naive Bayes, and **Support Vector Machine** | | Weighted Hierarchical Adaptive Voting Ensemble (WHAVE) Machine Learning Method |
| 9 | **Support Vector Machine** | Ding, J. W. and Q. Huang (2017) | Prediction of MCI to AD Conversion Using Laplace Eigenmaps Learned from FDG and MRI Images of AD Patients and Healthy Controls |
| 10 | **Support Vector Machine** | Drosou, K. and C. Koukouvinos (2017) | Proximal support vector machine techniques on medical prediction outcome |
| 11 | **Support Vector Machine** (SVM)-Linear SVM, Quadratic SVM, Cubic SVM, Medium Gaussian SVM, Decision Tree, and Ensemble Subspace Discriminant machine learning | Ekiz, S. and P. Erdogmus (2017) | Comparative Study of Heart Disease Classification |
| 12 | **Support Vector Machine** | Gatos, I., S. Tsantis, et al. (2017) | A Machine-Learning Algorithm toward Color Analysis for Chronic Liver Disease Classification, Employing Ultrasound Shear Wave Elastography |

| | | | |
|---|---|---|---|
| 13 | **Support Vector Machine** | Guo, H., F. Zhang, et al. (2017) | Machine Learning Classification Combining Multiple Features of A Hyper-Network of fMRI Data in Alzheimer's Disease |
| 14 | **Support Vector Machine** | Hojjati, S. H., A. Ebrahimzadeh, et al. (2017) | Predicting conversion from MCI to AD using resting-state fMRI, graph theoretical approach and SVM |
| 15 | **Support Vector Machine** (SVM)-Linear SVM | Holler, Y., A. C. Bathke, et al. (2017) | Combining SPECT and Quantitative EEG Analysis for the Automated Differential Diagnosis of Disorders with Amnestic Symptoms |
| 16 | **Support Vector Machine** | Iftikhar, S., K. Fatima, et al. (2017) | An evolution based hybrid approach for heart diseases classification and associated risk factors identification |
| 17 | **Support Vector Machine** | Khedher, L., I. A. Illan, et al. (2017) | Independent Component Analysis-Support Vector Machine-Based Computer-Aided Diagnosis System for Alzheimer's with Visual Support |
| 18 | **Support Vector Machine** | Kim, H., H. W. Chun, et al. (2017) | Longitudinal Study-Based Dementia Prediction for Public Health |
| 19 | **Support Vector** | Lee, D., Y. S. Kim, | A feasibility study for |

| | | | |
|---|---|---|---|
| | **Machine** | et al. (2017) | automatic lung nodule detection in chest digital tomosynthesis with machine learning based on support vector machine |
| 20 | **Support Vector Machine** | Maryam, N. A. Setiawan, et al. (2017) | A Hybrid Feature Selection Method Using Multiclass SVM for Diagnosis of Erythemato-Squamous Disease |
| 21 | **Support Vector Machine** | Meyer, S., K. Mueller, et al. (2017) | Predicting behavioral variant frontotemporal dementia with pattern classification in multi-center structural MRI data |
| 22 | **Support Vector Machine** and Artificial Neural Network | Morabito, F. C., M. Campolo, et al. (2017) | Deep Learning Representation from Electroencephalography of Early-Stage Creutzfeldt-Jakob Disease and Features for Differentiation from Rapidly Progressive Dementia |
| 23 | **Support Vector Machine** (SVM)-Linear SVM | Ogata, Y., A. Ozaki, et al. (2017) | Interhemispheric Resting-State Functional Connectivity Predicts Severity of Idiopathic Normal Pressure Hydrocephalus |
| 24 | **Support Vector** | Orimaye, S. O., J. | Predicting probable |

| | | | |
|---|---|---|---|
| | **Machine** | S. M. Wong, et al. (2017) | Alzheimer's disease using linguistic deficits and biomarkers |
| 25 | **Support Vector Machine** | Polat, H., H. D. Mehr, et al. (2017) | Diagnosis of Chronic Kidney Disease Based on Support Vector Machine by Feature Selection Methods |
| 26 | **Support Vector Machine** (SVM)-Linear SVM | Segovia, F., J. M. Gorriz, et al. (2017) | Multivariate Analysis of F-18-DMFP PET Data to Assist the Diagnosis of Parkinsonism |
| 27 | **Support Vector Machine**, Decision Trees, and Artificial Neural Network | Shrivastava, V. K., N. D. Londhe, et al. (2017) | A novel and robust Bayesian approach for segmentation of psoriasis lesions and its risk stratification |
| 28 | **Support Vector Machine** | Tan, L. R., X. Y. Guo, et al. (2017) | A Computational Model for the Automatic Diagnosis of Attention Deficit Hyperactivity Disorder Based on Functional Brain Volume |
| 29 | **Support Vector Machine**, and Logistic Regression | Tanaka, H., H. Adachi, et al. (2017) | Detecting Dementia Through Interactive Computer Avatars |
| 30 | **Support Vector Machine** | Wen, H. W., Y. Liu, et al. (2017) | Multi-threshold White Matter Structural Networks Fusion for Accurate Diagnosis of Tourette Syndrome Children |
| 31 | **Support Vector Machine** | Yahiaoui, A., O. Er, et al. (2017) | A new method of automatic recognition for tuberculosis |

| | | | |
|---|---|---|---|
| | | | disease diagnosis using support vector machines |
| 32 | Random Forest, K-Nearest Neighbours, and **Support Vector Machine** | Zhou, W. C., Y. Y. Ma, et al. (2017) | Predictive model for inflammation grades of chronic hepatitis B: Large-scale analysis of clinical parameters and gene expressions |
| 33 | Random Forest, and **Support Vector Machine** | Lin, T., T. B. Liu, et al. (2017) | Serum levels of chemical elements in esophageal squamous cell carcinoma in Anyang, China: a case-control study based on machine learning methods |
| 34 | Ensemble Classifier of Bagged Decision Tree, **Support Vector Machine**, Decision Trees, and Artificial Neural Network-Multilayer Perceptron | Mohebian, M. R., H. R. Marateb, et al. (2017) | A Hybrid Computer-aided-diagnosis System for Prediction of Breast Cancer Recurrence (HPBCR) Using Optimized Ensemble Learning |
| 35 | **Support Vector Machine**, and Artificial Neural Network- Multilayer Perceptron | Nalluri, M. R., K. Kannan, et al. (2017) | Hybrid Disease Diagnosis Using Multiobjective Optimization with Evolutionary Parameter Optimization |
| 36 | **Support Vector Machine** | Aboudi, N. L. and L. Benhlima (2016) | A New Approach Based on PCA and CE-SVM for Hepatitis Diagnosis |

| | | | |
|---|---|---|---|
| 37 | **Support Vector Machine** and Linear Discriminant Analysis | Argerich, S., S. Herrera, et al. (2016) | Evaluation of Periodic Breathing in Respiratory Flow Signal of Elderly Patients using SVM and Linear Discriminant Analysis |
| 38 | **Support Vector Machine**, Decision Trees, Naive Bayes, K-Nearest Neighbours | Asri, H., H. Mousannif, et al. (2016) | Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis |
| 39 | **Support Vector Machine**, and Artificial Neural Network- Multilayer Perceptron, Radial Basis Function | Azzawi, H., J. Y. Hou, et al. (2016) | Lung cancer prediction from microarray data by gene expression programming |
| 40 | **Support Vector Machine**, Random Forest, and Bayesian Network | Bazazeh, D. and R. Shubair (2016) | Comparative Study of Machine Learning Algorithms for Breast Cancer Detection and Diagnosis |
| 41 | Naive Bayes, K-Nearest Neighbours, and **Support Vector Machine** | Begum, S., D. Chakraborty, et al. (2016) | Identifying cancer biomarkers from leukemia data using feature selection and supervised learning |
| 42 | **Support Vector Machine**, Artificial Neural Network, Naive Bayes, and Logistic Regression | Berikol, G. B., O. Yildiz, et al. (2016) | Diagnosis of Acute Coronary Syndrome with a Support Vector Machine |
| 43 | **Support Vector** | Bokov, P., B. | Wheezing recognition |

| | | | |
|---|---|---|---|
| | **Machine** | Mahut, et al. (2016) | algorithm using recordings of respiratory sounds at the mouth in a pediatric population |
| 44 | **Support Vector Machine** | Caicedo-Torres, W., A. Paternina, et al. (2016) | Machine Learning Models for Early Dengue Severity Prediction |
| 45 | Decision Tree and **Support Vector Machine** | Karimi-Alavijeh, F., S. Jalili, et al. (2016) | Predicting metabolic syndrome using decision tree and support vector machine methods |
| 46 | Logistic Regression, **Support Vector Machines**, Decision Trees and Naive Bayes and Their Ensemble | Kate, R. J., R. M. Perez, et al. (2016) | Prediction and detection models for acute kidney injury in hospitalized older adults |
| 47 | An Ensemble of Artificial Neural Network- Multilayer Perceptron, K-Nearest Neighbours and **Support Vector Machine** | Lahijanian, B., F. V. Farahani, et al. (2016) | A New Multiple Classifier System for Diagnosis of Erythemato-Squamous Diseases Based on Rough Set Feature Selection |
| 48 | **Support Vector Machine** | Liu, C., Y. Huang, et al. (2016) | Hierarchical Feature Extraction for Nuclear Morphometry-Based Cancer Diagnosis |
| 49 | K-Nearest | Cardenas-Pena, | Enhanced Data |

| | | | |
|---|---|---|---|
| | Neighbours, **Support Vector Machine**, and Artificial Neural Networks | D., D. Collazos-Huertas, et al. (2017) | Representation by Kernel Metric Learning for Dementia Diagnosis |
| 50 | **Support Vector Machine** | Orimaye, S. O., J. S. M. Wong, et al. (2017) | Predicting probable Alzheimer's disease using linguistic deficits and biomarkers |
| 51 | Ensemble Classifiers of **Support Vector Machine**, Decision Tree-C5.0, and Naive Bayes | Helal, M. E., M. Elmogy, et al. (2017) | Hybrid Rough Set and Heterogeneous Ensemble Classifiers Model for Cancer Classification |
| 52 | Ensemble Classifiers of Random Forest, **Support Vector Machine**, and Artificial Neural Network- Extreme Learning Machine | Li, Y. M., L. Y. Yang, et al. (2017) | Classification of Parkinson's Disease by Decision Tree Based Instance Selection and Ensemble Learning Algorithms |
| 53 | Artificial Neural Network-Backpropagation Neural Network, and **Support Vector Machine** | Singh, B. K., K. Verma, et al. (2017) | Risk stratification of 2D ultrasound-based breast lesions using hybrid feature selection in machine learning paradigm |
| 54 | **Artificial Neural Network** | Alshamrani, B. S. and A. H. Osman (2017) | Investigation of Hepatitis Disease Diagnosis using Different Types of Neural |

| | | | |
|---|---|---|---|
| | | | Network Algorithms |
| 55 | **Artificial Neural Network** | Arabasadi, Z., R. Alizadehsani, et al. (2017) | Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm |
| 56 | **Artificial Neural Network** | Arabasadi, Z., R. Alizadehsani, et al. (2017) | Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm |
| 57 | **Artificial Neural Network**-Radial Basis Function Network, Adaptive Boosting (Adaboost) and Additive Logistic Regression | Aydin, F. and Z. Aslan (2017) | Diagnosis of neuro degenerative diseases using machine learning methods and wavelet transform |
| 58 | **Artificial Neural Network** | Azmi, M. H., M. I. Saripan, et al. (2017) | F-18-FDG PET brain images as features for Alzheimer classification |
| 59 | **Artificial Neural Network**-Convolutional Neural Network | Bernal, J., N. Tajkbaksh, et al. (2017) | Comparative Validation of Polyp Detection Methods in Video Colonoscopy: Results From the MICCAI 2015 Endoscopic Vision Challenge |
| 60 | **Artificial Neural Network**-Backpropagation | Bi, S. S., Q. W. Wang, et al. (2017) | Automatic Monolayer Identification Based on Genetic Neural Network |

| | | | |
|---|---|---|---|
| | Neural Network | | |
| 61 | **Artificial Neural Network**-Convolutional Neural Networks, and Random Forests | Burlina, P., S. Billings, et al. (2017) | Automated diagnosis of myositis from muscle ultrasound: Exploring the use of machine learning and deep learning methods |
| 62 | **Artificial Neural Network** | Choi, E., A. Schuetz, et al. (2017) | Using recurrent neural network models for early detection of heart failure onset |
| 63 | **Artificial Neural Network** | Choi, J. Y., T. K. Yoo, et al. (2017) | Multi-categorical deep learning neural network to classify retinal images: A pilot study employing small database |
| 64 | **Artificial Neural Network**-Extreme Learning Machine | Cui, G. Q., L. B. Xia, et al. (2017) | Automatic Classification of Epileptic Electroencephalogram Based on Multiscale Entropy and Extreme Learning Machine |
| 65 | **Artificial Neural Network**-Multilayer Perceptron | Iliou, T., C. N. Anagnostopoulos, et al. (2017) | A novel data preprocessing method for boosting neural network performance: A case study in osteoporosis prediction |
| 66 | **Artificial Neural Network**-Extreme Learning Machine | Kuppili, V., M. Biswas, et al. (2017) | Extreme Learning Machine Framework for Risk Stratification of Fatty Liver Disease Using Ultrasound |

| | | | |
|---|---|---|---|
| | | | Tissue Characterization |
| 67 | **Artificial Neural Network**-Extreme Learning Machine | Lama, R. K., J. Gwak, et al. (2017) | Diagnosis of Alzheimer's Disease Based on Structural MRI Images Using a Regularized Extreme Learning Machine and PCA Features |
| 68 | **Artificial Neural Network**-Extreme Learning Machine | Lu, S. Y., X. Qiu, et al. (2017) | A Pathological Brain Detection System based on Extreme Learning Machine Optimized by Bat Algorithm |
| 69 | **Artificial Neural Network** | Mamuda, M. and S. Sathasivam (2017) | The development of Adaptive Neuro-Fuzzy Inference System model to diagnosis diabetes disease data set |
| 70 | **Artificial Neural Network**-Extreme Learning Machine | Wang, Y., A. N. Wang, et al. (2017) | An adaptive kernel-based weighted extreme learning machine approach for effective detection of Parkinson's disease |
| 71 | **Artificial Neural Network**-Backpropagation Neural Network | Zaman, S. and R. Toufiq (2017) | Codon Based Back Propagation Neural Network Approach to Classify Hypertension Gene Sequences |
| 72 | **Artificial Neural Network**-Extreme Learning Machine | Avci, D. (2016) | An Automatic Diagnosis System for Hepatitis Diseases Based on Genetic |

| | | | |
|---|---|---|---|
| | | | Wavelet Kernel Extreme Learning Machine |
| 73 | **Artificial Neural Network**-Probabilistic, Multilayer Perceptron, Radial Basis Function, and Alternating Decision Tree (ADTree) | Das, D. K., C. Chakraborty, et al. (2016) | Automated Screening Methodology for Asthma Diagnosis that Ensembles Clinical and Spirometric Information |
| 74 | **Artificial Neural Network**-Multilayer Perceptron, and Cascade-forward Back Propagation Network | El-Baz, A. H., A. E. Hassanien, et al. (2016) | Identification of Diabetes Disease Using Committees of Neural Network-Based Classifiers |
| 75 | **Artificial Neural Network**-Multilayer Perceptron | Kumari, V. S. R. and P. R. Kumar (2016) | Classification of cardiac arrhythmia using hybrid genetic algorithm optimisation for multi-layer perceptron neural network |
| 76 | **Artificial Neural Networks**-Feedforward Neural Network that Uses Backpropagation Learning Algorithm, and Radial Basis Function Networks | Helwan, A., D. U. Ozsahin, et al. (2017) | One-Year Survival Prediction of Myocardial Infarction |

| | | | |
|---|---|---|---|
| 77 | **Artificial Neural Network**-Backpropagation Neural Network, and Support Vector Machine | Singh, B. K., K. Verma, et al. (2017) | Risk stratification of 2D ultrasound-based breast lesions using hybrid feature selection in machine learning paradigm |
| 78 | **Artificial Neural Network**-Backpropagation Neural Network | Sudha, M. (2017) | Evolutionary and Neural Computing Based Decision Support System for Disease Diagnosis from Clinical Data Sets in Medical Practice |
| 79 | Support Vector Machine, **Artificial Neural Network** | Alyami, R., J. Alhajjaj, et al. (2017) | Investigating the effect of Correlation based Feature Selection on breast cancer diagnosis using Artificial Neural Network and Support Vector Machines |
| 80 | Support Vector Machine and **Artificial Neural Network** | Morabito, F. C., M. Campolo, et al. (2017) | Deep Learning Representation from Electroencephalography of Early-Stage Creutzfeldt-Jakob Disease and Features for Differentiation from Rapidly Progressive Dementia |
| 81 | Support Vector Machine, Decision Trees, and **Artificial Neural Network** | Shrivastava, V. K., N. D. Londhe, et al. (2017) | A novel and robust Bayesian approach for segmentation of psoriasis lesions and its risk stratification |

| | | | |
|---|---|---|---|
| 82 | Ensemble Classifier of Bagged Decision Tree, Support Vector Machine, Decision Trees, and **Artificial Neural Network**-Multilayer Perceptron | Mohebian, M. R., H. R. Marateb, et al. (2017) | A Hybrid Computer-aided-diagnosis System for Prediction of Breast Cancer Recurrence (HPBCR) Using Optimized Ensemble Learning |
| 83 | Support Vector Machine, and **Artificial Neural Network**-Multilayer Perceptron | Nalluri, M. R., K. Kannan, et al. (2017) | Hybrid Disease Diagnosis Using Multiobjective Optimization with Evolutionary Parameter Optimization |
| 84 | K-Nearest Neighbours, **Artificial Neural Network**-Radial Basis Function Network, Probabilistic Neural Network, **Artificial Neural Networks**-Multilayer Feedforward Neural Network | Ucar, M. K., M. R. Bozkurt, et al. (2017) | Automatic detection of respiratory arrests in OSA patients using PPG and machine learning techniques |
| 85 | Support Vector Machine, and **Artificial Neural Network**-Multilayer Perceptron, Radial Basis Function | Azzawi, H., J. Y. Hou, et al. (2016) | Lung cancer prediction from microarray data by gene expression programming |
| 86 | Support Vector | Berikol, G. B., O. | Diagnosis of Acute Coronary |

| | | | |
|---|---|---|---|
| | Machine, **Artificial Neural Network**, Naive Bayes, and Logistic Regression | Yildiz, et al. (2016) | Syndrome with a Support Vector Machine |
| 87 | Naive Bayes, Decision Tree, **Artificial Neural Network**-Multilayer Perceptron, Radial Basis Function | Ionita, I. and L. Ionita (2016) | Prediction of Thyroid Disease Using Data Mining Techniques |
| 88 | An Ensemble of **Artificial Neural Network**-Multilayer Perceptron, K-Nearest Neighbours and Support Vector Machine | Lahijanian, B., F. V. Farahani, et al. (2016) | A New Multiple Classifier System for Diagnosis of Erythemato-Squamous Diseases Based on Rough Set Feature Selection |
| 89 | K-Nearest Neighbours, Support Vector Machine, and **Artificial Neural Networks** | Cardenas-Pena, D., D. Collazos-Huertas, et al. (2017) | Enhanced Data Representation by Kernel Metric Learning for Dementia Diagnosis |
| 90 | Ensemble Classifiers of Random Forest, Support Vector Machine, and **Artificial Neural Network**- Extreme Learning Machine | Li, Y. M., L. Y. Yang, et al. (2017) | Classification of Parkinson's Disease by Decision Tree Based Instance Selection and Ensemble Learning Algorithms |
| 91 | K-Nearest | Ucar, M. K., M. R. | Automatic detection of |

| | | | |
|---|---|---|---|
| | Neighbours, **Artificial Neural Networks**-Multilayer Feedforward Neural Network, Radial Basis Function Neural Network, Probabilistic Neural Network, and ensemble classification method | Bozkurt, et al. (2017) | respiratory arrests in OSA patients using PPG and machine learning techniques |
| 92 | Disjunctive Normal Form Rule Based Method, **Decision Trees**, Naive Bayes, and Support Vector Machine | Deng, C. and M. Perkowski (2017) | A General Data Mining Methodology Based on a Weighted Hierarchical Adaptive Voting Ensemble (WHAVE) Machine Learning Method |
| 93 | Support Vector Machine (SVM)-Linear SVM, Quadratic SVM, Cubic SVM, Medium Gaussian SVM, **Decision Tree**, and Ensemble Subspace Discriminant machine learning | Ekiz, S. and P. Erdogmus (2017) | Comparative Study of Heart Disease Classification |
| 94 | Support Vector Machine, **Decision Trees**, and Artificial Neural Network | Shrivastava, V. K., N. D. Londhe, et al. (2017) | A novel and robust Bayesian approach for segmentation of psoriasis lesions and its risk stratification |

| | | | |
|---|---|---|---|
| 95 | Ensemble Classifier of Bagged Decision Tree, Support Vector Machine, **Decision Trees**, and Artificial Neural Network-Multilayer Perceptron | Mohebian, M. R., H. R. Marateb, et al. (2017) | A Hybrid Computer-aided-diagnosis System for Prediction of Breast Cancer Recurrence (HPBCR) Using Optimized Ensemble Learning |
| 96 | Support Vector Machine, **Decision Trees**, Naive Bayes, K-Nearest Neighbours | Asri, H., H. Mousannif, et al. (2016) | Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis |
| 97 | **Decision Tree** and Support Vector Machine | Karimi-Alavijeh, F., S. Jalili, et al. (2016) | Predicting metabolic syndrome using decision tree and support vector machine methods |
| 98 | Logistic Regression, Support Vector Machines, **Decision Trees** and Naive Bayes and Their Ensemble | Kate, R. J., R. M. Perez, et al. (2016) | Prediction and detection models for acute kidney injury in hospitalized older adults |
| 99 | Naive Bayes, **Decision Tree**, Artificial Neural Network-Multilayer Perceptron, Radial Basis Function | Ionita, I. and L. Ionita (2016) | Prediction of Thyroid Disease Using Data Mining Techniques |
| 100 | K-Nearest Neighbours, Random | Amaral, J. L. M., A. J. Lopes, et al. | High-accuracy detection of airway obstruction in asthma |

| | | | |
|---|---|---|---|
| | Forest, AdaBoost with **Decision Trees**, and Feature-based Dissimilarity Space Classifier | (2017) | using machine learning algorithms and forced oscillation measurements |
| 101 | **Decision Tree** and K-Nearest Neighbour | Hashi, E. K., M. S. U. Zaman, et al. (2017) | An Expert Clinical Decision Support System to Predict Disease Using Classification Techniques |
| 102 | Ensemble Classifiers of Support Vector Machine, **Decision Tree**-C5.0, and Naive Bayes | Helal, M. E., M. Elmogy, et al. (2017) | Hybrid Rough Set and Heterogeneous Ensemble Classifiers Model for Cancer Classification |
| 103 | **Decision Trees** | Topalovic, M., S. Laval, et al. (2017) | Automated Interpretation of Pulmonary Function Tests in Adults with Respiratory Complaints |
| 104 | **Decision Trees**, and K-Nearest Neighbours | Hashi, E. K., M. S. U. Zaman, et al. (2017) | An Expert Clinical Decision Support System to Predict Disease Using Classification Techniques |
| 105 | **Decision Tree** | Balasus, D., M. Way, et al. (2016) | The association of variants in PNPLA3 and GRP78 and the risk of developing hepatocellular carcinoma in an Italian population |
| 106 | **Random Forest** | Ardekani, B. A., E. Bermudez, et al. | Prediction of Incipient Alzheimer's Disease |

| | | | |
|---|---|---|---|
| | | (2017) | Dementia in Patients with Mild Cognitive Impairment |
| 107 | **Random Forest** | Balakrishna, T., B. Narendra, et al. (2017) | Diagnosis of Chronic Kidney Disease Using Random Forest Classification Technique |
| 108 | **Random Forest** | Chaganti, S., K. P. Nabar, et al. (2017) | Phenotype Analysis of Early Risk Factors from Electronic Medical Records Improves Image-Derived Diagnostic Classifiers for Optic Nerve Pathology |
| 109 | **Random Forest** | Chiappini, F., A. Coilly, et al. (2017) | Metabolism dysregulation induces a specific lipid signature of nonalcoholic steatohepatitis in patients |
| 110 | **Random Forest** | Chirikov, V. V., F. T. Shaya, et al. (2017) | Tree-based Claims Algorithm for Measuring Pretreatment Quality of Care in Medicare Disabled Hepatitis C Patients |
| 111 | Ensemble Classifiers of **Random Forest**, Support Vector Machine, and Artificial Neural Network-Extreme Learning Machine | Li, Y. M., L. Y. Yang, et al. (2017) | Classification of Parkinson's Disease by Decision Tree Based Instance Selection and Ensemble Learning Algorithms |
| 112 | Support Vector Machine, **Random** | Bazazeh, D. and R. Shubair (2016) | Comparative Study of Machine Learning Algorithms |

| | | | |
|---|---|---|---|
| | **Forest**, and Bayesian Network | | for Breast Cancer Detection and Diagnosis |
| 113 | **Random Forest**, and Support Vector Machine | Lin, T., T. B. Liu, et al. (2017) | Serum levels of chemical elements in esophageal squamous cell carcinoma in Anyang, China: a case-control study based on machine learning methods |
| 114 | **Random Forest**, K-Nearest Neighbours, and Support Vector Machine | Zhou, W. C., Y. Y. Ma, et al. (2017) | Predictive model for inflammation grades of chronic hepatitis B: Large-scale analysis of clinical parameters and gene expressions |
| 115 | Support Vector Machine, Naive Bayes, **Random Forest**, and K-Nearest Neighbours | Chen, Y., Y. Luo, et al. (2017) | Machine-learning-based classification of real-time tissue elastography for hepatic fibrosis in patients with chronic hepatitis B |
| 116 | K-Nearest Neighbours, **Random Forest**, AdaBoost with Decision Trees, and Feature-based Dissimilarity Space Classifier | Amaral, J. L. M., A. J. Lopes, et al. (2017) | High-accuracy detection of airway obstruction in asthma using machine learning algorithms and forced oscillation measurements |
| 117 | **K-Nearest Neighbours**, Support Vector Machine, and | Cardenas-Pena, D., D. Collazos-Huertas, et al. | Enhanced Data Representation by Kernel Metric Learning for |

| | | | |
|---|---|---|---|
| | Artificial Neural Networks | (2017) | Dementia Diagnosis |
| 118 | Support Vector Machine, Naive Bayes, Random Forest, and **K-Nearest Neighbours** | Chen, Y., Y. Luo, et al. (2017) | Machine-learning-based classification of real-time tissue elastography for hepatic fibrosis in patients with chronic hepatitis B |
| 119 | Random Forest, **K-Nearest Neighbours**, and Support Vector Machine | Zhou, W. C., Y. Y. Ma, et al. (2017) | Predictive model for inflammation grades of chronic hepatitis B: Large-scale analysis of clinical parameters and gene expressions |
| 120 | Support Vector Machine, Decision Trees, Naive Bayes, **K-Nearest Neighbours** | Asri, H., H. Mousannif, et al. (2016) | Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis |
| 121 | Naive Bayes, **K-Nearest Neighbours**, and Support Vector Machine | Begum, S., D. Chakraborty, et al. (2016) | Identifying cancer biomarkers from leukemia data using feature selection and supervised learning |
| 122 | An Ensemble of Artificial Neural Network- Multilayer Perceptron, **K-Nearest** | Lahijanian, B., F. V. Farahani, et al. (2016) | A New Multiple Classifier System for Diagnosis of Erythemato-Squamous Diseases Based on Rough Set Feature Selection |

| | | | |
|---|---|---|---|
| | **Neighbours** and Support Vector Machine | | |
| 123 | **K-Nearest Neighbours**, Artificial Neural Network-Radial Basis Function Network, Probabilistic Neural Network, Artificial Neural Networks-Multilayer Feedforward Neural Network | Ucar, M. K., M. R. Bozkurt, et al. (2017) | Automatic detection of respiratory arrests in OSA patients using PPG and machine learning techniques |
| 124 | **K-Nearest Neighbours**, Random Forest, AdaBoost with Decision Trees, and Feature-based Dissimilarity Space Classifier | Amaral, J. L. M., A. J. Lopes, et al. (2017) | High-accuracy detection of airway obstruction in asthma using machine learning algorithms and forced oscillation measurements |
| 125 | Decision Tree and **K-Nearest Neighbour** | Hashi, E. K., M. S. U. Zaman, et al. (2017) | An Expert Clinical Decision Support System to Predict Disease Using Classification Techniques |
| 126 | Decision Trees, and **K-Nearest Neighbours** | Hashi, E. K., M. S. U. Zaman, et al. (2017) | An Expert Clinical Decision Support System to Predict Disease Using Classification |

| | | | |
|---|---|---|---|
| | | | Techniques |
| 127 | **K-Nearest Neighbours**, Artificial Neural Networks- Multilayer Feedforward Neural Network, Radial Basis Function Neural Network, Probabilistic Neural Network, and ensemble classification method | Ucar, M. K., M. R. Bozkurt, et al. (2017) | Automatic detection of respiratory arrests in OSA patients using PPG and machine learning techniques |
| 128 | **K-Nearest Neighbours** | Cuzzolin, F., M. Sapienza, et al. (2017) | Metric learning for Parkinsonian identification from IMU gait measurements |
| 129 | **Ensemble Classifier** of Bagged Decision Tree, Support Vector Machine, Decision Trees, and Artificial Neural Network- Multilayer Perceptron | Mohebian, M. R., H. R. Marateb, et al. (2017) | A Hybrid Computer-aided-diagnosis System for Prediction of Breast Cancer Recurrence (HPBCR) Using Optimized Ensemble Learning |
| 130 | **An Ensemble** of Artificial Neural Network- Multilayer Perceptron, K-Nearest Neighbours and Support Vector | Lahijanian, B., F. V. Farahani, et al. (2016) | A New Multiple Classifier System for Diagnosis of Erythemato-Squamous Diseases Based on Rough Set Feature Selection |

| | | | |
|---|---|---|---|
| | Machine | | |
| 131 | **Ensemble Classifiers** of Support Vector Machine, Decision Tree-C5.0, and Naive Bayes | Helal, M. E., M. Elmogy, et al. (2017) | Hybrid Rough Set and Heterogeneous Ensemble Classifiers Model for Cancer Classification |
| 132 | **Ensemble Classifiers** of Random Forest, Support Vector Machine, and Artificial Neural Network-Extreme Learning Machine | Li, Y. M., L. Y. Yang, et al. (2017) | Classification of Parkinson's Disease by Decision Tree Based Instance Selection and Ensemble Learning Algorithms |
| 133 | Logistic Regression, Support Vector Machines, Decision Trees and Naive Bayes and Their **Ensemble** | Kate, R. J., R. M. Perez, et al. (2016) | Prediction and detection models for acute kidney injury in hospitalized older adults |
| 134 | K-Nearest Neighbours, Artificial Neural Networks-Multilayer Feedforward Neural Network, Radial Basis Function Neural Network, Probabilistic | Ucar, M. K., M. R. Bozkurt, et al. (2017) | Automatic detection of respiratory arrests in OSA patients using PPG and machine learning techniques |

| | | | |
|---|---|---|---|
| | Neural Network, and **ensemble** classification method | | |
| 135 | Support Vector Machine (SVM)-Linear SVM, Quadratic SVM, Cubic SVM, Medium Gaussian SVM, Decision Tree, and **Ensemble** Subspace Discriminant machine learning | Ekiz, S. and P. Erdogmus (2017) | Comparative Study of Heart Disease Classification |
| 136 | Support Vector Machine, **Naive Bayes**, Random Forest, and K-Nearest Neighbours | Chen, Y., Y. Luo, et al. (2017) | Machine-learning-based classification of real-time tissue elastography for hepatic fibrosis in patients with chronic hepatitis B |
| 137 | Disjunctive Normal Form Rule Based Method, Decision Trees, **Naive Bayes**, and Support Vector Machine | Deng, C. and M. Perkowski (2017) | A General Data Mining Methodology Based on a Weighted Hierarchical Adaptive Voting Ensemble (WHAVE) Machine Learning Method |
| 138 | Support Vector Machine, Decision Trees, **Naive Bayes**, K-Nearest Neighbours | Asri, H., H. Mousannif, et al. (2016) | Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis |
| 139 | **Naive Bayes**, K-Nearest Neighbours, | Begum, S., D. Chakraborty, et al. | Identifying cancer biomarkers from leukemia |

| | | | |
|---|---|---|---|
| | and Support Vector Machine | (2016) | data using feature selection and supervised learning |
| 140 | Support Vector Machine, Artificial Neural Network, **Naive Bayes**, and Logistic Regression | Berikol, G. B., O. Yildiz, et al. (2016) | Diagnosis of Acute Coronary Syndrome with a Support Vector Machine |
| 141 | Logistic Regression, Support Vector Machines, Decision Trees and **Naive Bayes** and Their Ensemble | Kate, R. J., R. M. Perez, et al. (2016) | Prediction and detection models for acute kidney injury in hospitalized older adults |
| 142 | **Naive Bayes**, Decision Tree, Artificial Neural Network-Multilayer Perceptron, Radial Basis Function | Ionita, I. and L. Ionita (2016) | Prediction of Thyroid Disease Using Data Mining Techniques |
| 143 | Ensemble Classifiers of Support Vector Machine, Decision Tree-C5.0, and **Naive Bayes** | Helal, M. E., M. Elmogy, et al. (2017) | Hybrid Rough Set and Heterogeneous Ensemble Classifiers Model for Cancer Classification |
| 144 | **Bayesian Network** | Somnay, Y. R., M. Craven, et al. (2017) | Improving diagnostic recognition of primary hyperparathyroidism with machine learning |
| 145 | **Bayesian Network** | Ye, Y., M. M. | A study of the transferability |

| | | | |
|---|---|---|---|
| | | Wagner, et al. (2017) | of influenza case detection systems between two large healthcare systems |
| 146 | Support Vector Machine, Random Forest, and **Bayesian Network** | Bazazeh, D. and R. Shubair (2016) | Comparative Study of Machine Learning Algorithms for Breast Cancer Detection and Diagnosis |
| 147 | **Bayesian Networks** | Guerrero, J. M., R. Martinez-Tomas, et al. (2016) | Diagnosis of Cognitive Impairment Compatible with Early Diagnosis of Alzheimer's Disease |
| 148 | Support Vector Machine, and **Logistic Regression** | Tanaka, H., H. Adachi, et al. (2017) | Detecting Dementia Through Interactive Computer Avatars |
| 149 | Support Vector Machine, Artificial Neural Network, Naive Bayes, and **Logistic Regression** | Berikol, G. B., O. Yildiz, et al. (2016) | Diagnosis of Acute Coronary Syndrome with a Support Vector Machine |
| 150 | **Logistic Regression**, Support Vector Machines, Decision Trees and Naive Bayes and Their Ensemble | Kate, R. J., R. M. Perez, et al. (2016) | Prediction and detection models for acute kidney injury in hospitalized older adults |
| 151 | Artificial Neural Network-Radial Basis Function Network, Adaptive Boosting | Aydin, F. and Z. Aslan (2017) | Diagnosis of neuro degenerative diseases using machine learning methods and wavelet transform |

| | | | |
|---|---|---|---|
| | (Adaboost) and Additive **Logistic Regression** | | |
| 152 | Linear Regression Classification | Chen, Y., Y. Shao, et al. (2017) | A Feature-Free 30-Disease Pathological Brain Detection System by Linear Regression Classifier |

As can be observed in Table 2.2, a wide variety of approaches have been used to support disease diagnosis. We can analyse the popularity of the approaches to disease diagnosis based on the information in Table 2.2. Figure 2.1 presents the popularity of the approaches to disease diagnosis.

From Figure 2.1, we can see that, among the approaches to disease diagnosis, the support vector machine (SVM) is the most popular, followed by artificial neural networks (ANN), decision tree, random forest, k-nearest neighbours, ensemble, naïve Bayes, Bayesian network, and logistic regression.

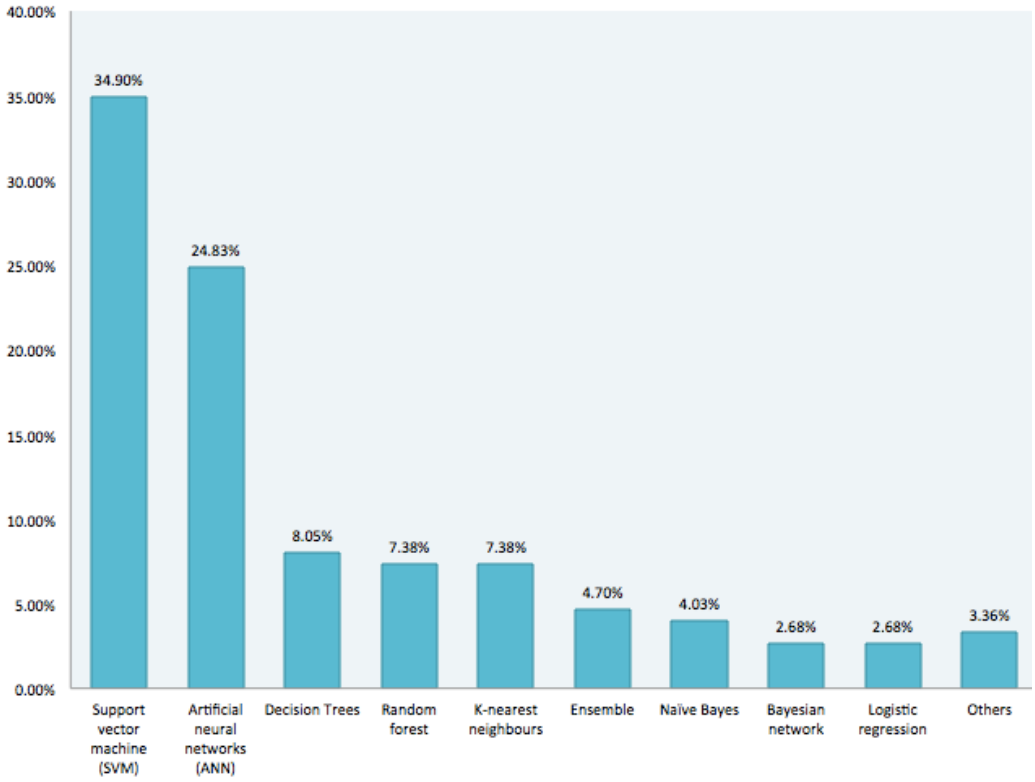



**Figure 2.1 Popularity of approaches to disease diagnosis**

For the sepsis diagnosis, a number of researchers have made attempts to use machine learning approaches. Mani et al (2014) used support vector machine (SVM), naïve Bayes, K-nearest neighbours, decision tree, random forest, logistic regression, and etc., to develop non-invasive predictive models for sepsis from off-the-shelf medical data and electronic medical records. It was found from the research of Gultepe et al (2014) that SVM classification can be used to predict mortality risk for patients with risk when the measurements of patients are summarized by summary statistics. Tang et al (2010) also used the nonlinear SVM in the classification of the sepsis continuum into severe sepsis and systemic inflammatory response syndrome (SIRS) groups. The study of Taylor et al (2016) shows that a machine learning approach using random forest methods outperformed clinical decision rules and traditional analytic techniques for predicting in-hospital mortality of emergency department patients with sepsis. Kam and Kim (2017) used neural networks to develop detection models for the early stage of sepsis.

## 2.5 Critical Analysis of Popular Approaches to Disease Diagnosis

In this section, some of the popular approaches to disease diagnosis identified from Figure 2.1 will be critically analysed in Table 2.3.

**Table 2.3 Advantages and disadvantages of popular approaches to disease diagnosis**

| Popular approaches to disease diagnosis | Advantages | Disadvantages |
|---|---|---|
| Support vector machine (SVM) | 1. SVMs have strong generalization ability, as they are based on structural risk minimization principle. (Liu et al., 2010)<br><br>2. SVMs can be robust, even when the training sample has some bias (Auria and Moro, 2008).<br><br>3. The SVM algorithm is stable (Liu et al., 2010).<br><br>4. The SVM classifier has a global optimum solution, as an SVM can be formulated as a quardratic programming problem (Abe, 2005). | The extension of SVM to multiclass problems is not straightforward, as SVM uses direct decision functions (Abe, 2005). |

| | | |
|---|---|---|
| Artificial neural networks (ANN) | 1. ANNs do not rely on the prescribed relationship between input and output, but rather seek its own relationship (Dowla and Rogers, 1995).<br><br>2. ANN have the capability to detect complex nonlinear relationship between input and output (Tu, 1996).<br><br>3. ANNs can be relatively tolerant to noisy, incomplete, or even spurious data (Dowla and Rogers, 1995).<br><br>4. The advantages of ANNs also include highly parallel processing, distributed memory, and error-correction (Graham and Milne, 1991). | 1. An ANN is a "black box" in nature (Braspenning, Thuijsman and Weijters, 1995). It is difficult to interpret the ANN solutions.<br><br>2. It is difficult to incorporate knowledge of a given problem (Braspenning, Thuijsman and Weijters, 1995).<br><br>3. The disadvantages of ANNs also include proneness to overfitting, heavy computational burden, and empirical nature of model development (Tu, 1996). |
| Decision tree | 1. Decision trees are self-explanatory and easy to follow (Rokach and Maimon, | 1. Decision trees generally perform well if a few highly relevant |

| | | |
|---|---|---|
| | 2015).<br><br>2. Decision trees can deal with both nominal and numeric input values (Rokach and Maimon, 2015).<br><br>3. No assumptions are needed for the space distribution and the classifier structure (Rokach and Maimon, 2015). | input variables exist, but less so if many complex interactions are present between input variables (Rokach and Maimon, 2015).<br><br>2. The disadvantages of decision trees also include over-sensitivity to the training set, irrelevant input variables, and noise (Rokach and Maimon, 2015). |
| Random Forest | 1. Random forest is more robust than just a single decision tree (Cole, 2018).<br><br>2. Random forest balances bias and variance (Hodeghatta and Nayak, 2016).<br><br>3. Random forest is more efficient to build than other models, e.g., SVM (Hodeghatta and Nayak, 2016). | 1. Random forest is computationally expensive, as the number of recommended trees is large (Moreira, Carvalho, and Horváth, 2018).<br><br>2. Random forest is not easy to interpret (Gupta, 2018).<br><br>3. If the data consists of correlated input variables, random forest variable importance |

| | | |
|---|---|---|
| | | measure is not reliable and can be misleading (Gupta, 2018). |
| K-nearest neighbours (KNN) | 1. KNN is very simple to understand and easy to implement (Cord and Cunningham, 2008).<br><br>2. As the process of KNN is transparent, KNN is easy to debug (Cord and Cunningham, 2008).<br><br>3. KNN can be effective if an analysis of the neighbours is useful as explanation in situations where an explanation of the output of the classifier is useful (Cord and Cunningham, 2008). | 1. KNN is very sensitive to irrelevant or redundant input variables, as all input variables contribute to similarity and thus to the classification (Cord and Cunningham, 2008).<br><br>2. KNN may be outperformed by the classifiers, e.g., SVM and ANN (Cord and Cunningham, 2008). |
| Ensemble | 1. The ensemble classifier generally produces more accurate predictions than the base classifiers from which the ensemble classifier is made (Patil, Aghav and Sareen, 2016). | The main disadvantages of ensemble classifiers include the difficulties in the interpretation of the decisions of the ensemble and their extensive computational |

| | | |
|---|---|---|
| | 2. The ensemble classifier may be more stable and it may have a smaller variance than base classifiers (Homenda and Pedrycz, 2018) | requirements (El-Gayar, Kittler and Roli, 2010). |
| Naïve Bayes | 1. The computational complexity of naïve Bayes is low compared to other classifiers, e.g., decision trees (Maimon and Rokach, 2005).<br><br>2. Naïve Bayes classifiers are simple and easy to understand (Maimon and Rokach, 2005).<br><br>3. Other advantages of Naïve Bayes include the easy adaptation to the incremental learning environments and the resistance to irrelevant input variables (Maimon and Rokach, 2005). | 1. Naïve Bayes assumes that the input variables in the data set are completely independent of each other (Mehta, 2017), which is not practical in the real world (Nicolas, 2015).<br><br>2. Naïve Bayes is limited to simplified models, which in some cases are far from representing the complicated nature of the problem (Maimon and Rokach, 2005). |
| Bayesian Network | 1. A distinct advantage of | 1. The first disadvantage |

| | | |
|---|---|---|
| | Bayesian networks is the capability to incorporate domain-specific knowledge into the network structure, so that the overall joint probability distribution is represented as a set of conditionally independent relationships which are easier to characterize (Mittal and Kassim, 2007).<br><br>2. The advantages of Bayesian network include explicit uncertainty characterization, efficient computation, easy construction, adaptability, good generalization with limited training data, and easy retaining when pruning or adding new input variables (Mittal and Kassim, 2007). | of Bayesian network is the computational difficulty of exploring a previously unknown network (Holmes and Jain, 2008).<br><br>2. The second disadvantage is about the quality and extent of the prior beliefs used in the Bayesian network (Holmes and Jain, 2008). A Bayesian network is only useful when the prior knowledge is reliable (Holmes and Jain, 2008). An excessively optimistic or pessimistic expectation of the quality of the prior beliefs will distort the entire Bayesian network and invalidate the results (Holmes and Jain, 2008). |
| Logistic regression | 1. Logistic regression is easily interpretable (Moreira, Carvalho, and | 1. Logistic regression classifiers are restricted to linearly separable binary classification tasks |

| | | |
|---|---|---|
| | Horváth, 2018).<br><br>2. The conditions of using logistic regression are less restrictive than those for linear discriminant analysis (Tuffery, 2011). | (Moreira, Carvalho, and Horváth, 2018).<br><br>2. Logistic regression classifiers are sensitive to correlative input variables and outliers (Moreira, Carvalho, and Horváth, 2018). |

From Table 2.3, we can see that there are some issues, e.g., interpretability and dependence of input variables in the popular approaches to disease diagnosis. To address these issues, we develop a new modelling and prediction approach, i.e., the rule-based inferential modelling and prediction approach to disease diagnosis, which will be introduced in Chapter 3.

# Chapter 3 Research Methodologies

## 3.1 Introduction

In this chapter, the research methodologies of the thesis are introduced briefly from the perspective of basic principles. The remainder of the chapter is organised as follows. Section 3.2 presents an overview of general research methods. Section 3.3 justifies the choices of functions in the univariate functions approximation and the bivariate functions approximation. In Section 3.4, we briefly introduce the evidential reasoning (ER) rule. The research methods for rule-based inferential modelling and prediction are presented in Section 3.5. Section 3.6 describes an adapted genetic algorithm used for the bilevel optimisation of the maximum likelihood evidential reasoning (MAKER)-based models in this research. Section 3.7 summarises this chapter.

## 3.2 General Research Methods

Research is concerned with making efforts to develop a better understanding of the functioning of the world (Oliver, 2010). Qualitative and quantitative research methods are generally two fundamentally different paradigms through which we study the world (Brannen, 2005).

Qualitative research is a type of empirical research in which the data are not numeric (Punch, 2013). Qualitative research is aimed at studying the social reality of individuals, groups, and cultures (McLeod, 2017). A wide variety of methods have been developed to understand how people perceive their social realities and how they behave in the social world. These methods include diary accounts, questionnaires, observations, interviews, and ethnographies.

Quantitative research is a type of research where data are gathered in a numerical form (McLeod, 2017). The objective of quantitative research is to formulate general laws of behaviours and phenomena in different settings. The typical methods to obtain quantitative data are experiments, controlled observations, questionnaires, and so on. We can use statistics, machine learning, etc., to transform quantitative data into useful information for decision-making. All the data used in this research are numeric, and we propose a new approach for sepsis diagnosis using quantitative data, which is compared with alternative approaches. Hence, the research methods in this research are essentially quantitative.

## 3.3 Data Collection

As discussed in Section 1.6, functions approximation in this thesis is used to explore the capacity of models based on the MAKER framework to approximate functions, allowing us to achieve a possible compromise between the complexity and accuracy of the MAKER-based models. Hence, we need to evaluate the capabilities of MAKER-based models to approximate different types of functions. In this research, we select several types of univariate functions that have different characteristics to evaluate the general applicability of the approximation of the MAKER-based models. These functions are exponential functions $\mathrm{y} = a^x$, logarithmic functions $y = \log_a x$, power functions $y = x^a$, function $y = -(x - 0.5)^2 + 0.25$, and function $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$, which represent convex functions, concave functions, functions of which the mean curvatures are large, simple non-monotonic univariate functions, and complex non-monotonic univariate functions. The functions $\mathrm{y} = a^x$, $\mathrm{y} = \log_a x$, and $\mathrm{y} = x^a$ are monotonic and have different convexity or mean curvatures. The functions $\mathrm{y} = -(x - 0.5)^2 + 0.25$ and $\mathrm{y} = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$ are both non-monotonic, and they are unimodal and bimodal respectively. The non-monotonic functions are generally more complex than monotonic functions. All of these functions are selected to check whether the MAKER-based models can well approximate different types of univariate functions.

To evaluate the capability of the MAKER-based models to approximate more complex functions such as bivariate functions, we take the Himmelblau function as an example to perform functions approximation. Himmelblau function is a commonly used function for testing optimisation techniques (Chen et al., 2011), and this function is a multi-modal function.

On the basis of the functions approximation, we move on to the validation of the MAKER-based models on classical data sets, including the Banana data set, Haberman's survival data set, and Iris data set.

The Banana data set, which includes 5300 observations, is an artificial data set in which the observations belong to several clusters with a banana shape. In the data set, there are two input variables, At1 and At2, corresponding to the x-axis and y-axis respectively. The output variable of the data set has two classes that represent two banana shapes.

The Haberman's survival data set (Haberman, 1976) consists of observations about the survival of patients who underwent surgery for breast cancer from a study conducted between 1958 and 1970 at the University of Chicago's Billings Hospital. There are a total of 306 observations in the data set. The data set has three input variables: age of patient at time of operation, patients' year of operation, and number of positive axillary nodes detected. In the data set, the output variable, which is about survival status, contains two classes: the patient survived 5 years or longer and the patient died within 5 years.

The Iris data set (Fisher, 1936; 1950) is one of the most famous data sets in the field of machine learning. The data set contains 4 input variables: sepal length, sepal width, petal length, and petal width. The output variable of the data set is composed of three classes: Iris Setosa, Iris Versicolour, and Iris Virginica. Each of

these classes contains 50 observations, so there are altogether 150 observations in the data set.

All these data sets are selected to check whether the MAKER-based models can perform well compared to other models in the classification of these data sets, and all these data sets are downloaded from the Knowledge Extraction based on Evolutionary Learning (KEEL) data set repository (Alcalá-Fdez et al., 2011) available at http://sci2s.ugr.es/keel/category.php?cat=clas. This repository is a data set repository of KEEL, which is an open-source Java software tool that can be used for a large number of different knowledge data discovery tasks. Each of the data sets mentioned above is divided into five folds using distribution optimally balanced stratified cross-validation before being downloaded from the KEEL data set repository so that the class distribution of the whole data set is reflected in separate folds (Aggarwal, 2015).

Based on the functions approximation and classification of classical data sets, we apply the inferential modelling and prediction approach to establish a classifier for sepsis diagnosis.

The original sepsis data set is collected from several hospitals in Northwest England. The data are totally confidential and fully anonymous. Table 3.1 summarises the data in this research.

**Table 3.1 Summary of the original sepsis data set**

| Items | Instances | Features of Patients | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Identification | Biomarkers | | Patient Information | | | Diagnosis | |
| | | | Core | None-core | Basic | Hospitalisation on Sample Day | Patient Outcome | Patient Group | Others |
| Number | 922 | 1 | 5 | 6 | 3 | 2 | 3 | 1 | 2 |

In this data set, there are 922 instances and four types of patient features: identification, patients' test results of biomarkers (core and non-core), patient information, and patients' diagnoses. Among them, patients' test results of biomarkers include CRP, IL6, IL10, PCT, and WCC.

**Table 3.2 Categories of patients of original sepsis data set**

| Categories | Explanation of Propositions |
|---|---|
| Sepsis-1 ($\theta_1$) | Patients Are Diagnosed with Sepsis (Pathogens only in Blood). |
| Sepsis-2 ($\theta_2$) | Patients Are Diagnosed with Sepsis (Pathogens in Blood and Elsewhere). |
| Sepsis-3 ($\theta_3$) | Patients Are Diagnosed with Sepsis (No Pathogens in Blood, but Pathogens Elsewhere). |
| Unknown ($\theta_4$) | There Are No Sufficient Pieces of Evidence to Support the Diagnosis of Sepsis or That of Non-Sepsis (No Pathogens in Blood; No Pathogens Elsewhere; but with Clinical Adjudication of Infection). |
| Non-sepsis ($\theta_5$) | Patients Are Diagnosed with Non-Sepsis (No Pathogens in Blood; No Pathogens Elsewhere; No Clinical Adjudication of Infection). |

Table 3.2 shows the categories of patients of the original data set. Among the categories of patients, $\theta_1$, $\theta_2$, and $\theta_3$ indicate that patients have sepsis, whereas $\theta_5$ indicates that patients do not have sepsis. $\theta_4$ implies that patients may or may not have sepsis on the basis of objective pieces of evidence and clinical adjudication for infection. On the basis of the original sepsis data set, we perform the data preparation discussed in Section 6.2 to generate a sepsis data set for sepsis diagnosis, which is presented in Chapter 6.

## 3.4 Review of Evidential Reasoning Theories

As mentioned in Chapter 2, classification is one of the most common supervised machine learning tasks. Classification imprecision is likely due to the fact that the values of an input variable of an observation cannot be mapped to a certain class explicitly (Zhang et al., 2014; Denoeux, 2000; 1995). Dempster-Shafer evidence theory (DST) can be used to deal with classification imprecision. In DST, we should first identify a frame of discernment (FoD) to contain all of pre-assigned class memberships. Then we can perform basic belief assignment to generate a belief distribution (BD) where the belief degrees are used to measure the extent to which data fragments of predictor variables point to different classes or subsets of classes. The BD of a data fragment of an input variable can be identified as a piece of evidence. We can take a number of ways e.g., core sample (Zhang et al., 2014), neural network (Denoeux, 2000), k-NN (Denoeux, 1995), and expert system (Dymova, Sevastianov and Bartosiewicz, 2010) to generate BDs of data fragments of input variables. Finally, we can take Dempster's combination (DC) rule to combine different pieces of evidence among input variables together to make a classification decision on combined BDs.

The classification decision process based on DC works well on classification imprecision. However, it does not embrace the consideration of inherent properties of evidence, i.e., quality of information source and relative importance of evidence. Evidential reasoning (ER) rule (Yang and Xu, 2013) was proposed to consider quality of information source, i.e., reliability and relative importance i.e., weights when we combine evidence. In ER rule, new concepts e.g., weighted evidence (WE), and weighted evidence with reliability (WER) were put forward to describe characteristics of evidence in complement of BD in DST. The ER rule assimilates DST and original ER algorithm, which is revealed in the evidence combination process i.e., the orthogonal sum operation on WEs or WERs. One of the most compelling characteristics of ER rule is that it makes up a generic process of

conjunctive probabilistic reasoning, or a generalized Bayesian process implemented on the power set of FoD.

There are some connections between ER rule and DC rule, and ER rule and original ER algorithm. The DC rule has been proven to be a special case of ER rule when all the evidence is fully reliable. It has also been proven that the original ER algorithm is a special case of ER rule when the normalised weights of all the evidence equals their respective reliabilities (Yang and Xu, 2013). The ER process inherently contains the belief structure to model different types of uncertainty (Yang and Singh, 1994; Xu, 2011), and the rule-or-utility-based information transformation techniques (Yang, 2001). In addition, ER algorithm has been applied in a wide range of areas and it has been integrated into traditional if-then-rule-based systems to generate the belief rule based (BRB) systems (Chen et al., 2015) which have been used for modelling of classification problems (Jiao et al., 2015; Chang et al., 2016; Kong et al., 2016). One of the problems that BRB systems have is high multiplicative complexity on combination of referential values of input variables in the base of belief rule (Chen et al., 2015).

## 3.5 Outline of Evidential Reasoning Rule

In the framework of the ER rule, $\Theta = \{h_1, \dots, h_K\}$ is a set of mutually exclusive and collectively exhaustive hypotheses. $\Theta$ is called a frame of discernment (FoD), whose power set includes all its subsets. Generally, a piece of evidence is a random set profiled by a belief distribution, as displayed in Equation (3.1),

$$e_j = \left\{ \left(\theta, p_{\theta,j}\right), \forall \theta \subseteq \Theta, \sum\nolimits_{\theta \subseteq \Theta} p_{\theta,j} = 1 \right\} \tag{3.1}$$

where $\left(\theta, p_{\theta,j}\right)$ is an element of evidence $e_j$, indicating that the $j^{th}$ piece of evidence points to proposition $\theta$ (θ can be any subset of $\Theta$ except the empty set)

with a probability $p_{\theta,j}$. In the case of the system of the MAKER framework of this research, a piece of evidence $\mathrm{e}_j^i$, that is, the $j^{th}$ piece of evidence from the $i^{th}$ input variable, refers to the data on the $j^{th}$ referential value of the $i^{th}$ input variable, pointing to different class memberships of the output variable with corresponding probabilities. An element of a piece of evidence indicates the data on the $j^{th}$ referential value of the $i^{th}$ input variable that points exactly to the $k^{th}$ class of the output variable with probability $p_{i,j}^k$.

Additionally, a piece of evidence $e_j$ is generally associated with reliability $r_j$ and weight $w_j$. Reliability is the ability or quality of data sources from which evidence is generated, and it generally measures the degree of support or opposition from evidence to a proposition. The reliability of a piece of evidence is the inherent property of that evidence. The weight is used to reflect the relative importance of evidence in comparison with other evidence, and the weight can be judged by decision makers. The weight can be subjective and different from reliability if different pieces of evidence are acquired from different sources and measured in different ways (Xu et al., 2017). If all pieces of evidence are measured in a joint space (Yang and Xu, 2014) or acquired from a data source, then $w_j = r_j$. Further, $w_{\theta,j} = r_{\theta,j}$, in which $w_{\theta,j}$ and $r_{\theta,j}$ refer to the weight and reliability of an element $e_{\theta,j}$ of evidence $e_j$ that points to assertion θ, respectively, if all pieces of evidence are acquired from a data source under the MAKER framework.

The predictive power of a single piece of evidence is limited. To achieve greater predictive power, it is necessary to combine different pieces of evidence to generate a probability distribution of combined evidence under different class memberships of the output variable, that is, an belief rule base, using weighted belief distribution with reliability, which includes consideration of the aforementioned properties of evidence (belief distribution, reliability, and weight). Weighted belief distribution with reliability is defined in Equation (3.2),

$$m_j = \{(\theta, \widetilde{m}_{\theta,j}), \forall \theta \subseteq \Theta, (P(\Theta), \widetilde{m}_{P(\Theta),j})\}, \quad (3.2)$$

$$\widetilde{m}_{\theta,j} = \begin{cases} 0, & \theta = \emptyset \\ c_{rw} m_{\theta,j}, & \theta \subseteq \Theta,\ \theta \neq \emptyset, \\ c_{rw,j}(1 - r_j), & \theta = P(\Theta) \end{cases}$$

where $m_{\theta,j} = w_j p_{\theta,j}$ and $\widetilde{m}_{\theta,j}$ is used to measure the degree of support from $e_j$ to $\theta$, with consideration of both weight $(w_j)$ and reliability $(r_j)$, and $c_{rw,j} = \frac{1}{1+w_j-r_j}$, a normalisation factor, satisfies $\sum_{\theta \subseteq \Theta} \widetilde{m}_{\theta,j} + \widetilde{m}_{P(\Theta),j} = 1$, if $\sum_{\theta \subseteq \Theta} p_{\theta,j} = 1$.

If two pieces of evidence are independent of each other – which means the information a piece of evidence carries is not dependent on the other piece of evidence and vice versa – the combined degree of belief $p_{\theta,e(2)}$ to which the two pieces of evidence $e_{j_1}$ and $e_{j_2}$ $(j_1 = 1, \ldots, J_1 \; and \; j_2 = 1, \ldots, J_2, j_1 \neq j_2)$ jointly support proposition $\theta$ is given by Equation (3.3),

$$p_{\theta,e(2)} = \begin{cases} 0, \ \theta = \emptyset \\ \frac{\widehat{m}_{\theta,e(2)}}{\sum_{D \subseteq \Theta} \widehat{m}_{D,e(2)}}, \theta \subseteq \Theta, \theta \neq \emptyset, \end{cases} \quad (3.3)$$

$$\widehat{m}_{\theta,e(2)} = \left[(1 - r_{j_2}) m_{\theta,j_1} + (1 - r_{j_1}) m_{\theta,j_2}\right] + \sum_{B \cap C = \theta} m_{B,j_1} m_{C,j_2}, \ \forall \theta \subseteq \Theta,$$

where $m_{\theta,j_1} = w_{\theta,j_1} * p_{\theta,j_1}$, and $m_{\theta,j_2} = w_{\theta,j_2} * p_{\theta,j_2}$.

The recursive formulae of the ER rule in Equation (3.3) can be used to combine multiple pieces of evidence in any order. It has been proved that Dempster's rule is a special case of the ER rule in Equation (3.3) when evidence is fully reliable.

## 3.6 Research Methods for Rule-based Inferential Modelling and Prediction

In this section, we briefly introduce the research methods for rule-based inferential modelling and prediction based on the MAKER framework, which will be discussed in Chapters 4, 5, and 6.

Suppose in the real world, there is a complex numerical system where a sample input-output data set of N instances $\mathrm{x(t)} = \{x_n(t) \mid n = 1, \dots, N\}$ is recorded at sampling time t. These instances are identified by variables such as M input variables $x_n\mathrm{(t)} = \{x_{n,i}(t) \mid n = 1, \dots, N; i = 1, \dots, M\}$ and an output variable $\mathrm{y(t)} = \{y_n(t) \mid n = 1, \dots, N\}$. These instances need to be classified as one of the class memberships in $\Theta = \{k \mid k = 1, \dots, K\}$, where an integer is used to represent a class membership. That is, a value of a nominal output variable $y_n(t)$ is assigned by a class membership in $\Theta$, which means $\mathrm{y(t)} = \{y_n(t) \mid y_n(t) = 1, \dots, K; n = 1, \dots, N\}$.

### 3.6.1 Evidence Acquisition from Data of Input Variables

To construct the system of the MAKER framework, we need to determine referential values for each of the input variables. As adjustable parameters, referential values can be initially determined by expertise or random rules without prior knowledge and subsequently trained using an input-output data set under a certain optimisation objective (Xu et al., 2017). With referential values for an input variable, we can transform input value $x_{i,n}$, of which the corresponding output value $y_n$ is k to the belief distribution of referential value $A_j^i$, as shown in Equation (3.4).

$$S_i(x_{n,i}) = \left\{\left(A_j^i, \alpha_{n,i,j}^k\right), j = 1, \dots, J_i\right\} \tag{3.4}$$

where

$\alpha_{n,i,j}^{k} = \frac{A_{j+1}^{i} - x_{n,i}}{A_{j+1}^{i} - A_{j}^{i}}$ and $\alpha_{n,i,j+1}^{k} = 1 - \alpha_{n,i,j}^{k}$, if $A_j^i \leq x_{n,i} \leq A_{j+1}^i$,

$\alpha_{n,i,j'}^{k} = 0, for\ j' = 1, \ldots, J_i$ and $j' \neq j, j+1$.

In Equation (3.4), $\alpha_{n,i,j}^{k}$ is the similarity degree to which the $n^{th}$ input value $x_{n,i}$ of $i^{th}$ input variable matches the referential value $A_j^i$ under the $k^{th}$ class membership of the output variable. After all the input values are transformed into belief distributions of referential values, the similarity degrees are aggregated in terms of referential values under different class memberships of the output variable to generate the frequencies of the referential values under these different class memberships, as shown in Equation (3.5). In the further study, we may also consider the non-linear situation in the data transformation or evidence mapping. For example, we may explore the possibility of using nonlinear utility functions to transform input values to belief distribution of referential value.

$$\alpha_{i,j}^{k} = \sum_{n=1}^{N} \alpha_{n,i,j}^{k}. \tag{3.5}$$

**Table 3.3 Frequencies of referential values of an input variable**

| $y_n \backslash x_{n,i}$ | $A_1^i$ | ⋯ | $A_j^i$ | ⋯ | $A_{J_i}^i$ | total |
|---|---|---|---|---|---|---|
| 1 | $\alpha_{i,1}^{1}$ | ⋯ | $\alpha_{i,j}^{1}$ | ⋯ | $\alpha_{i,J_i}^{1}$ | $\sum_{j=1}^{J_i} \alpha_{i,j}^{1}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| k | $\alpha_{i,1}^{k}$ | ⋯ | $\alpha_{i,j}^{k}$ | ⋯ | $\alpha_{i,J_i}^{k}$ | $\sum_{j=1}^{J_i} \alpha_{i,j}^{k}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| K | $\alpha_{i,1}^{K}$ | ⋯ | $\alpha_{i,j}^{K}$ | ⋯ | $\alpha_{i,J_i}^{K}$ | $\sum_{j=1}^{J_i} \alpha_{i,j}^{K}$ |
| total | $\sum_{k=1}^{K} \alpha_{i,1}^{k}$ | ⋯ | $\sum_{k=1}^{K} \alpha_{i,j}^{k}$ | ⋯ | $\sum_{k=1}^{K} \alpha_{i,J_i}^{k}$ | N |

Table 3.3 shows all the frequencies of the referential values of an input variable. According to Table 1, $\sum_{j=1}^{J_i} \sum_{k=1}^{K} \alpha_{i,j}^{k} = \sum_{k=1}^{K} \sum_{j=1}^{J_i} \alpha_{i,j}^{k} = N$. Then, the likelihood $c_{i,j}^{k}$, with

which the $j^{th}$ referential value of the $i^{th}$ input variable is true if the $k^{th}$ class membership of output variable is true, is calculated as shown in Equation (3.6).

$$c_{i,j}^{k} = \frac{\alpha_{i,j}^{k}}{\sum_{j=1}^{J_i} \alpha_{i,j}^{k}}, for\ \sum_{j=1}^{J_i} \alpha_{i,j}^{k} \neq 0, \tag{3.6}$$

$$c_{i,j}^{k} = 0, for\ \sum_{j=1}^{J_i} \alpha_{i,j}^{k} = 0.$$

**Table 3.4 Likelihoods of referential values of an input variable**

| $y_n \backslash x_{n,i}$ | $A_1^i$ | ⋯ | $A_j^i$ | ⋯ | $A_{J_i}^i$ | total |
|---|---|---|---|---|---|---|
| 1 | $c_{i,1}^{1}$ | ⋯ | $c_{i,j}^{1}$ | ⋯ | $c_{i,J_i}^{1}$ | $\sum_{j=1}^{J_i} c_{i,j}^{1}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| k | $c_{i,1}^{k}$ | ⋯ | $c_{i,j}^{k}$ | ⋯ | $c_{i,J_i}^{k}$ | $\sum_{j=1}^{J_i} c_{i,j}^{k}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| K | $c_{i,1}^{K}$ | ⋯ | $c_{i,j}^{K}$ | ⋯ | $c_{i,J_i}^{K}$ | $\sum_{j=1}^{J_i} c_{i,j}^{K}$ |
| total | $\sum_{k=1}^{K} c_{i,1}^{k}$ | ⋯ | $\sum_{k=1}^{K} c_{i,j}^{k}$ | ⋯ | $\sum_{k=1}^{K} c_{i,J_i}^{k}$ | K |

Table 3.4 displays the likelihoods calculated from the frequencies in Table 3.3. It is obvious from Table 3.4 that $\sum_{j=1}^{J_i} \sum_{k=1}^{K} c_{i,j}^{k} = \sum_{k=1}^{K} \sum_{j=1}^{J_i} c_{i,j}^{k} = K$. Regarding likelihoods, the probabilities with which the referential value $A_j^i$ points to $k^{th}$ class membership of output variable are given by Equation (3.7).

$$p_{i,j}^{k} = \frac{c_{i,j}^{k}}{\sum_{k=1}^{K} c_{i,j}^{k}}, for\ \sum_{k=1}^{K} c_{i,j}^{k} \neq 0, \tag{3.7}$$

$$p_{i,j}^{k} = 0, for\ \sum_{k=1}^{K} c_{i,j}^{k} = 0.$$

**Table 3.5 The probabilities with which the referential values of the observed values of the input variables point to different classes of the output variable of a data set**

| $y_n \backslash x_{n,i}$ | $A_1^i$ | ⋯ | $A_j^i$ | ⋯ | $A_{J_i}^i$ |
|---|---|---|---|---|---|
| 1 | $p_{i,1}^1$ | ⋯ | $p_{i,j}^1$ | ⋯ | $p_{i,J_i}^1$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| k | $p_{i,1}^k$ | ⋯ | $p_{i,j}^k$ | ⋯ | $p_{i,J_i}^k$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| K | $p_{i,1}^K$ | ⋯ | $p_{i,j}^K$ | ⋯ | $p_{i,J_i}^K$ |

Table 3.5 exhibits the degrees of belief calculated by the normalisation of the likelihoods in Table 3.4. Now a piece of evidence can be defined as a set of degrees of belief with which data on referential values point to different class memberships of the output variable, as presented in Equation (3.8).

$$e_j^i = \left\{ \left(k, p_{i,j}^k\right), \forall k \subseteq \Theta, \sum\nolimits_{\theta \subseteq \Theta} \beta_{i,j}^k = 1 \right\} \tag{3.8}$$

Table 3.6 visualises the relationship between evidence $e_j^i$ and degree of belief $\beta_{i,j}^k$.

**Table 3.6 Evidence and degrees of belief of referential values of an input variable**

| $y_n \backslash x_{n,i}$ | $e_1^i$<br>$A_1^i$ | ⋯<br>⋯ | $e_j^i$<br>$A_j^i$ | ⋯<br>⋯ | $e_{J_i}^i$<br>$A_{J_i}^i$ |
|---|---|---|---|---|---|
| 1 | $\beta_{i,1}^1$ | ⋯ | $\beta_{i,j}^1$ | ⋯ | $\beta_{i,J_i}^1$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| k | $\beta_{i,1}^k$ | ⋯ | $\beta_{i,j}^k$ | ⋯ | $\beta_{i,J_i}^k$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| K | $\beta_{i,1}^K$ | ⋯ | $\beta_{i,j}^K$ | ⋯ | $\beta_{i,J_i}^K$ |

## 3.6.2 Interdependence between Pairs of Evidence

If multiple input variables are taken into consideration at the same time, the vector of input variables, that is, $x_n = \{x_{i_1,n}, \dots, x_{i_l,n}, \dots, x_{i_m,n} | n = 1, \dots, N; i_l = 1, \dots, M; j_l =$

$1, \dots, J_{i_l}; l = 1, \dots, m; m = 2, \dots, M\}$, can be transformed to the distribution in Equation (3.9) for the combination of referential values $A_{j_1 \dots j_l \dots j_m}^{i_1 \dots i_l \dots i_m}$,

$$S_{i_1 \dots i_l \dots i_m}(x_{i,n}) = \left\{\left(A_{j_1 \dots j_l \dots j_m}^{i_1 \dots i_l \dots i_m}, \alpha_{n,i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m}^{k}\right) | n = 1, \dots, N; i_l = 1, \dots, M; j_l = 1, \dots, J_{i_l}; l = 1, \dots, m; m = 2, \dots, M\right\}. \quad (3.9)$$

where

$A_{j_1 \dots j_l \dots j_m}^{i_1 \dots i_l \dots i_m} = \left\{A_{j_1}^{i_1}, \dots, A_{j_l}^{i_l}, \dots, A_{j_m}^{i_m}\right\}$,

$\alpha_{n,i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m}^{k} = \alpha_{n,i_1,j_1}^{k} * \dots * \alpha_{n,i_l,j_l}^{k} * \dots * \alpha_{n,i_m,j_m}^{k}$,

$\alpha_{n,i_l,j_l}^{k} = \frac{A_{j_l+1}^{i_l} - x_{n,i_l}^{k}}{A_{j_l+1}^{i_l} - A_{j_l}^{i_l}}$ and $\alpha_{n,i_l,j_l+1}^{k} = 1 - \alpha_{n,i_l,j_l}^{k}$, if $A_{j_l}^{i_l} \leq x_{n,i_l}^{k} \leq A_{j_l+1}^{i_l}$,

$\alpha_{n,i_l,j_l'}^{k} = 0, for\ j_l' = 1, \dots, J_{i_l}$ and $j_l' \neq j_l, j_l + 1$.

$\alpha_{n,i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m}^{k}$ is the similarity degree to which the $n^{th}$ input vector $(x_n)$ of the input variable matches the combination of referential values, that is, $A_{j_1 \dots j_l \dots j_m}^{i_1 \dots i_l \dots i_m}$, under the $k^{th}$ class membership of the output variable. Then, we can aggregate similarity degrees in terms of combinations of referential values to generate the frequencies of combinations of referential values, as expressed in Equation (3.10).

$$\alpha_{i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m}^{k} = \sum_{n=1}^{N} \alpha_{n,i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m}^{k}. \quad (3.10)$$

Based on these frequencies, the likelihood $c_{k,i_1 \dots i_U, j_1 \dots j_U}$ to which the combination of referential values $A_{j_1 \dots j_l \dots j_m}^{i_1 \dots i_l \dots i_m}$ is expected to occur for the $k^{th}$ class membership of output variable is given by Equation (3.11).

$$c_{i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m}^{k} = p\left(A_{j_1 \dots j_l \dots j_m}^{i_1 \dots i_l \dots i_m} \middle| y_k\right) = \frac{\alpha_{i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m}^{k}}{\delta^k}, \quad (3.11)$$

$$\delta^k = \sum_{i_1 \dots i_l \dots i_m \in T} \sum_{j_1 \dots j_l \dots j_m \in H} \alpha^k_{i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m}$$

where T = $\{i_1 \dots i_l \dots i_m | i_l = 1, \dots, M; l = 1, \dots, m\}$ and H = $\{j_1 \dots j_l \dots j_m | j_l = 1, \dots, J_{i_l}; i_l = 1, \dots, M; l = 1, \dots, m\}$.

The degree of belief $p^k_{i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m}$ with which the combination of referential values $A^{i_1 \dots i_l \dots i_m}_{j_1 \dots j_l \dots j_m}$ points to the $k^{th}$ class is calculated as shown in Equation (3.12).

$$p^k_{i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m} = \begin{cases} \frac{c^k_{i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m}}{\sum_{k=1}^K c^k_{i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m}}, & \sum_{k=1}^K c^k_{i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m} \neq 0 \\ 0, & \sum_{k=1}^K c^k_{i_1 \dots i_l \dots i_m, j_1 \dots j_l \dots j_m} = 0 \end{cases} \quad (3.12)$$

In the original ER rule, we assume any two pieces of evidence for a combination are independent of each other. To enhance the generality of this rule, we introduce a new concept, 'interdependence index', denoted by 'ɑ', to measure the degree of interdependence between a pair of evidence. The interdependence index ɑ is defined by Equation (3.13),

$$\alpha_{\theta,(i_1,j_1),(i_2,j_2)} = \begin{cases} \frac{p_{\theta,i_1 i_2, j_1 j_2}}{p_{\theta,i_1,j_1} * p_{\theta,i_2,j_2}}, if\, p_{\theta,i_1,j_1} \neq 0 \; and \; p_{\theta,i_2,j_2} \neq 0 \\ 0, \quad if\, p_{\theta,i_1,j_1} = 0 \; or \; p_{\theta,i_2,j_2} = 0 \end{cases} \quad (3.13)$$

$$\alpha^k_{(i_1,j_1),(i_2,j_2)} = \begin{cases} \frac{p^k_{i_1 i_2, j_1 j_2}}{p^k_{i_1,j_1} * p^k_{i_2,j_2}}, if\, p^k_{i_1,j_1} \neq 0 \; and \; p^k_{i_1,j_1} \neq 0 \\ 0, \quad if\, p^k_{i_1,j_1} = 0 \; or \; p^k_{i_1,j_1} = 0 \end{cases}$$

Where the first equation is the general form of the interdependence index, and the second equation is the concrete form of the interdependence index under the new data-driven ER modelling approach. $p_{\theta,i_1 i_2,j_1 j_2}$ *or* $p^k_{i_1 i_2,j_1 j_2}$ is the degree of belief to which two pieces of evidence $e^{i_1}_{j_1}$ and $e^{i_2}_{j_2}$ jointly support proposition θ or $k^{th}$ class membership of the output variable. The $p_{\theta,i_1,j_1}$ *or* $p^k_{i_1,j_1}$ is the degree of belief that evidence $e^{i_1}_{j_1}$ points to proposition θ or $k^{th}$ class membership of the output variable,

and so is $p_{\theta,i_2,j_2}$ or $p^k_{i_2,j_2}$. Additionally, $\alpha^k_{(i_1,j_1),(i_2,j_2)}$ is the interdependence index to measure the interdependence between evidence $e^{i_1}_{j_1}$ and evidence $e^{i_2}_{j_2}$ under the $k^{th}$ class membership of the output variable. In the current study, the conditional independence between input variables is assumed to be true. According to Della Riccia, Kruse and Lenz (2014), the conditional independence has the following general description. Let X, Y, and Z be three disjoint sets of input variables, and both X and Y are nonempty. X is called independent of Y given Z with respect to a possibility distribution π on Ω, if for all instants of the input variables in Z, no information about the values of the input variables in Y changes probability degrees of the tuples over the input variables in X. In the further study, we may also consider the conditional dependence between the input variables.

### 3.6.3 Evidence Combination Based on the MAKER Framework

With consideration of interdependence between a pair of evidence, the combined degree of belief $p_{\theta,e(2)}$ to which two pieces of evidence $e_{j_1}$ and $e_{j_2}$ ($j_1\ and\ j_2 \in Z^+, j_1 \neq j_2$) jointly support proposition $\theta$ can be calculated by MAKER, as shown in Equation (3.14),

$$p_{\theta,e(2)} = \begin{cases} 0, & \theta = \emptyset \\ \frac{m_{\theta,e(2)}}{\sum_{C\subseteq\Theta} m_{C,e(2)}}, & \theta \subseteq \Theta \end{cases} \tag{3.14}$$

$m_{\theta,e(2)} = \left[\left(1-r_{j_2}\right)m_{\theta,j_1} + \left(1-r_{j_1}\right)m_{\theta,j_2}\right] + \sum_{A\cap B=\theta} \gamma_{A,B,j_1,j_2}\alpha_{A,B,j_1,j_2}m_{A,j_1}m_{B,j_2}$.

where $m_{\theta,j_1} = w_{\theta,j_1} * p_{\theta,j_1} = \omega_{j_1} * r_{\theta,j_1} * p_{\theta,j_1}$, and $m_{\theta,j_2} = w_{\theta,j_2} * p_{\theta,j_2} = \omega_{j_2} * r_{\theta,j_2} * p_{\theta,j_2}$;

$r_{j_1} = \sum_{\theta\subseteq\Theta} r_{\theta,j_1} p(e_{j_1}(\theta))$ is the reliability of evidence $e_{j_1}$; $r_{j_2} = \sum_{\theta\subseteq\Theta} r_{\theta,j_2} p(e_{j_2}(\theta))$ is the reliability of evidence $e_{j_2}$; and $\gamma_{A,B,j_1,j_2}$ is a non-negative parameter reflecting the degree of joint support for θ from both evidence $e_{j_1}$ and evidence $e_{j_2}$ relative to the individual support from evidence $e_{j_1}$ to proposition A and that from evidence

$e_{j_2}$ to proposition B. If all data are measured in a joint space, $\omega_{j_1} = 1$ and $\omega_{j_2} = 1$, and $w_{\theta,j_1} = r_{\theta,j_1}$ and $w_{\theta,j_2} = r_{\theta,j_2}$. The parameter $\gamma_{A,B,j_1,j_2}$ can be trained in deep learning.

The concrete form of the evidence combination based on the MAKER rule under the rule-based inferential and prediction approach is given in Equation (3.15).

$$p^k_{(i_1,j_1),(i_2,j_2)} = \begin{cases} 0, & \sum_{k=1}^{K} m^k_{(i_1,j_1),(i_2,j_2)} = 0 \\ \frac{m^k_{(i_1,j_1),(i_2,j_2)}}{\sum_{k=1}^{K} m^k_{(i_1,j_1),(i_2,j_2)}}, & \sum_{k=1}^{K} m^k_{(i_1,j_1),(i_2,j_2)} \neq 0 \end{cases}, \tag{3.15}$$

$$m^k_{(i_1,j_1),(i_2,j_2)} = \left[\left(1 - r_{i_2,j_2}\right)m^k_{i_1,j_1} + \left(1 - r_{i_1,j_1}\right)m^k_{i_2,j_2}\right] + \gamma^k_{(i_1,j_1),(i_2,j_2)}\alpha^k_{(i_1,j_1),(i_2,j_2)}m^k_{i_1,j_1}m^k_{i_2,j_2}$$

where $m^k_{i_1,j_1} = w^k_{i_1,j_1} * p^k_{i_1,j_1} = \omega_{i_1,j_1} * r^k_{i_1,j_1} * p^k_{i_1,j_1}$, and $m^k_{i_2,j_2} = w^k_{i_2,j_2} * p^k_{i_2,j_2} = \omega_{i_2,j_2} * r^k_{i_2,j_2} * p^k_{i_2,j_2}$; $r_{i_1,j_1} = \sum_{\theta \subseteq \Theta} r^k_{i_1,j_1} p(e^{i_1}_{j_1}(\theta))$ is the reliability of evidence $e^{i_1}_{j_1}$; $r_{i_2,j_2} = \sum_{\theta \subseteq \Theta} r^k_{i_2,j_2} p(e^{i_2}_{j_2}(\theta))$ is the reliability of evidence $e^{i_2}_{j_2}$; and $\gamma^k_{(i_1,j_1),(i_2,j_2)}$ is a non-negative parameter reflecting the degree of joint support for $k^{th}$ class membership of output variable from both evidence $e^{i_1}_{j_1}$ and evidence $e^{i_2}_{j_2}$ relative to the individual support from evidence $e^{i_1}_{j_1}$ to $k^{th}$ class membership of output variable and that from evidence $e^{i_2}_{j_2}$ to $k^{th}$ class membership of output variable. In studies that include the function approximation and cross-validation of classical data sets and the data set of sepsis of the new data-driven ER modelling approach, $\omega_{i_1,j_1} = 1$ and $\omega_{i_2,j_2} = 1$, and we $w^k_{i_1,j_1} = r^k_{i_1,j_1}$ and $w^k_{i_2,j_2} = r^k_{i_2,j_2}$. The parameter $\gamma^k_{(i_1,j_1),(i_2,j_2)}$ can be trained in deep learning, and it is assumed to be 1 in the numerical examples in this report.

### 3.6.4 Prediction Scheme Based on the MAKER Framework

On the basis of the belief rule base, we can make predictions about the class membership of output variable for each input vector $x_n = \{x_{n,i_1}, \dots, x_{i_l,n}, \dots, x_{i_m,n} | n =$

$1, \ldots, N; i_l = 1, \ldots, M; j_l = 1, \ldots, J_{i_l}; l = 1, \ldots, m; m = 2, \ldots, M\}$ using MAKER. Each value $x_{i_l,n}$ of $x_n$ can be located between a set of adjacent referential values of each input variable, and this set of adjacent referential values is therefore activated by $x_{i_l,n}$. Then corresponding similarity degree $S'_{i_u,j_u,n}$ between $x_{i_u,n}$ and each of the referential values between which $x_{i_u,n}$ is located can be calculated as displayed in Equation (3.16).

$$S'_i(x_{n,i}) = \{(A^i_j, \alpha'_{n,i,j}) | n = 1, \ldots, N; i = 1, \ldots, M; j = 1, \ldots, J_i\} \quad (3.16)$$

where

$\alpha'_{n,i,j} = \frac{A^i_{j+1} - x_{n,i}}{A^i_{j+1} - A^i_j}$ *and* $\alpha'_{n,i,j+1} = 1 - \alpha'_{n,i,j}$, *if* $A^i_j \le x_{n,i} \le A^i_{j+1}$,

$\alpha'_{n,i,j'} = 0, for\ j' = 1, \ldots, J_i$ *and* $j' \neq j, j+1$.

Given that we have an input vector $x_n = \{x_{n,i_1}, x_{n,i_2}\}$ available and we know $A^{i_1}_r \le x_{n,i_1} \le A^{i_1}_{r+1}$ and $A^{i_2}_t \le x_{n,i_2} \le A^{i_2}_{t+1}$, we can calculate the joint similarity degrees between input vector $x_n$ and activated combinations of referential values $\{A^{i_1}_r, A^{i_2}_t\}$, $\{A^{i_1}_r, A^{i_2}_{t+1}\}$, $\{A^{i_1}_{r+1}, A^{i_2}_t\}$, and $\{A^{i_1}_{r+1}, A^{i_2}_{t+1}\}$ using Equation (3.17).

$$S'_{i_1 \ldots i_l \ldots i_m}(x_{i,n}) = \left\{\left(A^{i_1 \ldots i_l \ldots i_m}_{j_1 \ldots j_l \ldots j_m}, \alpha'_{n,i_1 \ldots i_l \ldots i_m, j_1 \ldots j_l \ldots j_m}\right) | n = 1, \ldots, N; i_l = 1, \ldots, M; j_l = 1, \ldots, J_{i_l}; l = 1, \ldots, m; m = 2, \ldots, M\right\}. \quad (3.17)$$

where

$A^{i_1 \ldots i_l \ldots i_m}_{j_1 \ldots j_l \ldots j_m} = \left\{A^{i_1}_{j_1}, \ldots, A^{i_l}_{j_l}, \ldots, A^{i_m}_{j_m}\right\}$,

$\alpha'_{n,i_1 \ldots i_l \ldots i_m, j_1 \ldots j_l \ldots j_m} = \alpha'_{n,i_1,j_1} * \ldots * \alpha'_{n,i_l,j_l} * \ldots * \alpha'_{n,i_m,j_m}$.

where $\alpha'_{n,i_1 \ldots i_l \ldots i_m, j_1 \ldots j_l \ldots j_m}$ is the joint similarity degree to which a given input vector $x_n$ matches the combination of referential values, that is, $A^{i_1 \ldots i_l \ldots i_m}_{j_1 \ldots j_l \ldots j_m}$.

The joint similarity degrees between the input vector and activated combinations of referential values indicate to what degree we should invoke activated evidence combinations $\{e_r^{i_1}, e_t^{i_2}\}$, $\{e_r^{i_1}, e_{t+1}^{i_2}\}$, $\{e_{r+1}^{i_1}, e_t^{i_2}\}$, and $\{e_{r+1}^{i_1}, e_{t+1}^{i_2}\}$ to predict the probability for each class membership of a given input vector $x_n$. To apply MAKER rule to combine activated evidence for prediction, we still need to know the weight (reliability) of each activated evidence combination. From Equation (3.2), we can obtain Equation (3.18) as $w_{e(L)} = r_{e(L)}$ in this research.

$$r_{\theta,e(L)} = \begin{cases} 0, & \theta = \emptyset \\ \frac{m_{\theta,e(L)}}{p_{\theta,e(L)}}, & \theta \subseteq \Theta,\ \theta \neq \emptyset. \\ 1 - m_{\theta,e(L)}, & \theta = P(\Theta) \end{cases} \tag{3.18}$$

With Equation (3.18), it is not difficult to prove Equation (3.19).

$$r_{\theta,e(L)} = \frac{m_{\theta,e(L)}}{p_{\theta,e(L)}} = r_{P(\Theta),e(L)} = 1 - m_{P(\Theta),e(L)}. \tag{3.19}$$

With Equation (3.19), we can have the weight or reliability for each activated evidence combination. On the basis of the joint similarity degrees between the input vector and the activated combinations of referential values and the weight or reliability for each activated evidence combination, we can have an updated weight or reliability for each activated evidence combination considering the degree to which we should invoke these pieces of activated evidence to predict the probability for each class membership of a given input vector $x_n$. Then, with the updated weight or reliability for each activated evidence combination and the combined degrees of belief of each activated evidence combination acquired from the belief rule base, we can combine these pieces of activated evidence combinations to predict the probability of each class membership of a given input vector $x_n$ using the adapted conjunctive ER rule, as shown in Equations (14) and (15). The parameters $\gamma_{A,B,j_1,j_2}$ and $\alpha_{A,B,j_1,j_2}$ in Eq. (14) or $\gamma^k_{(i_1,j_1),(i_2,j_2)}$ and $\alpha^k_{(i_1,j_1),(i_2,j_2)}$ in

Equation (15) are assumed to be 1 in the numerical examples of this report, but they can be trained in deep learning.

If class k, which has the largest probability in the predicted probabilities set $\hat{P}_n = \{\hat{p}_{k,n}, k = 1,2,\dots,K, n = 1,\dots,N\}$ for input vector $x_n$, is the same as the $n^{th}$ value $y_n$ $(y_n = 1,\dots,K)$ of the output variable, we can have the judgment that the prediction for $x_n$ is correct. The accuracy, that is, the ratio of the total number of correct predictions to the total number of predictions, can be used to represent how the classifiers established by the approach of rule-based inferential modelling and prediction and other classical classifiers, such as decision trees, discriminant analysis, logistic regression, naive Bayes, support vector machine, and neural networks, perform in the classification.

In this research, the optimal learning model shown in Equation (3.20) can be established based on the minimum mean squared error to train the model parameters, such as referential values and weights. The mean squared error (MSE) is used to measure the difference between the predicted probabilities and the observed values of the classes of the output variable.

$$\min \ \delta = \frac{1}{N*K}\sum_{n=1}^{N}\sum_{k=1}^{K}(p_{k,n} - \hat{p}_{k,n})^2 \qquad (3.20)$$

$$s.t. r_{i_l,j_l}^{k}, w_{i_l,j_l}^{k}, \gamma_{(i_1,j_1),(i_2,j_2)}^{k} \in \Omega$$

where $\hat{p}_{k,n}$ $(\mathrm{k} = 1,\dots,\mathrm{K}, \mathrm{n} = 1,\dots,\mathrm{N})$ is the probability that the assertion pointing to the $k^{th}$ class membership is true for the $n^{th}$ observation. $p_{k,n}$ is the predicted probability for the $k^{th}$ class membership. Ω is the feasible space of the parameters, and all the parameters should satisfy certain constraints included in Ω. Equation (3.20) corresponds to Equation (5.9) which is a general form of optimal learning model. Numerical examples are given in Sections 5.6 and 6.3 to illustrate how the

approach of rule-based inferential modelling and prediction introduced in this chapter can be applied for the classification of data sets.

## 3.7 An Adapted Single-level Genetic Algorithm for Problems of Bilevel Optimisation

In the models based on the system of the MAKER framework for classification, there are two levels of parameters: the referential values of the observed values of the input variables of the data sets for classification, and the weights (reliabilities) of these referential values under different classes of observed values of the output variables of the data sets. In the models based on the system of the MAKER framework for functions approximation, there are two levels of parameters: the referential values of the observed values of the data sets for functions approximation and the weights (reliabilities) of the referential values of the observed values of the input variables of the data sets under different referential values of the observed values of the output variables of the data sets. The referential values of the observed values of the data sets for functions approximation include the referential values of the observed values of the input variables of the data sets and the referential values of the observed values of the output variables of the data sets. Because the weights (reliabilities) are set up for the referential values of input variables under different classes or referential values of the output variables, the referential values should be decided before the assignment of weights (reliabilities) to build the model. In other words, the optimisation of weights is nested within the optimisation of referential values. Such kind of optimisation is referred to as bilevel optimisation. According to Sinha, Malo, and Deb (2018), bilevel optimisation is a mathematical program in which an optimisation problem is nested within another optimisation problem. As stated by Sinha, Malo, and Deb (2016), the nested structure of bilevel optimisation may introduce some difficulties such as non-convexity, non-linearity, discreteness, and

non-differentiability. Because of these difficulties, classical algorithms may not work effectively to provide optimal solutions to complex bilevel optimisation problems.

As discussed by Vikhar (2016), an evolutionary algorithm is a subset of evolutionary computation and a generic population-based metaheuristic optimisation algorithm in the field of artificial intelligence. The most popular type of evolutionary algorithm is the genetic algorithm. Gen and Cheng (1997) concluded that there are three major advantages to applying the genetic algorithm to optimisation problems. First, the genetic algorithm does not have many mathematical requirements about optimisation problems, and the genetic algorithm can handle any type of objective function and any type of constraint (i.e., linear or nonlinear) defined on discrete, continuous, or mixed search spaces. Second, the ergodicity of evolution operators makes genetic algorithms very effective at performing a global search. Third, genetic algorithms provide us with great flexibility to hybridise with domain-dependent heuristics to achieve efficient implementation for a specific problem. According to Sinha, Malo, and Deb (2018), nested evolutionary algorithms are a popular method to address bilevel optimisation problems, where the lower level optimisation problem is solved corresponding to each upper level member. Nested strategies are effective but very computationally demanding and not suitable for large-scale bilevel optimisation problems.

The genetic algorithm, as an evolutionary algorithm, possesses strong robustness and good global search capability. For this reason, our solution to the bilevel optimisation problem is using an adapted single-level genetic algorithm to offset some of the difficulties mentioned previously. In this adapted genetic algorithm for the optimisation of the models based on the MAKER framework, the initial population of individuals is generated randomly in the ranges of the observed input values of the input variables of a data set. The population of individuals (chromosomes) is composed of 10 subpopulations, and each subpopulation

contains 20 individuals. In the optimisation of the models based on the system of the MAKER framework for classification, each individual (chromosome) of the population consists of both the referential values of input variables of a data set for classification and the weights (reliabilities) of these referential values for different classes of the output variables of the data set. Figure 3.1 presents an example of such an individual used in the adapted genetic algorithm.


**Figure 3.1 The Individual (Chromosome) of the Population Used in the Adapted Genetic Algorithm**

In Figure 3.1, $A_j^i$ indicates the $j^{th}$ referential value of the $i^{th}$ input variable, and $w_{i,j}^k$ represents the weight of the $j^{th}$ referential value of the $i^{th}$ input variable for the $k^{th}$ class of the output variable. These parameters are organized in the form shown in Figure 3.1 to facilitate parallel implementation of computation which can improve computation speed of objective values of individuals.

In the optimisation of the models based on the MAKER framework for functions approximation, each individual in the population contains both the referential values of the observed values of a data set for functions approximation and the weights (reliabilities) of the referential values of the input variables of the data set under different referential values of the output variables of the data set. After the generation of the initial population, the objective function value is calculated for each individual (solution) in the population. It is worth nothing that all the calculations of the objective function values in this adapted genetic algorithm are implemented using parallel computing. Then, on the basis of the population of the individuals stated above, a series of genetic algorithm operations, that is, selection, recombination, mutation, reinsertion, and migration, is performed iteratively to

obtain an optimised solution for referential values and weights. Specifically, in each iteration of the genetic algorithm operations stated above, the individuals (solutions) of each subpopulation are ranked in ascending order of the objective function values of these individuals. Afterwards, each individual (solution) of each subpopulation is assigned a fitness value by the principle that smaller objective values would be assigned larger fitness values. Then, the fittest individuals of each subpopulation, which have the smallest objective function values, would be selected by the method of stochastic universal sampling to breed a new generation of population. To generate a new generation of population, discrete recombination and real-value mutation are performed successively on the selected individuals. After recombination and mutation, the objective function value is calculated for each of the selected individuals. Then the best individuals out of the abovementioned selected ones by the fitness-based method, which have the smallest objective function values, are inserted into the population of the last generation to generate a new generation of population. It is noteworthy that elite individuals would migrate between subpopulations every 20 iterations. After a fixed number (e.g., 200) of iterations, the iterative process of genetic algorithm operations, as already stated, would be terminated to obtain an optimised solution. Compared with the nested evolutionary algorithm, our proposed adapted genetic algorithm, as an evolutionary algorithm, is robust and less computationally demanding. Additionally, our proposed adapted genetic algorithm is more suitable for large-scale bilevel optimisation problems and parallel computing than nested evolutionary algorithms because each individual (chromosome) in the adapted genetic algorithm contains both referential values and weights.

Based on the above-mentioned description, the adapted single-level genetic algorithm is summarised as follows.

**Algorithm 1:** The Adapted Single-level Genetic Algorithm

**Input:** The initial parameters for the genetic algorithm, e.g., number of referential

values, crossover rate, mutation rate, insertion rate, migration rate, number of subpopulation, number of individuals in each subpopulation, number of generations between migration (miggen), and maximum number of iterations (maxiteration).

**Output:** The optimised individual containing both optimised referential values and optimised weights.

1: **GENERATE** an initial population.

2: **CALCULATE** the objective function values of individuals of initial population.

3: **FOR** iteration = 1 to maxiteration

4: **RANK** the individuals in each subpopulation based on their objective function values.

5: **ASSIGN** the fitness values to these individuals.

6: **SELECT** the individuals according to their fitness values.

7: **PERFORM** discrete recombination over the selected individuals.

8: **PERFORM** real-value mutation over the selected individuals.

9: **CALCULATE** the objective function values of the selected individuals.

10: **REINSERT** the best individuals out of the selected individuals into the population of the last generation to generate a new generation of population.

11: **IF** REM(iteration, 20)=0 **THEN**

12: Elite individuals **MIGRATE** between subpopulations.

13: **ENDIF**

14:**ENDFOR**

15:**RETURN** the best individual.

It is noteworthy that in Algorithm 1, just before each of steps 2, 7, 8, and 9, the referential values of each individual are sorted in an ascending order from the leftmost position to the rightmost one of the referential values. This is due to two reasons: firstly, in the code for the calculation of the objective values of individuals, the referential values are designed to be in an ascending order from the leftmost position to the rightmost one of the referential values, and secondly, the operations

of recombination and mutation may lead to a non-ascending order for the referential values, which may affect the normal functioning of the subsequent operations.

## 3.8 Summary

In this chapter, we presented the research methodologies used in Chapters 4, 5, and 6 of this thesis. First, we justified the choice of functions used in the functions approximation and the choice of the classical data sets for classification and briefly introduced the original sepsis data set, which will be processed in Chapter 6 for sepsis diagnosis. Then, we briefly described the original ER rule, which is the theoretical foundation of the MAKER framework. Subsequently, we elaborated on the MAKER framework, which is a major research method employed in Chapters 4, 5, and 6 of this thesis. Finally, we illustrated the adapted genetic algorithm for bilevel optimisation used in the functions approximation and the classification of the classical data sets and sepsis data set.

# Chapter 4

# Referential-value-based Data Discretization Techniques

## 4.1 Introduction

This chapter is focused on referential-value-based data discretization techniques for transforming continuous data. The continuous functions approximations are used to check whether the approach of the rule-based inferential modelling and prediction based on the referential-value-based data discretization techniques can be used to well approximate all kinds of continuous functions. Continuous functions approximation is generally connected with classification, and classification can be considered as a simplified version of continuous functions approximation. Section 5.6.1 is focused on the correlation between continuous functions approximation and classification. From the continuous functions approximations, we can have the knowledge about how many referential values are suitable for an accurate approximation using MAKER-based model to a continuous function, in order to avoid the problem of underfitting or overfitting. Based on the knowledge, we can develop stopping criteria to guide the training process of the MAKER-based model for continuous functions approximation. As there is a connection between continuous functions approximation and classification, we can further have adapted stopping criteria to guide the training process of MAKER-based models for classification, according to the ones generated from continuous functions approximations. Hence, this chapter lays foundation for Chapters 5 and 6 in which we use the approach of rule-based inferential modelling and prediction to establish MAKER-based models for classification.

The remainder of this chapter is organized as follows. Section 4.2 performs a comparative analysis between mainstream data discretization techniques and referential-value based data discretization techniques. Sections 4.3 and 4.4 present, respectively, the univariate and the bivariate functions approximations using MAKER framework. In Section 4.5, new stopping criteria for the model training process is proposed. A summary of this chapter is provided in Section 4.6.

## 4.2 Comparative Analysis between Data Discretization Techniques

In general, data can be divided into qualitative data and quantitative data (Maimon and Rokach, 2005). Quantitative data can be further divided into two types: discrete data and continuous (Maimon and Rokach, 2005). Quantitative data are often involved in data mining applications, but learning from quantitative data is generally less efficient and less effective than that from qualitative data (Maimon and Rokach, 2005). We can use data discretization techniques, often used to transform one data type to another, to transform quantitative data to address this issue (Maimon and Rokach, 2005).

The data used in this research are mainly 'continuous data'. A variety of data discretization methods exist for transforming continuous data. Most discretization methods can be classified into primary and composite (Maimon and Rokach, 2005). Primary methods do not rely on any other discretization methods, while composite methods are based on primary methods (Maimon and Rokach, 2005). Primary discretization methods can be further classified into supervised and unsupervised: The former utilizing class information from training observations to determine cut-off points for discretization, the latter not using such information (Dougherty, Kohavi and Sahami, 1995).

Typical unsupervised discretization methods include equal-width discretization, equal-frequency discretization, and fixed-frequency discretization (Maimon and Rokach, 2005). In the equal-width discretization (Catlett, 1991; Kerber, 1992; Dougherty, Kohavi and Sahami, 1995), a predefined number k is used to divide the observations between the minimum observed value $v_{min}$ and the maximum observed value $v_{max}$ into k intervals of equal width. Thus, each interval has the width $w = \frac{v_{max} - v_{min}}{k}$ and the cut-off points are located at $v_{min} + w, v_{min} + 2w, \ldots, v_{min} + (k-1)w$. In the equal-frequency discretization (Catlett, 1991; Kerber, 1992; Dougherty, Kohavi and Sahami, 1995), a predefined number k is used to divide the sorted observed values into k intervals so that there is approximately the same number of training observations in each interval. In the fixed-frequency discretization (Yang and Webb, 2008), a sufficient interval frequency m is set to divide the sorted observations into a number of intervals, so that all the intervals have approximately the same number m of training observations with adjacent values.

As mentioned previously, unlike unsupervised learning, the supervised methods involve taking advantage of class information from training observations to determine the cut-off points for the discretization (Dougherty, Kohavi and Sahami, 1995). According to Fayyad and Irani (1993), multi-interval-entropy-minimization discretization (MIEMD) is a typical supervised discretization method. In MIEMD, the midpoint between each successive pair of sorted observed values is a candidate cut-off point for discretizing the observed values of an input variable (Fayyad and Irani, 1993). Each candidate cut-off point is used to divide the observed values into two intervals and the resulting entropy of the class information is calculated for each candidate cut-off point (Fayyad and Irani, 1993). The candidate cut-off point for which the entropy is minimal among all the candidates is selected as a cut-off point for a binary discretization (Fayyad and Irani, 1993). The binary discretization

is applied recursively to pick out optimal cut-off points until a certain criterion is satisfied (Fayyad and Irani, 1993).

In contrast to equal-width discretization, equal-frequency discretization, and fixed-frequency discretization, MIEMD is a form of hierarchical discretization, involving a split procedure (Maimon and Rokach, 2005). A merged method of hierarchical supervised discretization is ChiMerge, which uses the $\chi^2$ statistic to decide whether the relative class frequencies of adjacent intervals are significantly different or similar enough to be merged into a single interval (Kerber, 1992). The StatDisc discretization (Richeldi and Rossotto, 1995), another merged method of hierarchical supervised discretization, is an extended version of ChiMerge that allows any number of intervals to be merged rather than just two as in the discretization method of ChiMerge. InfoMerge (Freitas and Lavington, 1996) is another merged method of hierarchical supervised discretization that uses information loss to guide the merge procedure.

The above-mentioned discretization methods are the most commonly used in research. However, these discretization methods inevitably have limitations. This is due to the fact that all of them use cut-off points to discretize training observations into intervals, which naturally leads to information loss and distortion. For example, according to Reinartz (1999), the major disadvantage of equal-width discretization is the possibility of generating imbalanced intervals, some containing many training observations and others only a small number.

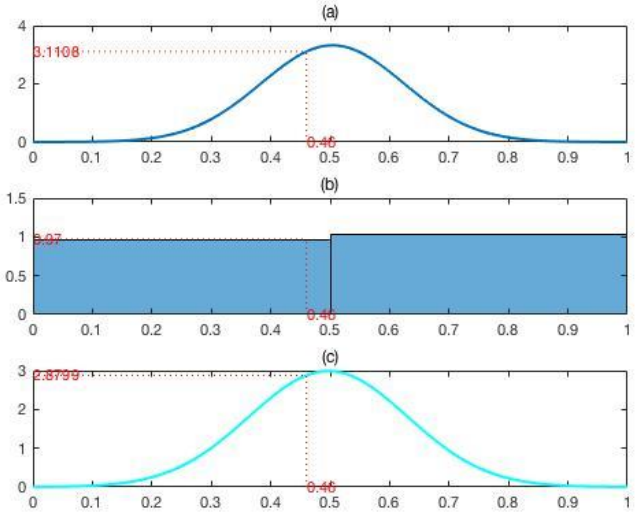



**Figure 4.1 Comparison between different data discretization techniques**

Another example is provided in Figure 4.1 in which (a) shows a probability density function curve of the input values between x=0 and x=1, (b) the corresponding histogram of the normalized frequencies of the input values based on the equal-width discretization, and (c) the corresponding probability density function curve of the input values based on the referential-value-based discretization. The y-coordinate values in the subfigures represent the probability densities of the inputs. The cut-off point in (b) and the referential value between the minimum and maximum input values in (c) are both taken as 0.5. We can see clearly that in (b), the probability densities of the bin between x=0 and x=0.5 are all 0.5. In (c), the probability densities generated by the referential-value-based discretization method for the input values between x=0 and x=0.5 generally change with the input values between x=0 and x=0.5. Specifically, if an input value is 0.46, the corresponding probability density generated by the equal-width discretization method for this input value is 0.97 as shown in (b), and the probability density generated by the referential-value-based discretization method for this input value is 2.8799 in (c), which is much closer to the frequency density: 3.1108 for this input value.

Additionally, in the referential-value-based discretization, the referential values between the minimum and maximum observed values of each input variable of a training data set could be optimized using the adapted genetic algorithm introduced in Section 3.4 to minimize the difference between the observed output values of the training set and the predicted output values of the MAKER-based model. Hence, the referential-value-based data discretization method is essentially a supervised discretization method. Overall, from what has been analysed above, it can be deduced that the referential-value-based data discretization method is better at reducing information loss and distortion, and better at presenting the characteristics of data, than the mainstream data discretization methods.

## 4.3 Univariate Functions Approximations

A Belief Rule Based (BRB) system is generally a distributed approximation process (Chen et al., 2013) in which belief rules, belief degrees of consequents of belief rules, and rule weights can be trained. The maximum likelihood evidential reasoning (MAKER) framework is also a distributed approximation process in which the evidence and weight of each evidential element that exactly points to an assertion in the state space can be trained. However, in the MAKER framework, the belief degree of each evidential element is acquired from statistical analysis on the basis of trained evidence rather than being trained by an optimal learning method. From the perspective of extracting useful information from data, the MAKER framework is more realistic and effective than a BRB system as the former acquires belief degrees of evidential elements through statistical analysis from data directly while the latter employs the optimal learning method to train the belief degrees of consequents of belief rules.

The capability of MAKER framework to approximate functions is explored in the following part of this section. We start with univariate functions to demonstrate this approximation power.

For the approximations of univariate functions, observed input-output data pairs need to be generated from functions. For example, given the function $y = 2^x$, for an input value (x), e.g., x=0.5, its corresponding output value (y) is $2^{0.5} = 1.4142$. Thus, x=0.5 and y=1.4142 form a data pair (0.5,1.4142). All other data pairs for function approximation can be generated by the same method. All data pairs generated by this method then form a data set for function approximation. The data set for function approximation includes the observed input values (x) of a function and the observed output values (y) of the function. With all of these observed input-output data pairs, we can use rule-based inferential Modelling and prediction to establish models, and use the adapted single-level genetic algorithm to train the parameters, e.g., referential values and weights of the model for function approximation based on the MAKER framework, by minimizing the differences between the observed output values (y) of the function and the predicted output values generated by the model for the function.

The approximation of univariate functions by a MAKER-based model involves two stages: initial learning and advanced learning. In the initial learning, the x-coordinate referential values and the y-coordinate referential values are fixed as the minima and maxima of the observed input values (x) and the global extrema of the observed output values (y) in the data set, respectively. In the advanced learning, the x-coordinate referential values and y-coordinate referential values comprise, not only the above minima and maxima, but also the trained referential values between the minima and maxima, to improve the model capacity for the approximation of complex functions. The advanced learning for each function has a group of approximations. Different approximations have different combinations of numbers of x-coordinate and y-coordinate referential values. In each approximation, we use the approach of rule-based inferential modelling and prediction to establish MAKER-based models based on a fixed number of x-coordinate and y-coordinate referential values, and we use the adapted single-level genetic algorithm to train the referential values and the relevant weights. In the

process of advanced learning, the number of x-coordinate referential values is increased while that of y-coordinate referential values keeps unchanged, and so does the number of y-coordinate referential values while that of x-coordinate referential values keeps unchanged. After the training of models in each approximation is finished, we can obtain a value of MSE (Mean Squared Error) to measure the difference between predicted values of a model and observed output values of a function. In other words, we can obtain an MSE for each approximation. Based on these MSEs, stopping criteria will be developed for terminating the process of advanced learning to have a model with a balance between model accuracy and model complexity. The stopping criteria will be illustrated in Section 4.5.

The processes of initial learning and advanced learning are summarized in the following groups of steps. It is noteworthy that only the referential values between the minima and maxima of observed input or output values of a function are trained in the univariate functions approximation including initial learning and advanced learning. Therefore, we do not need to train any referential values in the initial learning, as the referential values are fixed as the minima and maxima of observed input and output values of a function in the initial learning. This is well designed in the codes for implementation of initial and advanced learning.

It is also worth noting that the observed input and output values for the prediction in both initial learning and advanced learning are the observed input and output values of a function mentioned on Page 103 and in each function approximation of Sections 4.3.1 and 4.3.2, and the predicted outputs of the MAKER-based models are the probabilities of observed input values or x-coordinate referential values for different referential values of output variable or y-coordinate referential values (equivalent to the classes of output variable in classification). The sum of all the referential values of output variable or y-coordinate referential values multiplied

with their relevant probabilities can then be used as the predicted output values of the MAKER-based models for functions approximation.

The steps of initial learning are displayed in the following part.

Step 1: Generating the data set for a function approximation.

Step 2: The x-coordinate and y-coordinate referential values are fixed as the minima and maxima of observed input and output values of a function.

Step 3: Using the approach of rule-based inferential modelling and prediction to establish a MAKER-based model on the basis of the referential values for function approximation.

Step 4: Using the single-level adapted genetic algorithm to train the relevant weights of referential values to get the optimized referential values and weights.

Step 5: Generating the predicted output values for a function on the basis of the MAKER-based model of optimized referential values and weights.

Step 6: Calculating the mean squared error (MSE) between the observed and the predicted output values for a function.

The steps of advanced learning are shown in the following part.

Step 1: Generating the data set for a function approximation.

Step 2: Each group of function approximations of advanced learning consists of a number of approximations. Different approximations have different combinations of numbers of x-coordinate and y-coordinate referential values, and in each approximation, the numbers of x-coordinate and y-coordinate referential values are fixed, but the referential values between minima and maxima of observed input or output values are not fixed, and they can be trained using adapted genetic algorithm. Among all the approximations of a group of function approximations of advanced learning, both the numbers of trained x-coordinate and y-coordinate referential values (referential values between minima and maxima of observed input or output values) can be increased from 0 to a certain number. The number of trained x-coordinate or y-coordinate referential values is increased in such a way

that the number of one type of the trained x-coordinate and y-coordinate referential values is increased from 0 to a certain number while the number of the other type keeps unchanged. For example, in the group of function approximations of advanced learning for $y = \log_6 x$, the number of trained y-coordinate referential values can be increased from 0 to 4, while the number of trained x-coordinate referential values can be kept at 1, as we move along the dimension of number of trained y-coordinate referential values. The certain number is the maximum number that trained x-coordinate or y-coordinate referential values can be in a group of function approximations, and it is designed depending on what the function is to check how many referential values are enough to well approximate a function. For instance, we select 5 as the maximum number that the trained x-coordinate referential values can be in the groups of function approximations of advanced learning for monotonic functions, i.e., $\mathrm{y} = 6^x$, $y = \log_6 x$, and $y = x^{\frac{1}{6}}$, and simple non-monotonic function, i.e., $y = -(x - 0.5)^2 + 0.25$ and 4 as the one that the corresponding trained y-coordinate referential values can be. This maximum number becomes 10 for the trained x-coordinate referential values and 6 for the trained y-coordinate referential values respectively, in the group of function approximations of advanced learning for complex non-monotonic function, i.e., $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$.

Step 3: In each approximation of a group of function approximations of advanced learning, we use the referential values and weights of each individual of population used in the single-level adapted genetic algorithm to establish a MAKER-based model, and use the single-level adapted genetic algorithm to train the referential values between minima and maxima of observed input or output values and relevant weights of MAKER-based model, to obtain the optimized set of referential values and weights.

Step 4: Generating the predicted output values for a function based on the model of optimized referential values and weights for each experiment of a function approximation.

Step 5: Calculating the mean squared error (MSE) between the observed and the predicted output values for a function for each experiment of a function approximation.

### 4.3.1 Initial Learning

For the initial learning, four common types of functions are used as examples to demonstrate the approximation capability MAKER-based models framework for different types of functions: exponential functions $\mathrm{y} = a^x$, logarithmic functions $y = \log_a x$, power functions $y = x^a$, and the function $y = -(x - 0.5)^2 + 0.25$. Each type of function has its own characteristics and is used to represent similar functions. Specifically, the four types just listed are used to represent convex functions, concave functions, functions whose mean curvatures are large, and non-monotonic univariate functions, respectively.

Each type of function, as mentioned above, can present one or more specific functions. Each specific function is used to generate a data set of observed input-output data pairs for the approximation of this specific function. The observed input values (x) of a function in the data sets are distributed uniformly and the interval between any two adjacent observed input values (x) of a function is set to 0.01. The values on the x axes of Figures 4.2

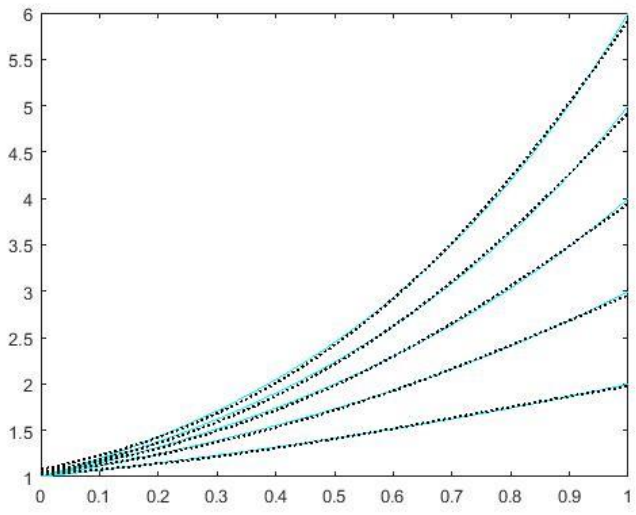



**Figure 4.2 Initial learning for functions $\mathrm{y} = a^x$ for MAKER-based models**

through 4.5 illustrating initial learning represent the observed input values (x) and the values on the y axes of following figures about initial learning represent both the observed output values (y) and the predicted output values of a MAKER-based model. The cyan cyan-coloured curves represent the observations of the functions and the black dotted curves represent the predictions of the functions made by the MAKER-based models.

In Figure 4.2, a collection of exponential functions $\mathrm{y} = a^x$ (a=2, 3, 4, 5, and 6), is used to generate data sets to approximate the exponential functions. From these data sets, MAKER-based models are built to perform initial learning. The mean squared errors (MSEs) between the observed output values and the predicted output values are $1.60 * 10^{-4}, 4.91 * 10^{-4}$, $7.81 * 10^{-4}$, $6.58 * 10^{-4}$, and $1.07 * 10^{-3}$ for the five functions $\mathrm{y} = 2^x$, $\mathrm{y} = 3^x$, $\mathrm{y} = 4^x$, $\mathrm{y} = 5^x$, and $\mathrm{y} = 6^x$ respectively. From Figure 4.2 and the MSEs, it is evident that the exponential functions $\mathrm{y} = a^x$ are very well approximated by the MAKER-based models, whose parameters are optimally trained as shown in Figure 4.2.

**Table 4.1 Trained weights (w) of x-coordinate referential values (x-rv) for different y-coordinate referential values (y-rv) for $\mathrm{y} = a^x$**

| Approx. | w of x-rv 0 | | w of x-rv 1 | |
|---|---|---|---|---|
| | under y-rv no.1 (minimum) | under y-rv no.2 (maximum) | under y-rv no.1 (minimum) | under y-rv no.2 (maximum) |
| $\mathrm{y} = 2^{\mathrm{x}}$ | 0.5493 | 0.0388 | 0.0266 | 0.4012 |
| $\mathrm{y} = 3^{\mathrm{x}}$ | 0.5757 | 0.0337 | 0.0207 | 0.3555 |
| $\mathrm{y} = 4^{\mathrm{x}}$ | 0.5764 | 0.0263 | 0.0155 | 0.3148 |
| $y = 5^{\mathrm{x}}$ | 0.5100 | 0.0161 | 0.0084 | 0.2529 |
| $\mathrm{y} = 6^{\mathrm{x}}$ | 0.5714 | 0.0203 | 0.0083 | 0.2628 |

Table 4.1 displays the weights for the x-coordinate referential values for the different y-coordinate referential values, for the initial learning of the MAKER-based models for functions $\mathrm{y} = a^x$. From Table 4.1, it can be seen that the weights of x-

coordinate referential values 0 and 1 for the y-coordinate referential values have a direct influence on the approximation for functions $\mathrm{y} = a^x$ in the MAKER-based model. The weights of x-coordinate referential value 0 for y-coordinate referential value no.1 (the minimum of the observed y-coordinate values in the data set) fluctuate between 0.5 and 0.6. Meanwhile, the weights of x-coordinate referential value 1 for y-coordinate referential value no.2 (the maximum of the observed y-coordinate values in the data set) are generally decreasing as the function approximated by the MAKER-based model is changed from $y = 2^x$ to $y = 6^x$. The weights of x-coordinate referential value 0 for y-coordinate referential value no.2 and those of x-coordinate referential value 1 for y-coordinate referential value no.1 play a small role in the approximation of the functions $\mathrm{y} = a^x$, as they are generally very close to 0. This suggests that we use only a few parameters to train the models to approximate functions $\mathrm{y} = a^x$ with a small error.

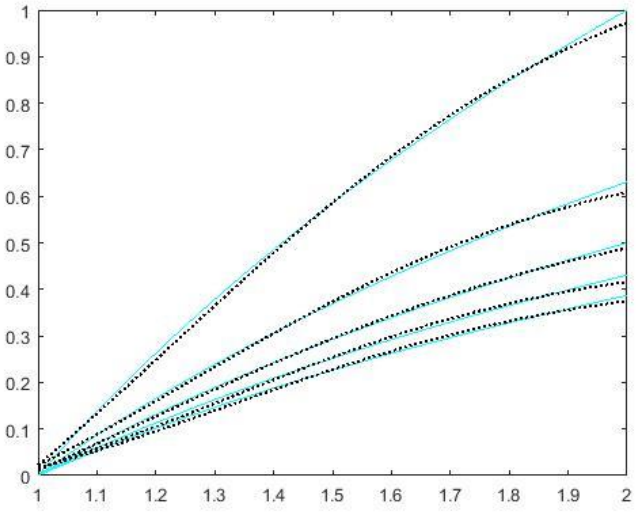



**Figure 4.3 Initial learning for functions $y = \log_a x$ for MAKER-based models**

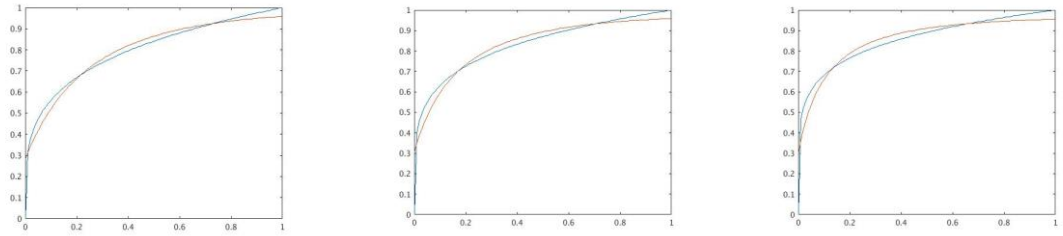
In Figure 4.3, observed input-output data pairs for the approximation of logarithmic functions $y = \log_a x$. Again, MAKER-based models are built to perform the initial learning. The MSEs for this collection of functions $\mathrm{y} = \log_a x$ (a=2, 3, 4, 5, and 6) are $1.07 * 10^{-4}$, $3.37 * 10^{-5}$, $4.13 * 10^{-5}$, $1.65 * 10^{-5}$, and $2.40 * 10^{-5}$ respectively. Figure 4.3 and the relevant MSEs verify that these MAKER-based models also have superior approximation capability for logarithmic functions $y = \log_a x$.

**Table 4.2 Trained weights (w) of x-coordinate referential values (x-rv) for y-coordinate referential values (y-rv) for $y = \log_a x$**

| Approx. | w of x-rv 1 | | w of x-rv 2 | |
|---|---|---|---|---|
| | under y-rv no.1 (minimum) | under y-rv no.2 (maximum) | under y-rv no.1 (minimum) | under y-rv no.2 (maximum) |
| $y = \log_2 x$ | 0.2981 | 0.0166 | 0.0200 | 0.4063 |
| $y = \log_3 x$ | 0.2629 | 0.0129 | 0.0164 | 0.3608 |
| $y = \log_4 x$ | 0.3686 | 0.0236 | 0.0348 | 0.5050 |
| $y = \log_5 x$ | 0.2674 | 0.0131 | 0.0191 | 0.3692 |
| $y = \log_6 x$ | 0.3647 | 0.0249 | 0.0321 | 0.4963 |

Table 4.2 exhibits the trained parameters, i.e., the weights for the x-coordinate referential values for various y-coordinate referential values in the MAKER-based models. The ratios of the weights of x-coordinate referential value 1 for y-coordinate referential value no.1 to the corresponding weights of x-coordinate referential value 2 for y-coordinate referential value no.2 in this case, hover around 0.73, as the shapes of the curves used for the approximation of the functions $y = \log_a x$ are generally similar to each other. At the same time, the weights of x-coordinate referential value 1 for y-coordinate referential value no.1 and those of x-coordinate referential value 2 for y-coordinate referential value no.2 have little impact on the approximation, as they generally approach 0. All this indicates that we can use a limited number of parameters to train the MAKER-based models to approximate functions of the form $y = \log_a x$ with high accuracy.

Next, a collection of power functions $y = x^{1/4}$, $y = x^{1/5}$, and $y = x^{1/6}$, are used in the same way as above in Figure 4.4. The cyan solid curve represent the observations of the function and the red solid curve represents the predictions of the MAKER-based models. The MSEs are 0.001421, 0.001687, and 0.001836 respectively.



y=$x^{\frac{1}{4}}$ y=$x^{\frac{1}{5}}$ y=$x^{\frac{1}{6}}$

**Figure 4.4 Initial learning for functions $y = x^a$ for MAKER-based models**

Table 4.3 displays the trained parameters, i.e., the weights for the x-coordinate referential values for various y-coordinate referential values, in the initial learning for functions $y = x^a$ from the MAKER-based models. The ratios of the weights of x-coordinate referential value 1 and y-coordinate referential value no.2 to the corresponding weights of x-coordinate referential value 0 and y-coordinate referential value no.1 are generally diminishing as the function being approximated by the MAKER-based models changes from $y = x^{1/6}$ to $y = x^{1/4}$, as the mean curvatures of those functions is generally diminishing.

**Table 4.3 Trained weights (w) of x-coordinate referential values (x-rv) under y-coordinate referential values (y-rv) for $y = x^a$**

| Approx. | w of x-rv 0 | | w of x-rv 1 | |
|---|---|---|---|---|
| | under y-rv no.1 (minimum) | under y-rv no.2 (maximum) | under y-rv no.1 (minimum) | under y-rv no.2 (maximum) |
| $y = x^{1/4}$ | 0.1247 | 0.0863 | 0.0672 | 0.7448 |
| $y = x^{1/5}$ | 0.0732 | 0.0536 | 0.0640 | 0.6713 |
| $y = x^{1/6}$ | 0.0337 | 0.0236 | 0.0415 | 0.4527 |

From Figures 4.2, 4.3, and 4.4 and the relevant MSEs in Tables 4.1, 4.2, and 4.3, it can be observed that the approximations of the power functions $y = x^a$ are generally worse than those of both the exponential functions $y = a^x$ and the logarithmic functions $y = \log_a x$. This is mainly due to the fact that the mean curvatures of the power functions are generally larger than those of the latter two

and the fact that there are only four referential values (i.e., the minimum and the maximum of the input values and the global extrema of the output values) of functions in the $y = x^a$ case. This suggests more referential values should be used to approximate functions with large mean curvatures.

As demonstrated above, the MAKER framework can be used to accurately approximate monotonic univariate functions with moderate mean curvatures, e.g., exponential functions $y = a^x$ and logarithmic functions $y = \log_a x$, even if there are only four referential values in the approximation. However, four referential values are apparently not enough for the MAKER framework to accurately approximate monotonic univariate functions with large mean curvatures, e.g., power functions $y = x^a$. Extra referential values would be needed in such cases.

**Table 4.4 Trained weights (w) of x-coordinate referential values (x-rv) under y-coordinate referential values (y-rv) for $y = -(x - 0.5)^2 + 0.25$**

| Approx. | w of x-rv 0 | | w of x-rv 1 | |
|---|---|---|---|---|
| | under y-rv no.1 (minimum) | under y-rv no.2 (maximum) | under y-rv no.1 (minimum) | under y-rv no.2 (maximum) |
| $y = -(x - 0.5)^2 + 0.25$ | 0.0719 | 0 | 0.3326 | 1 |

Initial learning for monotonic univariate functions based on the MAKER framework has been demonstrated and discussed above in detail. Next, it is necessary to address the initial learning for non-monotonic univariate functions. Hence, the function $y = -(x - 0.5)^2 + 0.25$ is used as an example. Again, the function is utilized to generate a data set of observed input-output data pairs for approximation purposes. Using this data set, initial learning is performed by building MAKER-based models. The MSE calculated for the data set of this function is 0.0042.

Table 4.4 exhibits the trained parameters, i.e., the weights for the x-coordinate referential values and y-coordinate referential values, in the abovementioned initial

learning. From Figure 4.5 and the relevant MSEs in Table 4.4, it can be observed that there is a significant difference between the solid cyan curve representing the observations of the function $y = -(x - 0.5)^2 + 0.25$ and the red solid curve representing the predictions of the MAKER-based model, which suggests the initial MAKER-based model with only four referential values (minimum and maximum input values and global extrema of output values) is unable to properly approximate this non-monotonic univariate function. Therefore, it appears necessary to perform advanced learning for non-monotonic univariate functions.

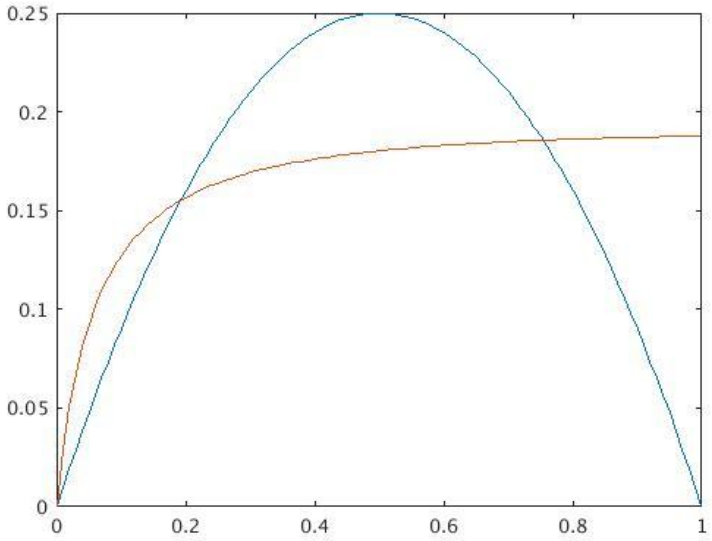



**Figure 4.5 Initial learning for the function $y = -(x - 0.5)^2 + 0.25$ for MAKER-based model**

### 4.3.2 Advanced Learning

As previously mentioned, with the initial learning for monotonic univariate functions using MAKER-based models, the x-coordinate and y-coordinate referential values are fixed as the minimum and maximum of the observed input values (x) of the function in the data set and the global extrema of the observed output values (y)

of the function in the data set, respectively, and we only need to train the weights of the x-coordinate referential values given each y-coordinate referential value.

With the advanced learning, both the x-coordinate and y-coordinate referential values can be trained to further improve the approximation capability of the MAKER framework. The advanced learning is applied to each type of functions used to present the initial learning above. A typical function is selected from each type of functions and it is approximated as an example to demonstrate advanced learning for this type of functions. Specifically, the functions $\mathrm{y} = 6^x$, $y = \log_6 x$, $y = x^{\frac{1}{6}}$, and $y = -(x - 0.5)^2 + 0.25$ are approximated as examples of advanced learning for exponential functions $\mathrm{y} = a^x$, logarithmic functions $y = \log_a x$, power functions $y = x^a$, and non-monotonic univariate functions $y = -(x - a)^2 + b$ respectively. Additionally, the approximation of a multi-extremal function $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$, is used as an example of advanced learning for complex non-monotonic univariate functions.

Each of these typical functions is utilized to generate a data set of observed input-output data pairs. In these data sets, the observed input values (x) of the function are generated uniformly with the interval between any two adjacent observed input values set to 0.01 (0.05 for the function $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$). From the data set of observed input-output data pairs, advanced learning is performed by building MAKER-based models.

With the advanced learning of MAKER-based models, the referential values of the observed input values (x-coordinates) include the minimum and maximum values in the data set, and the values between the minimum and maximum. The referential values of the observed output values (y-coordinates) encompass the global extrema and the values between them.

In the Figures 4.6 through 4.10 that illustrates advanced learning, the values on the x-axes represent the observed input values (x) of the functions in the data sets and the values on the y-axes represent the observed output values (y) and the predicted output values of the MAKER-based models. The solid cyan curves and the solid red curves represent the observations and the predictions respectively. The red points on the y-axes and magenta points on the x-axes indicate the trained y-coordinate and x-coordinate referential values respectively.

Tables 4.5 through 4.9 show the MSEs for the advanced-learning MAKER approximations with different numbers of trained x-coordinate and y-coordinate referential values. The 'trained x-coordinate referential values' and the 'trained y-coordinate referential values' used in the following part of this section are short for the trained referential values of observed input values (x) between the minimum and the maximum of the observed input values (x) of the function in the data set and the trained referential values of observed output values (y) between the global extrema of observed output values (y) of the function in the data set respectively.

**Table 4.5 MSEs for approximations with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values (advanced learning case) for** $\mathrm{y} = 6^{x}$

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0.001067 | 0.001635 | 0.002931 | 0.000724 | 0.000826 | 0.001005 |
| 1 | 0.000886 | 0.000956 | 0.000860 | 0.000876 | 0.000887 | 0.000696 |
| 2 | 0.000784 | 0.000912 | 0.000827 | 0.000811 | 0.000799 | 0.000646 |
| 3 | 0.000602 | 0.000796 | 0.001047 | 0.000941 | 0.000926 | 0.000575 |
| 4 | 0.000225 | 0.001057 | 0.000528 | 0.000593 | 0.000659 | 0.000531 |

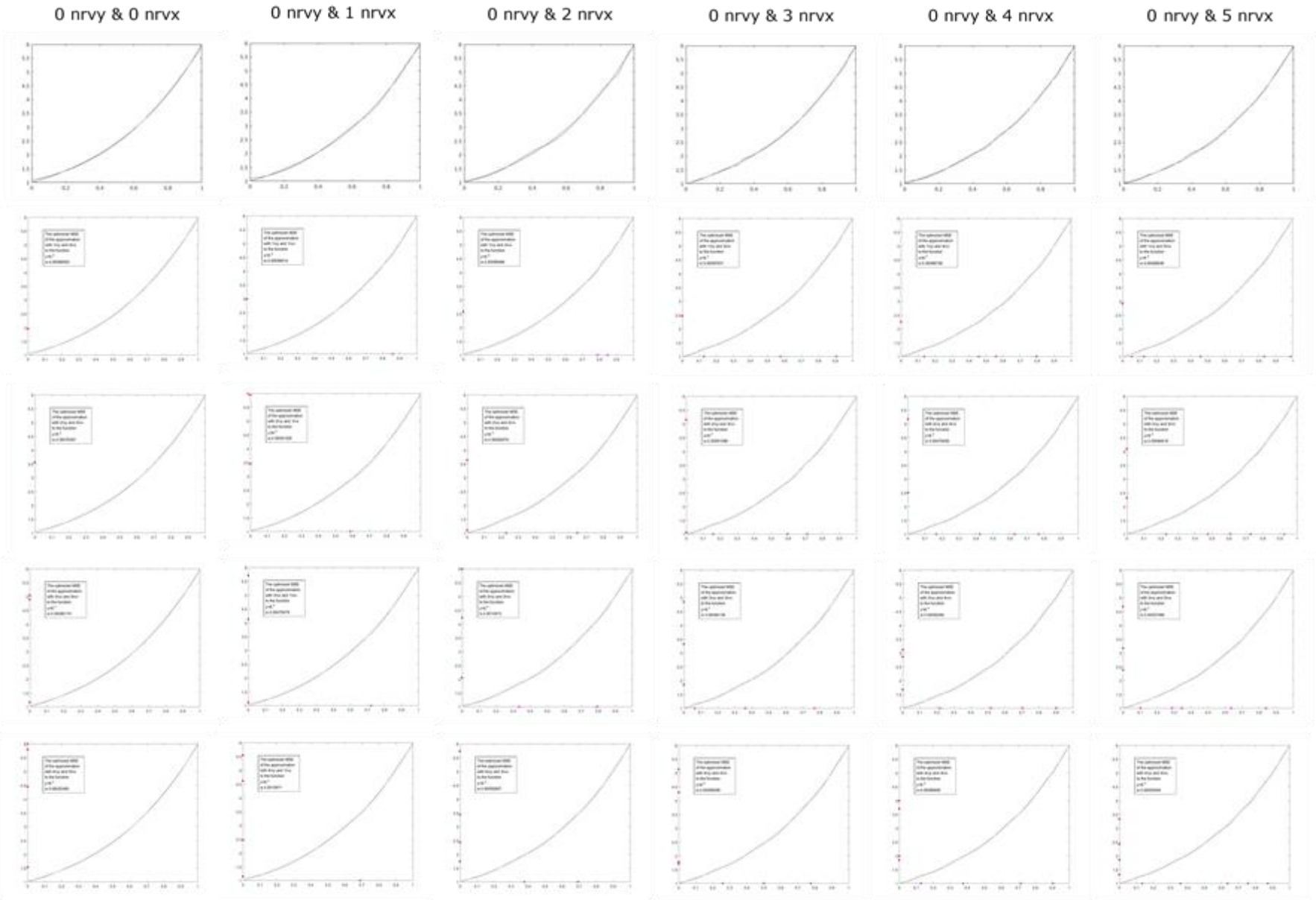



**Figure 4.6 Advanced learning for the exponential function $y = 6^x$ for MAKER-based model**

Figure 4.6 exhibits the advanced learning for the exponential function $\mathrm{y} = 6^x$ and Table 4.5 shows the corresponding MSEs for different numbers of trained x-coordinate and y-coordinate referential values. We find that no matter how many trained x-coordinate and y-coordinate referential values there are, and where they are located in the approximations, the solid red and cyan curves generally match each other well in all subfigures. In Table 4.5, it can be observed that there is little difference between the MSE for the approximation with zero trained x-coordinate referential values and zero trained y-coordinate referential values, and the MSEs for the approximations with more-than-zero of each. Both suggest there is no need to have extra referential values in addition to the four boundary referential values (minimum and maximum of x values and global extrema of observed y values) approximating the function $\mathrm{y} = 6^x$ using the MAKER framework.

Figure 4.7 reveals the advanced learning for the logarithmic function $y = \log_6 x$ and Table 4.6 displays the corresponding MSEs. It is clear from Figure 4.6 that the solid red curves coincide well with the solid cyan curves in all subfigures, no matter how many trained x-coordinate and y-coordinate referential values there are and where they are located in the approximations.

**Table 4.6 MSEs for approximations with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) (advanced learning case) for $y = \log_6 x$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | $2.40 * 10^{-5}$ | $1.38 * 10^{-5}$ | $1.07 * 10^{-5}$ | $8.44 * 10^{-6}$ | $7.23 * 10^{-6}$ | $8.32 * 10^{-6}$ |
| 1 | $1.25 * 10^{-5}$ | $6.4 * 10^{-6}$ | $7.13 * 10^{-6}$ | $3.39 * 10^{-6}$ | $7.34 * 10^{-6}$ | $4.34 * 10^{-6}$ |
| 2 | $3.75 * 10^{-6}$ | $4.69 * 10^{-6}$ | $9.56 * 10^{-6}$ | $6.83 * 10^{-6}$ | $4.96 * 10^{-6}$ | $4.72 * 10^{-6}$ |
| 3 | $6.28 * 10^{-6}$ | $4.32 * 10^{-6}$ | $7.42 * 10^{-6}$ | $8.19 * 10^{-6}$ | $6.57 * 10^{-6}$ | $4.56 * 10^{-6}$ |
| 4 | $7.62 * 10^{-6}$ | $7.24 * 10^{-6}$ | $8.19 * 10^{-6}$ | $4.99 * 10^{-6}$ | $4.77 * 10^{-6}$ | $3.48 * 10^{-6}$ |

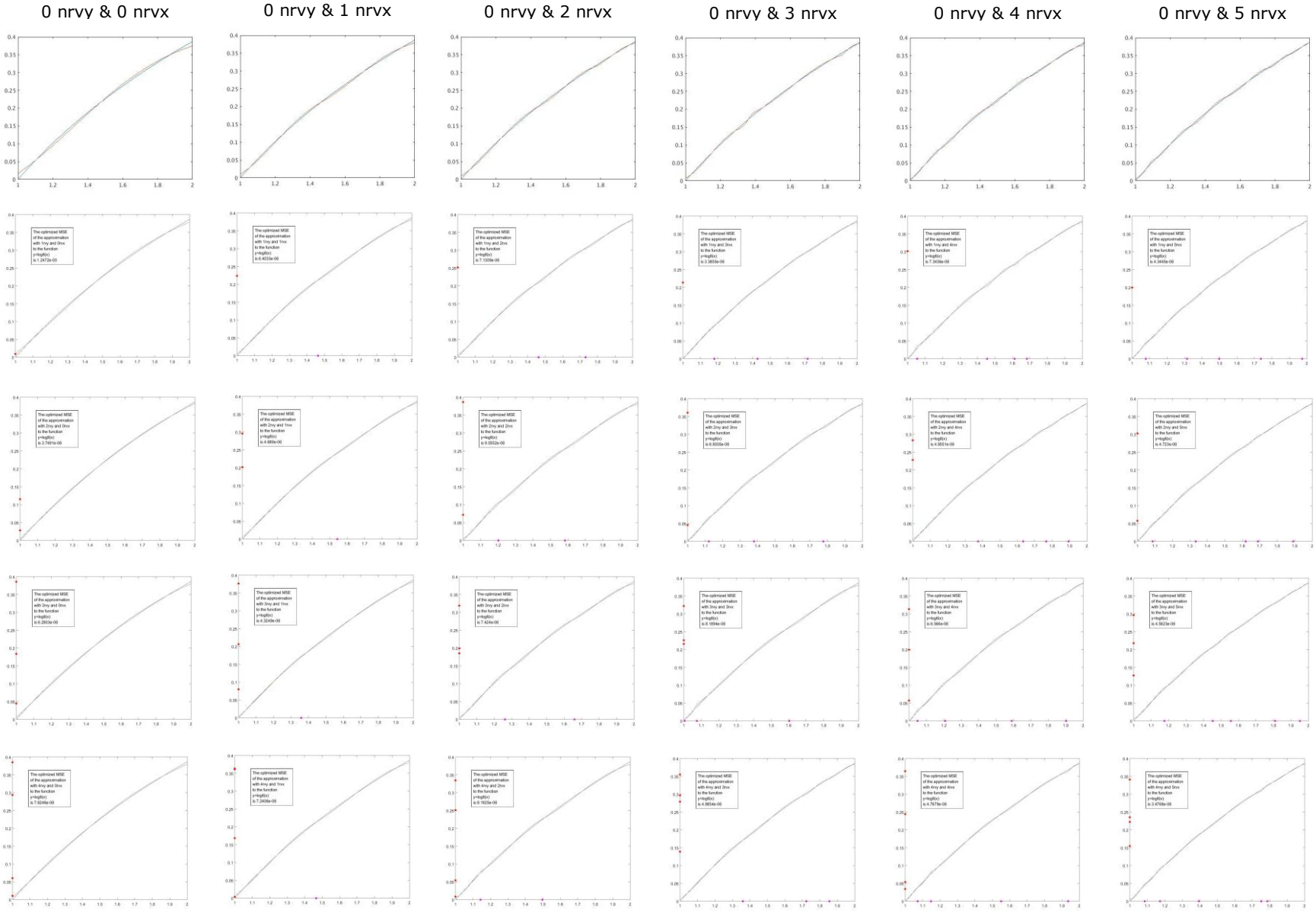



**Figure 4.7 Advanced learning for the logarithmic function $y = \log_6 x$ for MAKER-based model**

It is apparent from Table 4.6 that there is little difference between the MSE for the approximation with zero trained x-coordinate and zero trained y-coordinate referential values, and the MSEs for the approximations with more-than-zero of each and the MSE of the latter are slightly smaller. Although there is a slight improvement in the accuracy of the models with more-than-four referential values compared to the models with four, the slight improvement in accuracy is achieved at the cost of increased model complexity. This leads us to consider the possible trade-off between accuracy and complexity in the MAKER-based models.

Overall, taking all the findings mentioned above from Figure 4.7 and Table 4.6 into consideration, we can conclude that four referential values (minimum and maximum of x and the global extrema of observed y are enough for the MAKER-based model to accurately approximate the function $y = \log_6 x$.

Figure 4.8 depicts the advanced learning for the power function $y = x^{\frac{1}{6}}$ and Table 4.7 provides the corresponding MSEs. It is noteworthy that the mean curvature of power function $y = x^{\frac{1}{6}}$ between x=0 and x=1 is relatively large compared to that of the exponential function $y = 6^x$ between x=0 and x=1 and that of the logarithmic function $y = \log_6 x$ between x=1 and x=2.

**Table 4.7 MSEs for approximations with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values (advanced learning case) for $y = x^{\frac{1}{6}}$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | $1.84 * 10^{-3}$ | $1.42 * 10^{-4}$ | $8.00 * 10^{-5}$ | $3.88 * 10^{-5}$ | $2.67 * 10^{-5}$ | $2.69 * 10^{-5}$ |
| 1 | $1.85 * 10^{-3}$ | $7.66 * 10^{-5}$ | $3.14 * 10^{-5}$ | $2.2 * 10^{-5}$ | $2.17 * 10^{-5}$ | $9.97 * 10^{-6}$ |
| 2 | $1.80 * 10^{-3}$ | $9.76 * 10^{-5}$ | $4.16 * 10^{-5}$ | $1.43 * 10^{-5}$ | $1.05 * 10^{-5}$ | $1.16 * 10^{-5}$ |
| 3 | $1.79 * 10^{-3}$ | $8.08 * 10^{-5}$ | $4.47 * 10^{-5}$ | $1.26 * 10^{-5}$ | $1.17 * 10^{-5}$ | $1.32 * 10^{-5}$ |
| 4 | $1.86 * 10^{-3}$ | $1.11 * 10^{-4}$ | $2.62 * 10^{-5}$ | $1.51 * 10^{-5}$ | $1.58 * 10^{-5}$ | $8.86 * 10^{-6}$ |


**Figure 4.8 Advanced learning for the power function $y = x^{\frac{1}{6}}$ for MAKER-based model**

It is clear from Table 4.7 that no significant changes in MSEs occur as the number of trained y-coordinate referential values is increased in the approximations of $y = x^{\frac{1}{6}}$. By scrutinizing the subfigures of the approximations with zero trained x-coordinate referential values, we can see that the shapes of the red solid curves representing the predictions of MAKER-based models with more-than-zero trained y-coordinate referential values are almost the same as the shape of that with zero trained y-coordinate referential values, even as the number of trained y-coordinate referential values increased from one to four and despite these trained y-coordinate referential values being distributed in different locations of the y-axis. Thus, it can be deduced that the number and locations of trained y-coordinate referential values have no significant effect on the accuracy of the MAKER-based model approximations

From Table 4.7, we can easily observe a significant drop from the MSEs of approximations with zero trained x-coordinate referential values to the MSEs of the approximations with one trained x-coordinate referential value, and a slight reduction in the MSEs. Hence, it can be inferred that more-than-zero trained x-coordinate referential values is more appropriate for the MAKER-based model approximations of the function $y = x^{\frac{1}{6}}$.

From Figure 4.8, we can see clearly that the solid red curves in the subfigures for the approximations with more-than-zero trained x-coordinate referential values provide a better fit to the solid cyan curves than those in the subfigures for zero trained x-coordinate referential values, especially in the range between x=0 and x=0.4, where the mean curvature is large. Furthermore, it can be observed from Figure 4.7 that, compared to the approximations with zero trained x-coordinate referential values, the approximations with one trained x-coordinate referential value all have an extra trained x-coordinate referential value located between x=0 and x=0.2 where the mean curvature is large. This explains why the

approximations with one trained x-coordinate referential value provide a better fit to $y = x^{\frac{1}{6}}$ than those with zero trained x-coordinate referential values.

In addition to this, Figure 4.8 shows that, when there is more than one trained x-coordinate referential value, one of them is always located between x=0 and x=0.2. Besides this, as mentioned previously, the MSEs of the approximations with more-than-zero trained x-coordinate referential values fall slightly as the number of the trained x-coordinate referential values change from one to five, and this slight reduction is not as significant as that observed when moving from zero to one trained x-coordinate referential value. From the analysis above, we can reach the conclusion that the trained x-coordinate referential value located within the range of the function with large mean curvature plays a significant role in improving the accuracy of the MAKER-based model approximations when the function being approximated has large mean curvature.

From the above analysis of the MAKER-based model approximations for the functions $\mathrm{y} = 6^x$, $y = \log_6 x$, and $y = x^{\frac{1}{6}}$, we can reasonably conclude that only four referential values i.e., the minimum and maximum input values of the function and global extrema of the output values of the function, are enough to enable the MAKER-based models to provide accurate approximations to the monotonic univariate functions with moderate mean curvatures, e.g., $\mathrm{y} = 6^x$ and $y = \log_6 x$, but that it is necessary to have more-than-zero trained x-coordinate referential values to gain an adequate fit to monotonic univariate functions with large mean curvatures, e.g., $y = x^{1/6}$.

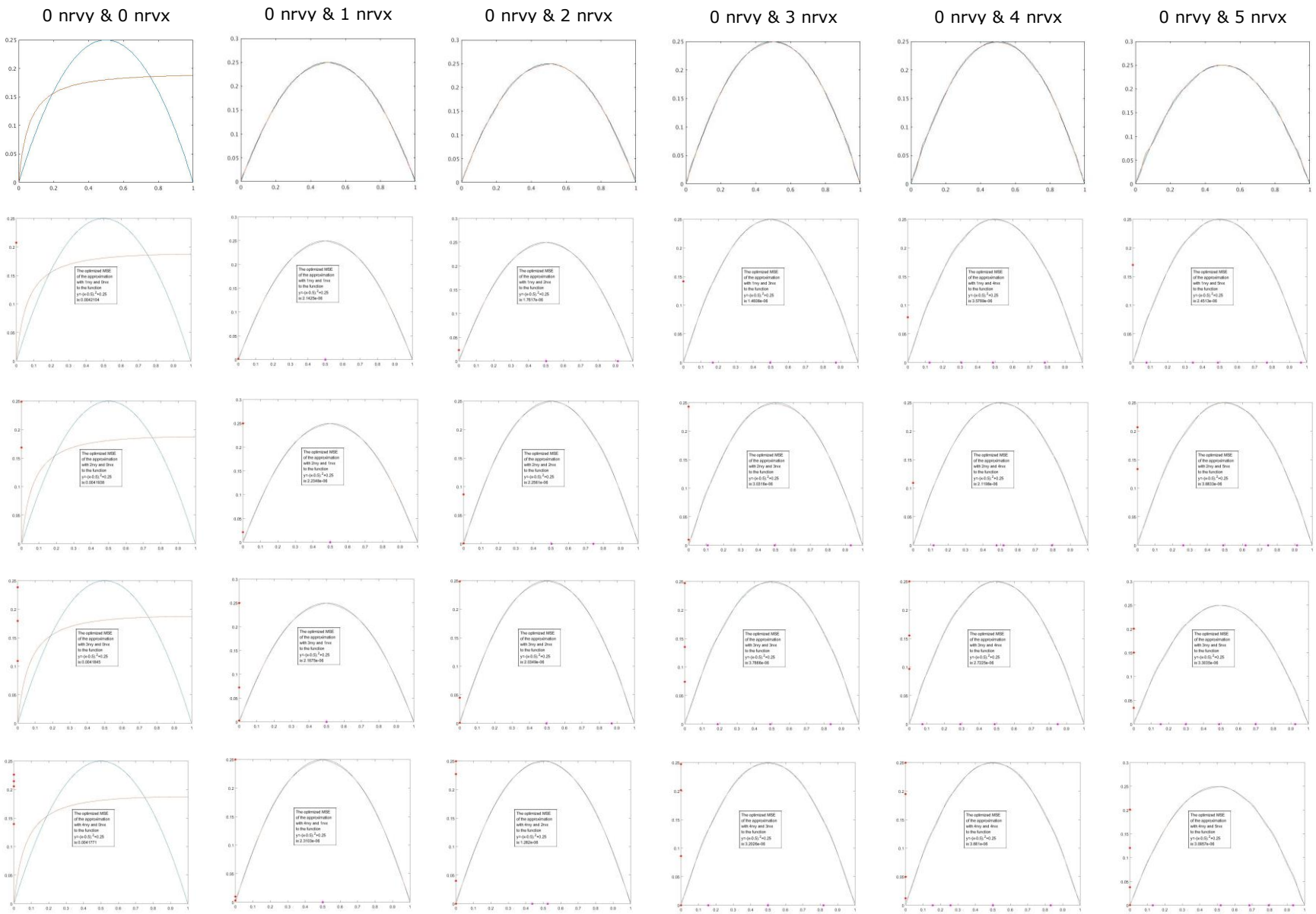



**Figure 4.9 Advanced learning for the power function $y = -(x - 0.5)^2 + 0.25$ for MAKER-based model**

**Table 4.8 MSEs for approximations with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values (advanced learning case) for $y = -(x - 0.5)^2 + 0.25$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | $4.242 * 10^{-3}$ | $2.49 * 10^{-6}$ | $3.00 * 10^{-6}$ | $1.96 * 10^{-6}$ | $3.76 * 10^{-6}$ | $5.56 * 10^{-6}$ |
| 1 | $4.21 * 10^{-3}$ | $2.14 * 10^{-6}$ | $1.76 * 10^{-6}$ | $1.46 * 10^{-6}$ | $3.58 * 10^{-6}$ | $2.45 * 10^{-6}$ |
| 2 | $4.194 * 10^{-3}$ | $2.23 * 10^{-6}$ | $2.26 * 10^{-6}$ | $3.03 * 10^{-6}$ | $2.12 * 10^{-6}$ | $3.88 * 10^{-6}$ |
| 3 | $4.184 * 10^{-3}$ | $2.17 * 10^{-6}$ | $2.03 * 10^{-6}$ | $3.79 * 10^{-6}$ | $2.72 * 10^{-6}$ | $3.3 * 10^{-6}$ |
| 4 | $4.177 * 10^{-3}$ | $2.31 * 10^{-6}$ | $1.28 * 10^{-6}$ | $3.2 * 10^{-6}$ | $3.66 * 10^{-6}$ | $3.1 * 10^{-6}$ |

Figure 4.9 shows the advanced learning for the basic non-monotonic univariate function $y = -(x - 0.5)^2 + 0.25$ and Table 4.8 the relevant MSEs for different numbers of trained x-coordinate and y-coordinate referential values. In Table 4.8, we can identify no noticeable change in MSE as the number of trained y-coordinate referential values increases. Taking a closer look at the subfigures with 0 trained x-coordinate referential values in Figure 4.8, we can see that, no matter how many trained y-coordinate referential values there are and where they are located in the approximations, the solid red curves representing the predictions of the MAKER models are all monotonic curves. Thus, it can be concluded that the number and locations of the trained y-coordinate referential values do not significantly impact the accuracy of the MAKER model approximations.

As Table 4.8 shows, the MSEs drop substantially when the number of trained x-coordinate referential values moves from zero to one and then generally remain level with slight fluctuations as the number increases to five. This leads us to conclude that more than zero trained x-coordinate referential values enable the MAKER models to provide more accurate approximations to the monotonic univariate function $y = -(x - 0.5)^2 + 0.25$.

As Figure 4.9 shows, the solid red curves in the subfigures based on zero trained x-coordinate referential values are just monotonic curves, which are obviously unable to effectively approximate the solid cyan curves, while the red curves in the subfigures based on more than zero trained x-coordinate referential values match

the cyan curves well. From Figure 4.9, it can also be observed that, in each of the approximations with one trained x-coordinate referential value, that value is located at x=0.5, i.e., the x-coordinate of the critical point on the curve of $y = -(x - 0.5)^2 + 0.25$. Moreover, we can see clearly from Figure 4.9 that the trained x-coordinate referential values in the approximations with more than one such value all include one located at x=0.5.

Combining the locations of the trained x-coordinate referential values and how the red solid curves match the solid cyan curves in Figure 4.9 with the pattern of MSEs displayed in Table 4.8, we can safely conclude that one trained x-coordinate referential value is enough for the MAKER model to accurately approximate the basic non-monotonic univariate function $y = -(x - 0.5)^2 + 0.25$ with only one critical point and hence two monotone intervals. Also, this trained x-coordinate referential value is generally located at the x-coordinate of the critical point of the function, minimizing the difference between the predicted and observed output values.

When coupled with the approximations for the monotonic univariate functions ($y = 6^x$, $y = \log_6 x$, and $y = x^{1/6}$), those for the non-monotonic univariate function $y = -(x - 0.5)^2 + 0.25$ naturally lead us to conclude that two adjacent x-coordinate referential values can only be used to approximate monotonic curves, while we need at least one trained x-coordinate referential value to approximate non-monotonic univariate functions using MAKER models.

The approximation for $y = -(x - 0.5)^2 + 0.25$ provides a perspective on MAKER model approximations for basic non-monotonic univariate functions. For the more general case, i.e., complex non-monotonic univariate functions, as mentioned previously, we use a multi-extremal function, $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$. This function has three extrema, i.e., a local minimum at x=0, a local maximum at x=-2, and a global maximum at x=2, and hence four monotone intervals. Furthermore, this function has four inflection points, which are located at approximately x=-2.71, x=-

1.29, x=1.29, and x=2.71 respectively. With the data set of observed input-output data pairs, advanced learning is performed for this function by again building MAKER models.

Figure 4.10 illustrates the advanced learning for this function and Table 4.9 shows the MSEs for different numbers of trained x-coordinate and y-coordinate referential values. Figure 4.11 then visualizes the MSEs from Table 4.9 to provide a more intuitive display of how the MSEs change with the number of trained x-coordinate and y-coordinate referential values.

From Table 4.10 and Figure 4.11, it can be observed that there are significant drops in the MSEs of the approximations as we move from zero to one, to two, and finally to three trained x-coordinate referential values. The MSEs generally then remain stable, experiencing only slight fluctuations, as we move from three to ten trained x-coordinate referential values. Meanwhile, no significant changes in the MSEs can be observed as the number of trained y-coordinate referential values changes.


**Figure 4.10 Advanced learning for the power function $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$ for MAKER-based model**

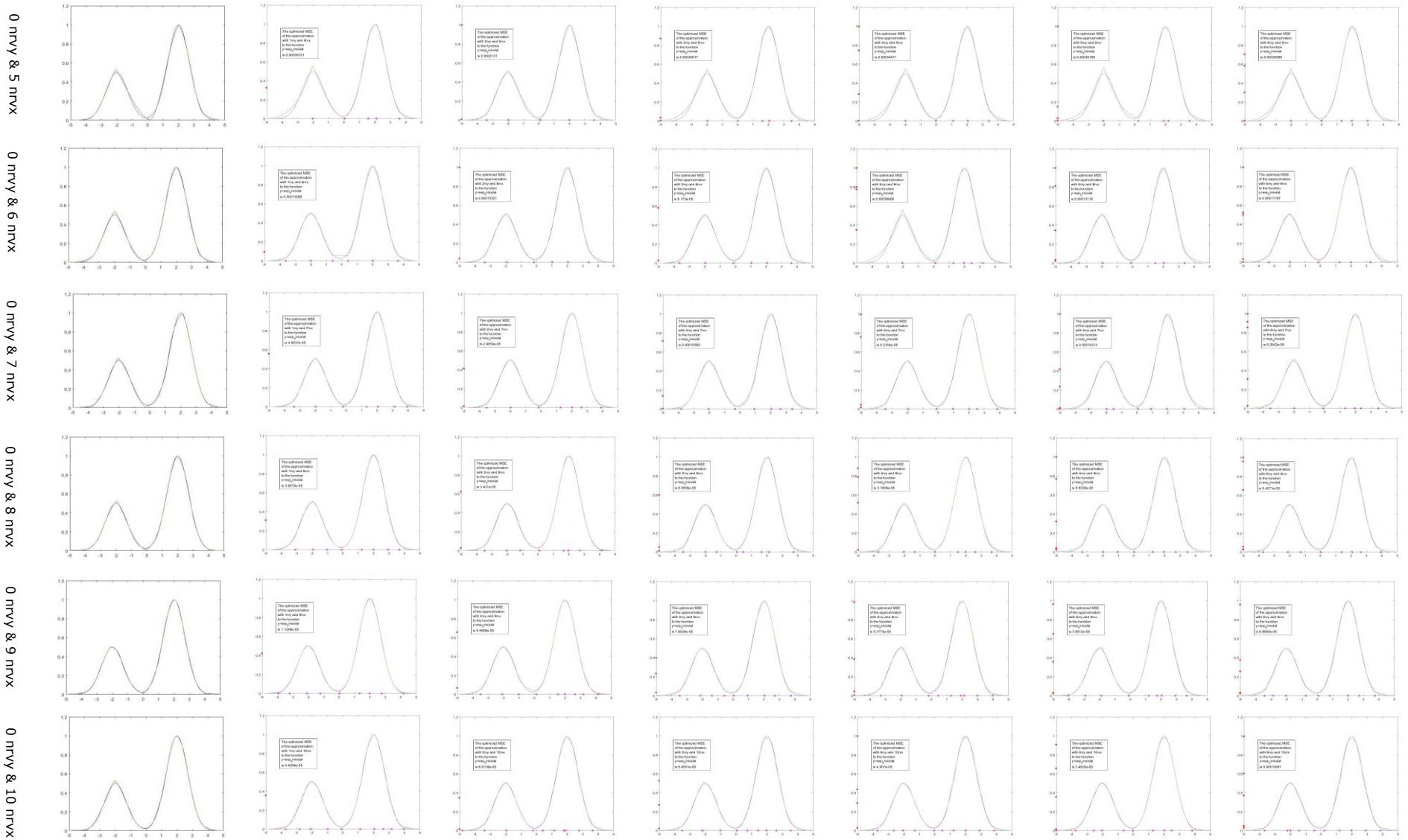



**Figure 4.10 Advanced learning for the power function $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$ for MAKER-based model**

**Table 4.9 MSEs for approximations with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values (advanced learning case) for $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | $7.52 * 10^{-2}$ | $2.50 * 10^{-2}$ | $1.68 * 10^{-2}$ | $2.32 * 10^{-3}$ | $3.42 * 10^{-3}$ | $3.04 * 10^{-4}$ | $1.04 * 10^{-4}$ | $1.47 * 10^{-4}$ | $5.41 * 10^{-5}$ | $4.45 * 10^{-5}$ | $5.73 * 10^{-5}$ |
| 1 | $7.55 * 10^{-2}$ | $2.50 * 10^{-2}$ | $1.49 * 10^{-2}$ | $1.68 * 10^{-3}$ | $4.74 * 10^{-4}$ | $3.54 * 10^{-4}$ | $1.33 * 10^{-4}$ | $4.06 * 10^{-5}$ | $3.59 * 10^{-5}$ | $1.13 * 10^{-5}$ | $4.43 * 10^{-5}$ |
| 2 | $7.51 * 10^{-2}$ | $2.50 * 10^{-2}$ | $1.52 * 10^{-2}$ | $1.69 * 10^{-3}$ | $3.73 * 10^{-4}$ | $2.13 * 10^{-4}$ | $1.03 * 10^{-4}$ | $2.50 * 10^{-5}$ | $3.43 * 10^{-5}$ | $5.56 * 10^{-5}$ | $6.01 * 10^{-5}$ |
| 3 | $7.51 * 10^{-2}$ | $2.50 * 10^{-2}$ | $1.49 * 10^{-2}$ | $1.82 * 10^{-3}$ | $3.97 * 10^{-4}$ | $3.46 * 10^{-4}$ | $8.17 * 10^{-5}$ | $1.06 * 10^{-4}$ | $6.06 * 10^{-5}$ | $7.65 * 10^{-5}$ | $6.46 * 10^{-5}$ |
| 4 | $7.51 * 10^{-2}$ | $2.50 * 10^{-2}$ | $1.51 * 10^{-2}$ | $1.56 * 10^{-3}$ | $4.94 * 10^{-4}$ | $3.45 * 10^{-4}$ | $3.98 * 10^{-4}$ | $5.22 * 10^{-5}$ | $3.18 * 10^{-5}$ | $5.78 * 10^{-5}$ | $4.01 * 10^{-5}$ |
| 5 | $7.51 * 10^{-2}$ | $2.50 * 10^{-2}$ | $1.51 * 10^{-2}$ | $1.69 * 10^{-3}$ | $4.42 * 10^{-4}$ | $4.52 * 10^{-4}$ | $1.31 * 10^{-4}$ | $1.52 * 10^{-4}$ | $9.83 * 10^{-5}$ | $3.90 * 10^{-5}$ | $2.47 * 10^{-5}$ |
| 6 | $7.51 * 10^{-2}$ | $2.50 * 10^{-2}$ | $1.52 * 10^{-2}$ | $1.32 * 10^{-2}$ | $4.55 * 10^{-4}$ | $3.10 * 10^{-4}$ | $1.18 * 10^{-4}$ | $5.26 * 10^{-5}$ | $5.43 * 10^{-5}$ | $6.87 * 10^{-5}$ | $1.01 * 10^{-4}$ |

Combining the locations of the trained x-coordinate and y-coordinate referential values displayed in Figure 4.10 with Figure 4.11, we can conclude that significant changes in MSEs are highly associated with the number and locations of the trained x-coordinate referential values.

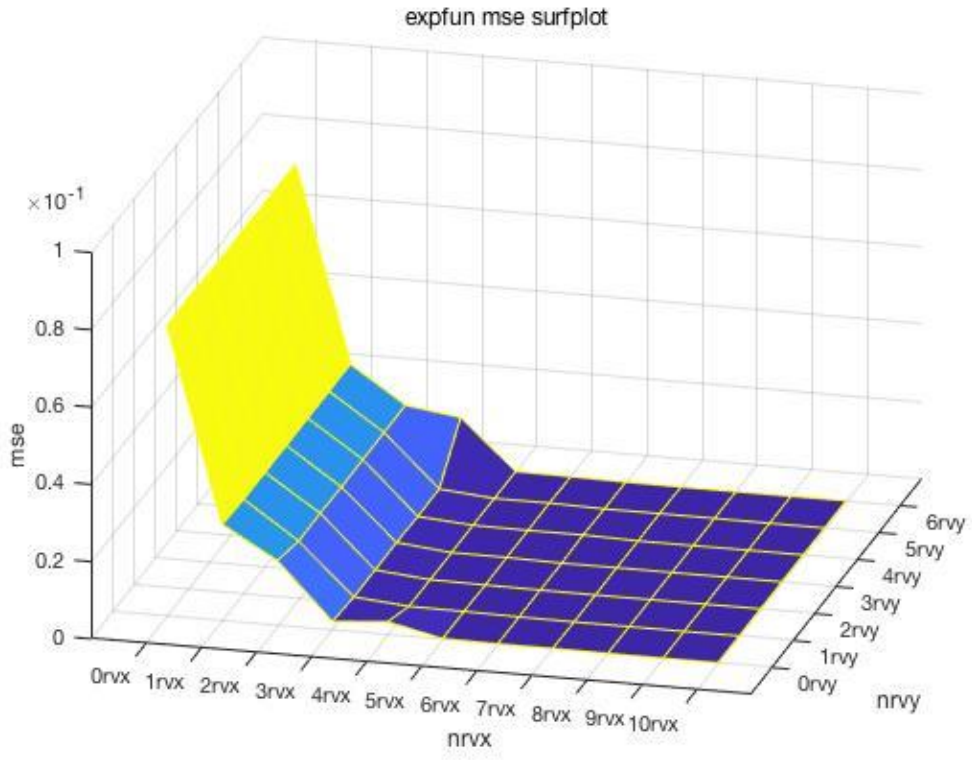



**Figure 4.11 Surface plot of MSEs for approximations with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values in the advanced learning for** $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$

Specifically, a significant drop in MSEs is observed in the approximations with only one trained x-coordinate referential value that is located approximately at the x-coordinate of the global maximum of the function (i.e., x=2) relative to the approximations with zero trained x-coordinate referential values. In the approximations with two trained x-coordinate referential values located close to the x-coordinate of the local minimum of the function (i.e., x=0) and the x-coordinate of the global maximum (i.e., x=2) respectively, we can see clearly that a significant drop occurs in the MSEs relative to the approximations with only one trained x-

coordinate referential value. Similar significant drops in MSEs can be observed when we move from the approximations with two trained x-coordinate referential values to those with three trained x-coordinate referential values located close to the x-coordinate of the local maximum of the function (i.e., x=-2), the x-coordinate of the local minimum of the function (i.e., x=0), and the x-coordinate of the global maximum of the function (i.e., x=2) respectively.

From Table 4.9, Figure 4.10, and Figure 4.11, we can also see that, as we move from the approximations with three trained x-coordinate referential values to those with ten, although the number of trained x-coordinate referential values increases, the MSEs generally remain level with only slight fluctuations. Also, from Figure 4.10, there are significant differences between the solid red curves representing the predictions of the MAKER models and the corresponding solid cyan curves representing the observations of the function in the subfigures corresponding to one and two trained x-coordinate referential values, while in those with three trained x-coordinate referential values, the solid red curves basically match the solid cyan curves.

Taking all the findings from Table 4.9, Figure 4.10, and Figure 4.11 into consideration, we naturally come to the conclusion that three trained x-coordinate referential values are enough for the MAKER framework to provide an adequate fit to the function $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$ which has a local minimum at x=0, a local maximum at x=-2, and a global maximum at x=2 and hence four monotone intervals, and that, through optimization, the trained x-coordinate referential values of the approximations are generally located around the x-coordinates of the extrema of the function.

By scrutinizing the second column (with the header '0') of Table 4.9 and the subfigures with zero trained x-coordinate referential values in Figure 4.10, we can observe that increasing the number of trained y-coordinate referential values does

not improve the MAKER model approximations of the multi-extremal function $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$ and there is no significant relationship between the number and locations of the trained y-coordinate referential values and the accuracy of the approximations. Taking a closer look at the subfigures corresponding to more than three trained x-coordinate referential values in Figure 4.9, we can observe that these MAKER models provide a better fit to the multi-extremal function than those based on just three trained x-coordinate referential values, as these approximations feature extra trained x-coordinate referential values located between adjacent trained x-coordinate referential values that are situated approximately at the endpoints of the monotone intervals of the function.

For instance, as can be seen from Figure 4.10, that shows the approximations with one trained y-coordinate referential value, the approximation with four trained x-coordinate referential values has an extra trained x-coordinate referential value located between the global maximum (i.e., x=2) and the maximum of the observed input values (i.e., x=5) of the multi-extremal function, relative to the approximation with three trained x-coordinate referential values, giving the approximation with four trained x-coordinate referential values a closer fit to the multi-extremal function in the range between x=2 and x=5.

This is mainly because of the inflection point on the multi-extremal function at about x=2.71, which divides the curve between x=2 and x=5 into two segments, i.e. that between x=2 and x=2.71 which is concave downward and that between x=2.71 and x=5 which is convex downward. Moreover, the initial learning for exponential functions $\mathrm{y} = a^x$ and logarithmic functions $y = \log_a x$ showed that the MAKER framework can accurately approximate these two types of function using only two adjacent x-coordinate referential values and two adjacent y-coordinate referential values. For these two reasons, an extra trained x-coordinate referential value between two adjacent x-coordinate referential values could lead to a better approximation to a segment of a function that has both convexity and concavity.

Similarly, the approximation with five trained x-coordinate referential values has an extra one at approximately x=1.29 i.e., the x-coordinate of the inflection point between x=0 and x=2, relative to the approximation with four such values. In addition, this extra trained x-coordinate referential value is situated between that at about x=0 (local minimum of the function) and that at about x=2 (global maximum of the function).

Without doubt, this extra trained x-coordinate referential value leads to the approximation with five trained x-coordinate referential values having a closer fit to the curve of the extremal function in the range between x=0 and x=2. With a better fit to the segment between x=0 and x=2 and a better fit between x=2 and x=5, the approximation with five trained x-coordinate referential values naturally provides a better fit to the critical point of the multi-extremal function at x=2, and its vicinity, than the approximation with four trained x-coordinate referential values, which can be observed from Figure 4.10.

As previously mentioned, an extra trained x-coordinate referential value close to the x-coordinate of the point at which a function has large curvature, between adjacent x-coordinate referential values, can be used to effectively improve the MAKER model approximation of the curved segment of a function with large mean curvature. Thus, by comparing the approximation based on four trained x-coordinate referential values to that based on seven, we can see that, between x=2 and x=5, both approximations have trained x-coordinate referential values close to x=2, i.e., at a critical point of the multi-extremal function, and close to x=2.71, at an inflection point of the function, and that the approximation based on seven trained x-coordinate referential values has an extra one close to x=4, where the multi-extremal function has large curvature, which leads to that one providing a better fit to the curve of the multi-extremal function between x=3 and x=5.

As shown in the subfigure of Figure 4.11 for nine trained x-coordinate referential values, the solid red curve representing the predictions of the MAKER model generally coincides very well with the solid cyan curve representing the observations of the multi-extremal function. This is mainly due to the fact that each of the important points, i.e., the critical points and the inflection points of the multi-extremal function, has a corresponding trained x-coordinate referential value. In addition to this, the monotonic curve segments of the multi-extremal function between x=-5 and x=-3 and between x=3 and x=5 have large mean curvatures and corresponding trained x-coordinate referential values located near to x=-4 and x=4 respectively.

## 4.4 Bivariate Function Approximations

The approximations of univariate functions elucidated in Section 4.3 have preliminarily indicated the great approximation capability of MAKER models. In this section, approximations of a bivariate function, namely the Himmelblau function (Himmelblau, 1972), are used as an example to illustrate the approximation capability of MAKER models for more complex cases, i.e., bivariate functions. The Himmelblau function is a benchmark function used to test optimization techniques. It is displayed in Equation (4.1) (Andrei, 2008). It has one local maximum, $z = 181.616$, at the point (-0.270844, -0.923038), and four identical local minima, $z = 0$, at the points (3.0, 2.0), (3.584428, -1.848126), (-2.805118, 3.131312), and (-3.779310, -3.283186). A three-dimensional surface plot of the Himmelblau function is presented in Figure 4.12.

$$z = (x^2 + y - 11)^2 + (x + y^2 - 7)^2, -6 \leq x, y \leq 6 \tag{4.1}$$

In this study, we use the Himmelblau function to generate observed input-output data pairs. For instance, for an observed input value of the variables x and y, e.g. x=1.0 and y=1.6, the corresponding observed output value of the Himmelblau function can be derived (z=82.3936). Thus, x=1.0, y=1.6, and z=82.3936 form a data record (1.0, 1.6, 82.3936). Using the same method, we can generate other data records to form a data set for approximation of the Himmelblau function.

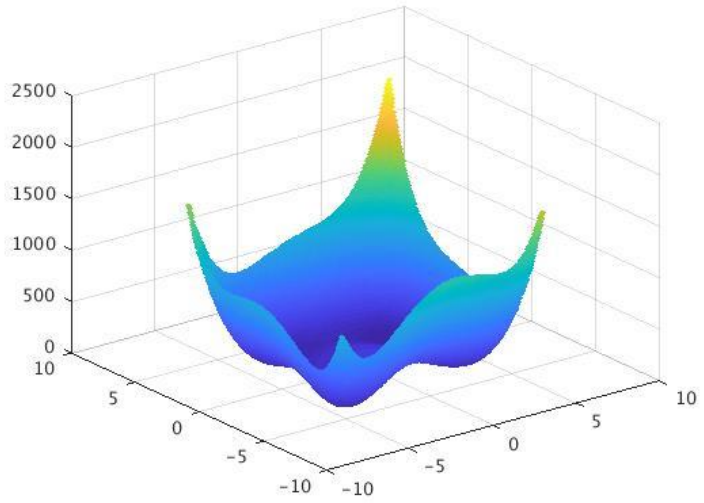



**Figure 4.12 The three-dimensional surface plot of Himmelblau function**

In the data set for the approximation, the observed input values of variables x and y are distributed uniformly, and the interval between any two adjacent x or y values is set to 0.2. Moreover, the data set for the approximation also includes the observed output values of the Himmelblau function, which are represented by the z-coordinate values in Figure 4.12. With this data set, MAKER models are built to approximate the Himmelblau function.

The parameters of the models, i.e., the x-coordinate, y-coordinate, and z-coordinate referential values, and the weights, are acquired by minimizing the differences between the observed z-coordinate values of the Himmelblau function and the predicted z-coordinate values generated by the models using the adapted genetic algorithm. The numbers of trained x-coordinate and y-coordinate referential values range from 1 to 16. The number of trained z-coordinate referential values ranges from 1 to 6. It is noteworthy that the 'trained x-coordinate

referential values' and the 'trained y-coordinate referential values' are short for the trained referential values of observed input values of the variable x of the Himmelblau function and the trained referential values of observed input values of the variable y of the Himmelblau function respectively. Both the trained x-coordinate and y-coordinate referential values are located between the minimum and the maximum of the observed input x and y values respectively. Similarly, the 'trained z-coordinate referential values' is short for the trained referential values of observed output values between the minimum and the maximum of observed output values of the Himmelblau function.

Figure 4.13 exhibits the approximations of the MAKER models for the Himmelblau function. In the subfigures of Figure 4.13, the red points on the x-axes, y-axes, and z-axes indicate the trained x-coordinate, y-coordinate, and z-coordinate referential values respectively. The blue points in the subfigures indicate the predictions generated by the MAKER models and the semitransparent cyan surface denotes the surface generated by the observations of the Himmelblau function.

The primary means of evaluating the accuracy of the Himmelblau function approximations provided by the MAKER models is to check the normalized MSEs for the approximations and determine whether the surfaces of the blue points representing the predictions generated by the models fit into the semitransparent cyan surfaces representing the observations of the Himmelblau function. Additionally, the local maxima and minima can be identified from the predicted z-coordinate values generated by the models, and these extrema generated by the models can then be compared to those generated by the observed output values of the Himmelblau function to find out whether the locations of the two are close to one another, as a further accuracy check.

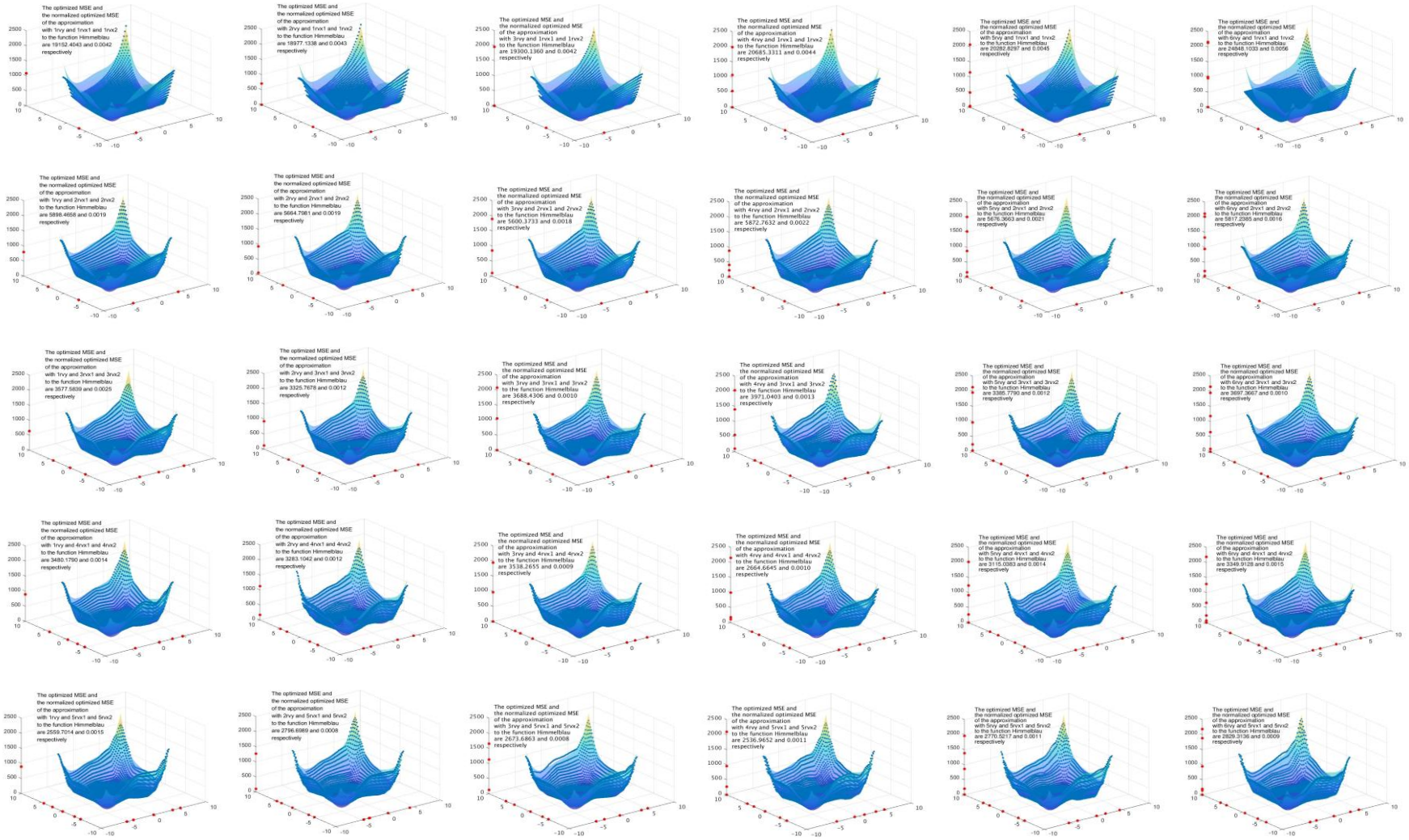



**Figure 4.13 Advanced learning for the Himmelblau function for MAKER-based model**

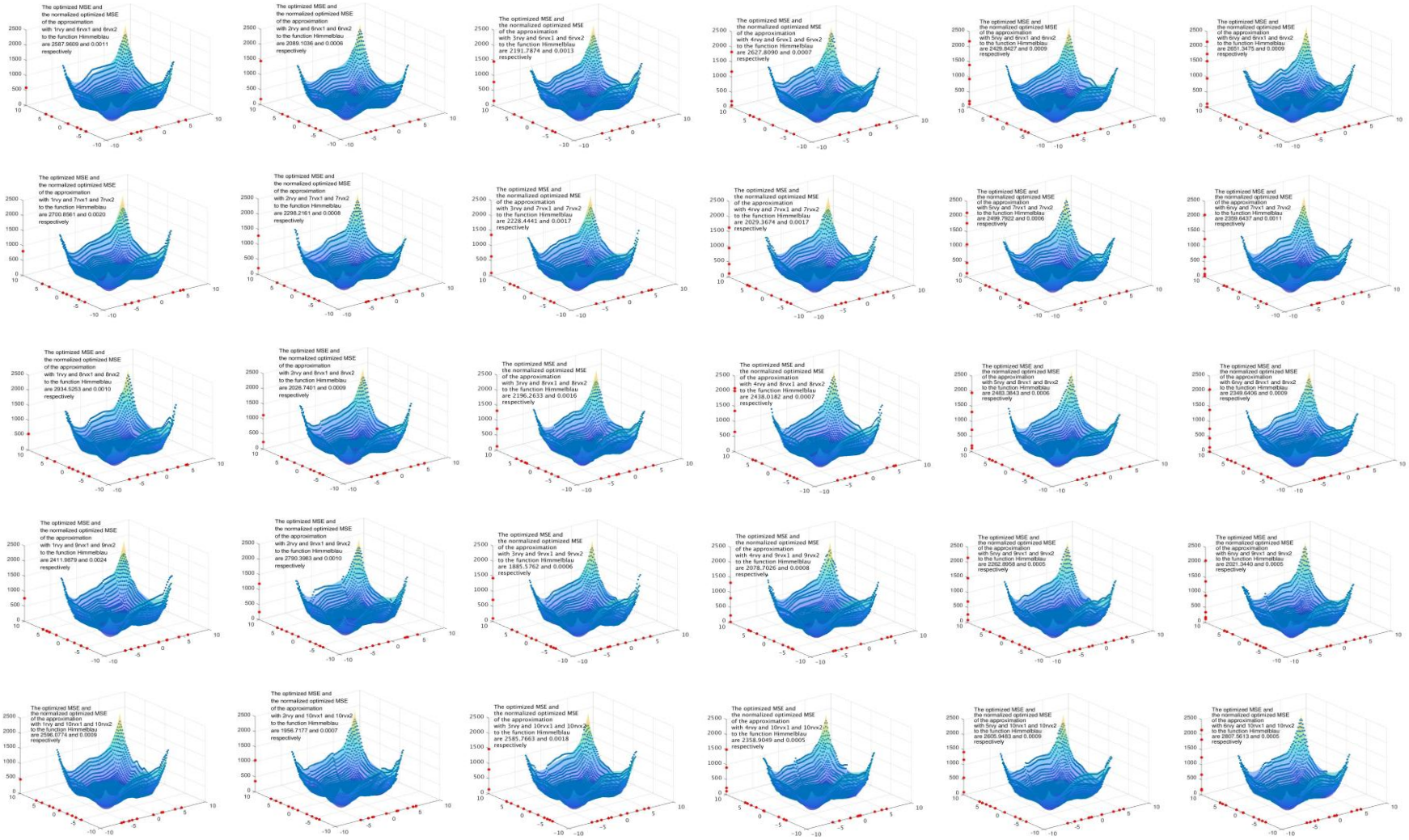


**Figure 4.13 Advanced learning for the Himmelblau function for MAKER-based model**

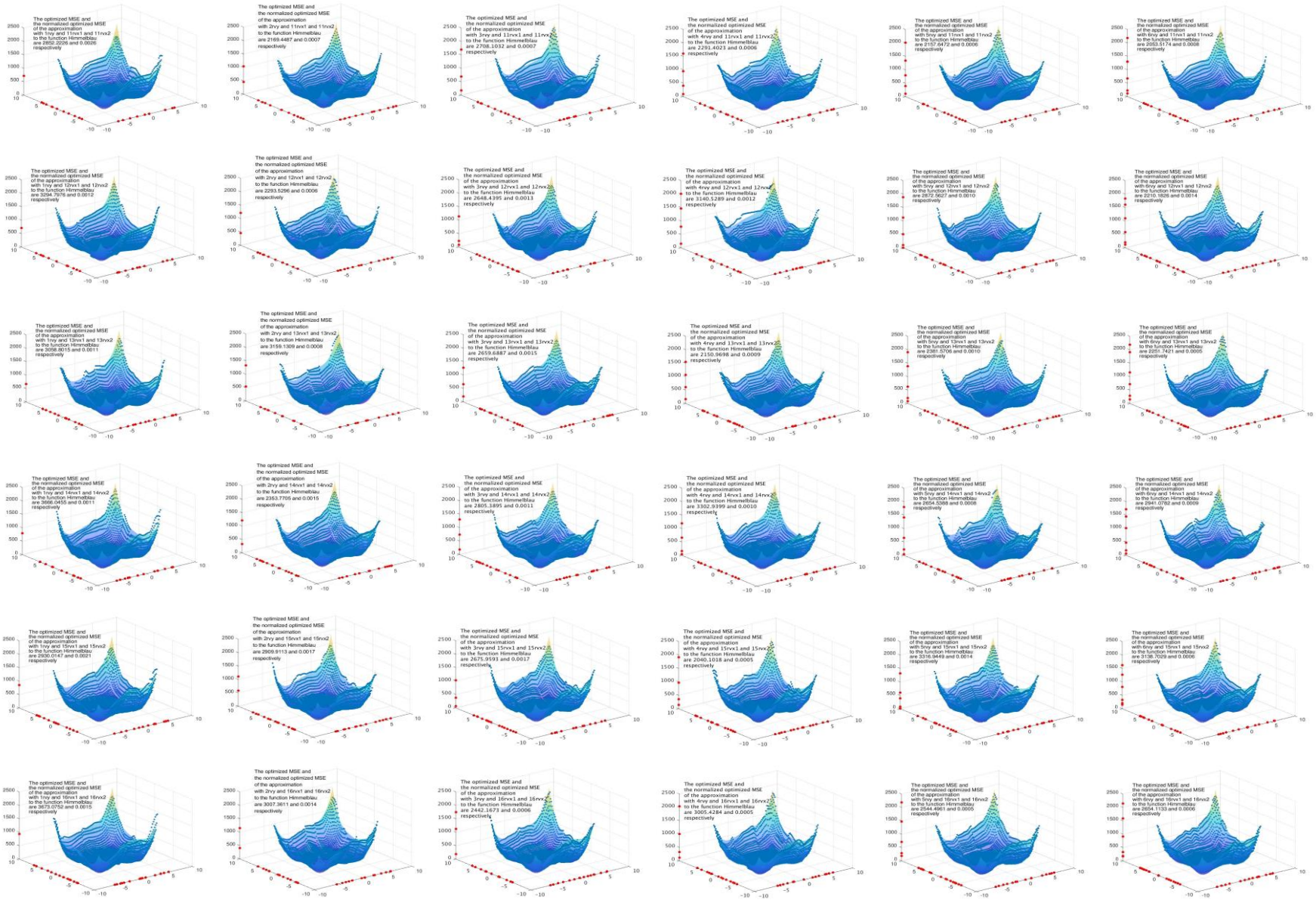


**Figure 4.13 Advanced learning for the Himmelblau function for MAKER-based model**

### 4.4.1 Normalized Mean Squared Error (MSE)

In line with what has been stated above, we start with the MSEs of the MAKER model approximations of the Himmelblau function to evaluate their accuracy. Table 4.10 displays the normalized MSEs for different numbers of trained x-coordinate, y-coordinate, and z-coordinate referential values. The main reason for using the normalized MSEs to evaluate the accuracy of the approximations, according to Lughofer (2013), is that different ranges of different output values could cause completely different MSE values, although the quality of the models might be the same, which means the MSE could be quite uninterpretable as it can be an isolated measure of model accuracy. To address this issue, we can either normalize the data or use a normalized error measure. The normalized error measure is preferable to data normalization in that data within the original ranges are better understood by the experts than normalized data. Thus, as a normalized error measure, the normalized MSE is defined as in Equation (4.2), where $\hat{z}$ indicates the predicted output values generated by the model and z indicates the observed output values of the Himmelblau function.

$$mse_{norm} = \frac{1}{N}\sum_{n=1}^{N}\frac{(\hat{z}(n) - z(n))^2}{(\max(z) - \min(z))^2} \tag{4.2}$$

Figure 4.14 provides a visualization of the MSEs exhibited in Table 4.10, as a more intuitive display of how the MSEs change with the number of trained x-coordinate, y-coordinate, and z-coordinate referential values. From Table 4.10 and Figure 4.13, it can be observed that there are significant drops in the MSEs as we move from the approximations with one trained x-coordinate referential value and one trained y-coordinate referential value to those with two of each, and again as we move to those with three of each. Additionally, it can be seen from Table 4.10 and Figure 4.13 that the MSEs generally remain stable with just slight fluctuations as we move

from the approximations with three to those with sixteen trained x-coordinate and y-coordinate referential values. Finally, we can identify no significant changes in the MSEs as the number of trained z-coordinate referential values increases.

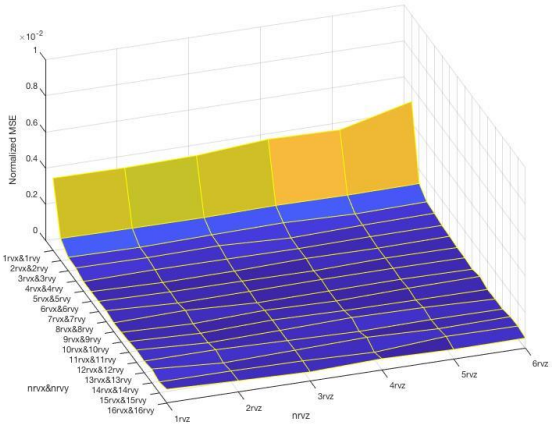



**Figure 4.14 Surface plot of MSEs for approximations with different numbers of trained x-coordinate (nrvx), y-coordinate (nrvy), and z-coordiante referential values in the advanced learning for Himmelblau Function**

### 4.4.2 Surface Fitting

By integrating the locations of the trained x-coordinate, y-coordinate, and z-coordinate referential values exhibited in Figure 4.13 with the pattern of MSEs presented in Table 4.10 and Figure 4.14, we can observe a strong association between the significant changes in the MSEs and the number and locations of trained referential values.

Specifically speaking, compared to the approximations with one trained referential value in either the x- or y-coordinate, the approximations with two trained referential values in either the x- or y-coordinate have a trained x-coordinate referential value located around -3.779310 and a trained y-coordinate referential value located around -3.283186, which correspond to the x- and y-coordinates (-3.779310, -3.283186) of a local minimum at which z=0. These referential values are similar to those of the approximations with one trained x-coordinate referential value and one trained y-coordinate referential value.

Moreover, the approximations with two trained x-coordinate and two trained y-coordinate referential values have an extra one of each, which are generally located at around x=3.584428, i.e., the x-coordinate of a local minimum (z=0) at (3.584428, -1.848126) and y=3.131312, i.e., the y-coordinate of a local minimum (z=0) at (-2.805118, 3.131312), respectively, which leads to a significant drop in the MSEs of these approximations compared to those of the approximations with one trained x-coordinate and one trained y-coordinate referential value.

As previously mentioned, a similar significant drop in MSEs can be observed when we move from the approximations with two trained x-coordinate and two trained y-coordinate referential values to those with three of each. Both approximations have trained x-coordinate referential values located near to -3.779310, i.e., the x-coordinate of a local minimum (z=0) at (-3.779310, -3.283186) and 3.584428, i.e., the x-coordinate of a local minimum (z=0) at (3.584428, -1.848126), respectively, and trained y-coordinate referential values located near to -3.283186, i.e., the y-coordinate of a local minimum (z=0) at (-3.779310, -3.283186) and 3.131312, i.e., the y-coordinate of a local minimum (z=0) at (-2.805118, 3.131312) respectively.

Compared to the approximations with two trained x-coordinate and two trained y-coordinate referential values, those with three of each have an extra one near to -0.270844, i.e., the x-coordinate of a local maximum z=181.616 at (-0.270844, -0.923038) and near to 2.0, i.e., the y-coordinate of a local minimum (z=0) at (3.0, 2.0) respectively. This leads to a significant drop in the MSEs.

In addition to the patterns observed in Table 4.10, Figure 4.13, and Figure 4.14, as previously stated, we can evaluate the accuracy of the MAKER model approximations by checking whether the surfaces of the blue points indicating the predictions of the models provide a good fit to the semitransparent cyan surfaces indicating the observations of the Himmelblau function.

**Table 4.10 Normalized MSEs for approximations with different numbers of trained x-coordinate (nrvx), y-coordinate (nrvy), and z-coordinate (nrvz) referential values, for the Himmelblau function**

| nrvx & nrvy\nrvz | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 & 1 | $4.01 * 10^{-3}$ | $3.97 * 10^{-3}$ | $4.04 * 10^{-3}$ | $4.33 * 10^{-3}$ | $4.24 * 10^{-3}$ | $5.20 * 10^{-3}$ |
| 2 & 2 | $1.23 * 10^{-3}$ | $1.19 * 10^{-3}$ | $1.17 * 10^{-3}$ | $1.23 * 10^{-3}$ | $1.19 * 10^{-3}$ | $1.22 * 10^{-3}$ |
| 3 & 3 | $7.49 * 10^{-4}$ | $6.96 * 10^{-4}$ | $7.72 * 10^{-4}$ | $8.31 * 10^{-4}$ | $7.09 * 10^{-4}$ | $7.74 * 10^{-4}$ |
| 4 & 4 | $7.28 * 10^{-4}$ | $6.87 * 10^{-4}$ | $7.40 * 10^{-4}$ | $5.58 * 10^{-4}$ | $6.52 * 10^{-4}$ | $7.01 * 10^{-4}$ |
| 5 & 5 | $5.36 * 10^{-4}$ | $5.85 * 10^{-4}$ | $5.60 * 10^{-4}$ | $5.31 * 10^{-4}$ | $5.80 * 10^{-4}$ | $5.92 * 10^{-4}$ |
| 6 & 6 | $5.42 * 10^{-4}$ | $4.37 * 10^{-4}$ | $4.59 * 10^{-4}$ | $5.50 * 10^{-4}$ | $5.08 * 10^{-4}$ | $5.55 * 10^{-4}$ |
| 7 & 7 | $5.65 * 10^{-4}$ | $4.81 * 10^{-4}$ | $4.66 * 10^{-4}$ | $4.25 * 10^{-4}$ | $5.23 * 10^{-4}$ | $4.94 * 10^{-4}$ |
| 8 & 8 | $6.14 * 10^{-4}$ | $4.24 * 10^{-4}$ | $4.60 * 10^{-4}$ | $5.10 * 10^{-4}$ | $5.20 * 10^{-4}$ | $4.92 * 10^{-4}$ |
| 9 & 9 | $5.05 * 10^{-4}$ | $5.84 * 10^{-4}$ | $3.95 * 10^{-4}$ | $4.35 * 10^{-4}$ | $4.74 * 10^{-4}$ | $4.23 * 10^{-4}$ |
| 10 & 10 | $5.43 * 10^{-4}$ | $4.09 * 10^{-4}$ | $5.41 * 10^{-4}$ | $4.94 * 10^{-4}$ | $5.45 * 10^{-4}$ | $5.88 * 10^{-4}$ |
| 11 & 11 | $5.97 * 10^{-4}$ | $4.54 * 10^{-4}$ | $5.67 * 10^{-4}$ | $4.80 * 10^{-4}$ | $4.52 * 10^{-4}$ | $4.30 * 10^{-4}$ |
| 12 & 12 | $6.89 * 10^{-4}$ | $4.80 * 10^{-4}$ | $5.54 * 10^{-4}$ | $6.57 * 10^{-4}$ | $6.01 * 10^{-4}$ | $4.63 * 10^{-4}$ |
| 13 & 13 | $6.40 * 10^{-4}$ | $6.61 * 10^{-4}$ | $5.57 * 10^{-4}$ | $4.50 * 10^{-4}$ | $4.98 * 10^{-4}$ | $4.71 * 10^{-4}$ |
| 14 & 14 | $7.67 * 10^{-4}$ | $4.93 * 10^{-4}$ | $5.87 * 10^{-4}$ | $6.91 * 10^{-4}$ | $5.56 * 10^{-4}$ | $6.15 * 10^{-4}$ |
| 15 & 15 | $6.13 * 10^{-4}$ | $6.09 * 10^{-4}$ | $5.60 * 10^{-4}$ | $4.27 * 10^{-4}$ | $6.94 * 10^{-4}$ | $6.57 * 10^{-4}$ |
| 16 & 16 | $7.69 * 10^{-4}$ | $6.29 * 10^{-4}$ | $5.11 * 10^{-4}$ | $6.29 * 10^{-4}$ | $5.32 * 10^{-4}$ | $5.55 * 10^{-4}$ |

Taking a closer look at Figure 4.13, we can observe that, in the subfigures for the approximations with one trained x-coordinate and one trained y-coordinate referential value, there are severe discrepancies between the surfaces of the model predictions and the semitransparent cyan surfaces of the function observations. The only trained x-coordinate referential value located near to -3.779310, i.e., the x-coordinate of the local minimum (-3.779310, -3.283186), and the only trained y-coordinate referential values located near to -3.283186, i.e., the y-coordinate of local minimum (-3.779310, -3.283186), help the models provide a reasonable fit to the observations of the Himmelblau function in the area between x=-5 and x=-6 and y=-5 and y=-6, while the models do not provide a suitable fit to the observations in the other areas, as there are insufficient trained referential values in the corresponding ranges of the x-axis and y-axis. This suggests that one trained x-coordinate referential value and one trained y-coordinate referential value are obviously not enough for the models to provide an adequate fit to the Himmelblau function.

From the subfigures of Figure 4.13 showing the approximations based on two trained x-coordinate referential values and two trained y-coordinate referential values, it can be seen there is an extra trained x-coordinate referential value and an extra trained y-coordinate referential value, which are generally located near to 3.584428, i.e., the x-coordinate of local minimum (3.584428, -1.848126) and 3.131312, i.e., the y-coordinate of local minimum (-2.805118, 3.131312) respectively.

This significantly improves the fit of the models to the Himmelblau function, although the surfaces of the model predictions in the approximations with two trained x-coordinate referential values and two trained y-coordinate referential values are generally flatter than the semitransparent cyan surfaces of observations of the Himmelblau function, and there are significant discrepancies between the surfaces of predictions and surfaces of observations.

As shown in the subfigures illustrating the approximations with two trained x-coordinate and two trained y-coordinate referential values, and those with three of each, both of these have trained x-coordinate referential values situated near to the local minima at (-3.779310, -3.283186) and (3.584428, -1.848126) respectively and trained y-coordinate referential values situated near to the local minima at (-3.779310, -3.283186) and (-2.805118, 3.131312) respectively.

Additionally, compared to the approximations with two trained x-coordinate and two trained y-coordinate referential values, those with three have an extra trained x-coordinate referential value and an extra trained y-coordinate referential value which are generally located near to the local maxima (where $z = 181.616$) at (-0.270844, -0.923038) and (-0.270844, -0.923038) respectively.

As a result of these extra trained referential values, the surfaces of the predictions of the models based on three trained x-coordinate and three trained y-coordinate referential values are more curved and hence have a closer fit to the semitransparent cyan surfaces of observations of the Himmelblau function than those with two of each.

Likewise, the approximations with three trained x-coordinate and three trained y-coordinate referential values and those with four of each share a number of trained referential values at similar locations near the extrema of the Himmelblau function. Compared with the approximations with three, those with four trained x-coordinate and four trained y-coordinate referential values have an additional trained x-coordinate referential value located near to a local minimum $z=0$ at (3.0, 2.0) and an additional trained y-coordinate referential value located near to a local minimum $z=0$ at (3.584428, -1.848126), which leads to the models providing a better fit to the corresponding parts of the surface generated by the observations of the Himmelblau function.

In a like manner, by comparing the approximations with four and five trained x-coordinate and y-coordinate referential values, we find both have a number of trained referential values at similar locations near the x-coordinates or y-coordinates of the extrema of the Himmelblau function. In addition, the approximations with five of each have an extra trained x-coordinate referential value located near to local minimum (-2.805118, 3.131312) and an extra trained y-coordinate referential value located near to local minimum (3.0, 2.0).

Due to the existence of these extra trained referential values, the models in the approximations with five trained x-coordinate and five trained y-coordinate referential values provide a better fit to the corresponding parts of the surface generated by the observations of the Himmelblau function than those with four of each.

Furthermore, as indicated in Figure 4.13, there are no significant changes on the surfaces of the predictions of the models in the approximations with more than five trained x-coordinate and more than five trained y-coordinate referential values, compared to those with five of each, despite the fact that the extra trained referential values improve the fit of the models to some parts of the surface generated by the observations of the Himmelblau function.

From the above analysis, it can safely be concluded that five trained x-coordinate referential values and five trained y-coordinate referential values are the minimum requirements for the MAKER models to provide a good fit to the Himmelblau function, as the function has four local minima and one local maximum and each of these extrema has a corresponding trained x-coordinate referential value located near to the x-coordinate of this extremum and a corresponding trained y-coordinate referential value located near to the y-coordinate of this extremum. Additionally, through optimization, these trained referential values are generally located near to the x-coordinates or the y-coordinates of the extrema of the Himmelblau function.

### 4.4.3 Local Minima and Maxima

As stated previously, in order to evaluate the accuracy of the MAKER model approximations to the Himmelblau function, the local minima and local maxima can be identified from the predicted z-coordinate values generated by the models and we can compare these extrema to those generated by the observed output values of the Himmelblau function to find out whether their locations are close to each other.

Table 4.11 displays the coordinates of the extrema of the predicted z-coordinate values generated by the models, which are closest to the corresponding extrema generated by the Himmelblau function among the relevant extrema of the models. We can observe that no local maximum located near to (-0.270844, -0.923038), i.e., the coordinates of the local maximum of the surface generated by the observed output values of the Himmelblau function, can be found from the surface generated by the predicted output values of the MAKER model approximations based on less than three trained x-coordinate referential values and less than three trained y-coordinate referential values. In these approximations, only a local minimum could be identified from the surface made up of the predicted output values from the models.

From those predicted output values in some of the approximations with three trained x-coordinate referential values and three trained y-coordinate referential values, we can identify not only two local minima but also a local maximum, located near to (-0.270844, -0.923038), i.e., the local maximum of the Himmelblau function.

This is generally due to the fact that both the approximations with two trained x-coordinate referential values and two trained y-coordinate referential values and

those with three of each have trained x-coordinate referential values located near to the local minima (z=0) at (-3.779310, -3.283186) and (3.584428, -1.848126) respectively and trained y-coordinate referential values located near to the local minima (z=0) at (-3.779310, -3.283186) and (-2.805118, 3.131312) respectively.

Despite the similarities of the trained referential values of these two approximations, compared to the approximations with two trained x-coordinate referential values and two trained y-coordinate referential values, those with three of each have extra ones located near local maxima (z=$181.616$) at (-0.270844, -0.923038) and (-0.270844, -0.923038) respectively. In addition, Table 4.11 shows there are four local minima near to (3.0, 2.0), (3.584428, -1.848126), (-2.805118, 3.131312), and (-3.779310, -3.283186), respectively, which are the coordinates of the local minima of the Himmelblau function.

A local maximum located near to (-0.270844, -0.923038), i.e., the coordinates of the local maximum of the Himmelblau function, can be identified from the predicted output values of the models in some of the approximations with five trained x-coordinate referential values and five trained y-coordinate referential values. This can be explained by the fact that the approximations with five of each have trained x-coordinate referential values located near to 3.0, 3.584428, -2.805118, -3.779310, and -0.270844, respectively, which are the x-coordinates of the extrema (both local minima and local maxima) of the Himmelblau function, and trained y-coordinate referential values located near to 2.0, -1.848126, 3.131312, -3.283186, and -0.923038, respectively, which are the y-coordinates of the extrema of the Himmelblau function.

**Table 4.11 The extrema of the predicted output values of the models in the approximations with different numbers of trained x-coordinate (nrvx), y-coordinate (nrvy), and z-coordinate (nrvz) referential values, which are closest to the corresponding extrema of the Himmelblau function among the relevant extrema of the models**

| nrvx & nrvy & nrvz\extremums | local minimum 1 at (3.0, 2.0) | local minimum 2 at about (3.6, -1.8) | local minimum 3 at about (-2.8, 3.1) | local minimum 4 at about (-3.8, -3.3) | local maximum at about (-0.3, -0.9) |
|---|---|---|---|---|---|
| 1 & 1 & 1 | (-3.2, -3.0) | (-3.2, -3.0) | (-3.2, -3.0) | (-3.2, -3.0) | (-6.0, -6.0) |
| 2 & 2 & 1 | (-4.2, -0.4) | (-4.2, -0.4) | (-4.2, -0.4) | (-4.2, -0.4) | (-6.0, -6.0) |
| 3 & 3 & 1 | (4.2, -0.2) | (4.2, -0.2) | (-4.0, 1.4) | (-4.0, -1.6) | (-0.4, -0.2) |
| 4 & 4 & 1 | (2.8, 1.4) | (3.6, -3.0) | (-4.2, 2.6) | (-4.2, -2.6) | (0.4, -0.6) |
| 5 & 5 & 1 | (3.6, 0.8) | (3.6, -2.2) | (-3.8, 3.2) | (-3.8, -2.4) | (0.0, -0.4) |
| 6 & 6 & 1 | (3.8, 0.8) | (3.8, -1.6) | (-2.4, 3.6) | (-3.6, -2.8) | (-0.4, -0.4) |
| 7 & 7 & 1 | (3.2, 1.2) | (3.4, -1.8) | (-3.2, 1.4) | (-3.2, -2.0) | (-0.2, 0.0) |
| 8 & 8 & 1 | (3.2, 1.6) | (2.6, -2.2) | (-3.2, 3.4) | (-4.4, -3.0) | (-1.0, -1.0) |
| 9 & 9 & 1 | (3.2, 2.0) | (3.2, -2.4) | (-3.4, 2.4) | (-3.4, -2.2) | (0.4, -0.8) |
| 10 & 10 & 1 | (2.4, 2.4) | (3.0, -2.4) | (-3.0, 2.2) | (-3.4, -2.8) | (0.0, -1.6) |
| 11 & 11 & 1 | (3.6, 2.4) | (3.6, -2.4) | (-2.8, 2.6) | (-3.6, -2.6) | (-0.6, -0.6) |
| 12 & 12 & 1 | (3.6, 2.0) | (3.6, -1.0) | (-3.2, 2.6) | (-3.6, -2.6) | (0.0, 0.6) |
| 13 & 13 & 1 | (3.0, 1.8) | (3.0, -2.4) | (-3.0, 1.8) | (-3.6, -3.2) | (-0.2, -0.8) |
| 14 & 14 & 1 | (2.4, 2.2) | (3.2, -1.8) | (-3.4, 3.8) | (-4.0, -3.2) | (-0.4, -0.4) |
| 15 & 15 & 1 | (2.8, 1.8) | (3.0, -2.6) | (-3.2, 2.4) | (-3.8, -3.8) | (0.0, -0.6) |
| 16 & 16 & 1 | (3.0, 1.8) | (4.2, -1.4) | (-3.4, 2.4) | (-3.8, -3.4) | (-0.8, -0.8) |
| 1 & 1 & 2 | (-3.8, -2.8) | (-3.8, -2.8) | (-3.8, -2.8) | (-3.8, -2.8) | (-6.0, -6.0) |
| 2 & 2 & 2 | (-2.6, 3.8) | (-2.6, 3.8) | (-2.6, 3.8) | (-2.6, 3.8) | (-6.0, -6.0) |
| 3 & 3 & 2 | (4.2, 2.4) | (4.2, -2.2) | (-4.0, 3.2) | (-4.0, -2.6) | (-0.2, 0.2) |
| 4 & 4 & 2 | (4.0, -0.2) | (4.0, -0.2) | (-3.4, 3.2) | (-3.4, 3.2) | (1.4, -6.0) |
| 5 & 5 & 2 | (4.0, 3.0) | (4.0, -0.4) | (-3.4, 3.4) | (-3.6, -3.2) | (0.0, 0.8) |
| 6 & 6 & 2 | (3.2, 2.6) | (3.2, -1.4) | (-3.2, 3.0) | (-3.2, -2.6) | (0.2, 0.4) |
| 7 & 7 & 2 | (3.6, 2.6) | (3.6, -2.4) | (-2.8, 3.0) | (-3.0, -2.6) | (0.6, -1.4) |
| 8 & 8 & 2 | (3.0, 2.0) | (3.0, -2.0) | (-3.4, 2.6) | (-3.6, -3.0) | (0.2, 0.2) |
| 9 & 9 & 2 | (3.2, 2.0) | (3.2, -1.2) | (-2.6, 2.2) | (-3.6, -3.4) | (-0.6, 0.2) |
| 10 & 10 & 2 | (2.8, 1.6) | (3.8, -2.2) | (-2.8, 2.8) | (-3.4, -2.8) | (-0.2, -0.6) |
| 11 & 11 & 2 | (2.8, 1.6) | (3.4, -2.0) | (-1.8, 2.8) | (-3.6, -3.0) | (-0.8, -0.4) |
| 12 & 12 & 2 | (3.0, 2.0) | (3.2, -2.4) | (-3.2, 2.6) | (-3.6, -2.4) | (-0.6, -1.4) |
| 13 & 13 & 2 | (2.2, 2.6) | (4.0, -2.0) | (-2.6, 2.2) | (-3.8, -3.0) | (-0.2, 0.0) |
| 14 & 14 & 2 | (3.4, 1.8) | (3.4, -2.0) | (-2.6, 2.4) | (-3.8, -3.0) | (-1.2, -0.2) |
| 15 & 15 & 2 | (3.0, 2.4) | (3.4, -2.2) | (-3.0, 3.0) | (-3.4, -3.0) | (-0.2, -0.8) |
| 16 & 16 & 2 | (3.6, 2.6) | (3.6, -1.0) | (-2.8, 2.8) | (-3.8, -3.0) | (0.2, -0.4) |
| 1 & 1 & 3 | (-4.2, -2.8) | (-4.2, -2.8) | (-4.2, -2.8) | (-4.2, -2.8) | (-6.0, -6.0) |
| 2 & 2 & 3 | (-2.0, 3.6) | (-2.0, 3.6) | (-2.0, 3.6) | (-2.0, 3.6) | (-6.0, -6.0) |

Continued on the next page

Continued from the previous page

| nrvx & nrvy & nrvz\extremums | local minimum 1 at (3.0, 2.0) | local minimum 2 at about (3.6, -1.8) | local minimum 3 at about (-2.8, 3.1) | local minimum 4 at about (-3.8, -3.3) | local maximum at about (-0.3, -0.9) |
|---|---|---|---|---|---|
| 3 & 3 & 3 | (4.2, -0.4) | (4.2, -0.4) | (-4.0, 3.0) | (-4.0, -2.0) | (-0.2, 2.2) |
| 4 & 4 & 3 | (4.0, 0.8) | (4.0, -1.0) | (-4.0, 2.8) | (-4.0, -3.8) | (0.2, 0.0) |
| 5 & 5 & 3 | (3.8, 2.2) | (3.8, -1.4) | (-3.2, 3.0) | (-3.2, -1.6) | (-0.2, -0.4) |
| 6 & 6 & 3 | (3.6, 2.2) | (3.6, -1.8) | (-2.6, 3.0) | (-2.8, -2.4) | (0.2, 0.2) |
| 7 & 7 & 3 | (3.4, 1.6) | (3.6, -1.6) | (-3.4, 3.2) | (-3.6, -3.0) | (0.0, 0.0) |
| 8 & 8 & 3 | (3.2, 1.0) | (3.2, -2.4) | (-2.2, 2.8) | (-3.8, -3.4) | (0.0, -0.4) |
| 9 & 9 & 3 | (3.4, 1.0) | (3.6, -2.0) | (-3.0, 2.6) | (-3.6, -3.4) | (0.2, -1.0) |
| 10 & 10 & 3 | (3.0, 2.0) | (3.4, -2.2) | (-2.2, 2.6) | (-3.6, -2.4) | (0.0, -0.4) |
| 11 & 11 & 3 | (3.4, 1.8) | (3.4, -1.8) | (-2.6, 2.6) | (-3.6, -2.8) | (-1.8, -0.8) |
| 12 & 12 & 3 | (2.8, 2.2) | (2.8, -1.4) | (-3.0, 3.0) | (-3.4, -3.0) | (-0.4, -0.6) |
| 13 & 13 & 3 | (3.2, 2.0) | (3.2, -1.2) | (-3.2, 2.6) | (-3.6, -2.6) | (-0.6, -0.2) |
| 14 & 14 & 3 | (2.8, 1.2) | (2.8, -1.2) | (-3.0, 2.6) | (-3.8, -3.4) | (-1.0, -0.4) |
| 15 & 15 & 3 | (3.0, 2.6) | (3.0, -2.4) | (-3.8, 3.4) | (-3.8, -2.8) | (-0.8, -1.6) |
| 16 & 16 & 3 | (2.6, 1.8) | (3.8, -1.8) | (-3.0, 3.2) | (-3.4, -2.4) | (0.2, 0.4) |
| 1 & 1 & 4 | (-4.0, -3.2) | (-4.0, -3.2) | (-4.0, -3.2) | (-4.0, -3.2) | (-6.0, -6.0) |
| 2 & 2 & 4 | (2.4, 3.8) | (2.4, 3.8) | (2.4, 3.8) | (2.4, 3.8) | (-6.0, -6.0) |
| 3 & 3 & 4 | (4.0, 0.0) | (4.0, 0.0) | (-3.4, 3.4) | (-4.0, 0.4) | (-6.0, -6.0) |
| 4 & 4 & 4 | (4.0, 2.2) | (4.0, 2.2) | (-3.2, 1.4) | (-3.2, 1.4) | (-6.0, -6.0) |
| 5 & 5 & 4 | (3.2, 2.4) | (3.2, -2.0) | (-4.2, 3.0) | (-4.2, -3.2) | (-0.8, 0.4) |
| 6 & 6 & 4 | (2.8, 0.8) | (3.0, -0.8) | (-2.8, 3.0) | (-3.2, -3.6) | (-1.0, -6.0) |
| 7 & 7 & 4 | (2.8, 1.2) | (3.4, -0.8) | (-2.6, 3.2) | (-3.8, -3.2) | (-0.4, -0.6) |
| 8 & 8 & 4 | (3.8, 2.2) | (3.8, -1.6) | (-2.8, 3.0) | (-3.4, -3.2) | (0.0, 0.0) |
| 9 & 9 & 4 | (3.4, 2.0) | (3.6, -2.2) | (-3.0, 2.8) | (-3.6, -3.0) | (-0.4, -1.0) |
| 10 & 10 & 4 | (3.0, 1.8) | (3.8, -3.0) | (-3.6, 3.0) | (-3.6, -3.0) | (-0.2, 0.8) |
| 11 & 11 & 4 | (2.6, 2.2) | (3.4, -1.8) | (-3.0, 2.4) | (-3.8, -3.0) | (0.2, -1.4) |
| 12 & 12 & 4 | (3.0, 2.0) | (3.6, -2.4) | (-2.4, 2.0) | (-2.8, -2.4) | (-0.4, -0.8) |
| 13 & 13 & 4 | (3.2, 1.6) | (3.6, -1.8) | (-2.8, 3.0) | (-3.4, -3.2) | (-0.6, -1.4) |
| 14 & 14 & 4 | (2.8, 2.2) | (3.2, -1.8) | (-3.0, 2.6) | (-3.4, -3.4) | (0.0, -0.6) |
| 15 & 15 & 4 | (3.0, 1.8) | (3.4, -2.6) | (-3.0, 2.2) | (-3.8, -3.2) | (0.0, -0.6) |
| 16 & 16 & 4 | (3.0, 1.6) | (3.0, -1.6) | (-3.2, 3.0) | (-3.2, -3.0) | (0.0, -1.2) |
| 1 & 1 & 5 | (-4.2, -3.0) | (-4.2, -3.0) | (-4.2, -3.0) | (-4.2, -3.0) | (-6.0, -6.0) |
| 2 & 2 & 5 | (-3.2, 3.8) | (-3.2, 3.8) | (-3.2, 3.8) | (-3.2, 3.8) | (-6.0, -6.0) |
| 3 & 3 & 5 | (4.0, -1.2) | (4.0, -1.2) | (-4.0, -2.6) | (-4.0, -2.6) | (-6.0, -6.0) |
| 4 & 4 & 5 | (4.0, -1.0) | (4.0, -1.0) | (-3.2, 3.2) | (-3.2, -0.4) | (-6.0, -6.0) |
| 5 & 5 & 5 | (2.8, 0.4) | (4.4, -1.8) | (-3.6, 3.0) | (-3.6, -2.8) | (-6.0, -6.0) |
| 6 & 6 & 5 | (3.8, 1.4) | (3.8, -0.4) | (-3.2, 2.4) | (-3.2, -3.2) | (-0.2, 0.6) |
| 7 & 7 & 5 | (3.4, 2.4) | (3.8, -2.0) | (-3.0, 2.4) | (-3.4, -3.0) | (0.2, 0.2) |
| 8 & 8 & 5 | (3.2, 1.6) | (3.2, -2.2) | (-2.8, 2.8) | (-3.2, -2.8) | (0, -0.2) |

Continued on the next page

Continued from the previous page

| nrvx & nrvy & nrvz\extremums | local minimum 1 at (3.0, 2.0) | local minimum 2 at about (3.6, -1.8) | local minimum 3 at about (-2.8, 3.1) | local minimum 4 at about (-3.8, -3.3) | local maximum at about (-0.3, -0.9) |
|---|---|---|---|---|---|
| 9 & 9 & 5 | (2.8, 2.0) | (2.6, -1.8) | (-2.6, 2.0) | (-3.2, -2.8) | (-1.4, -1.2) |
| 10 & 10 & 5 | (3.2, 1.8) | (3.4, -2.8) | (-2.8, 2.8) | (-3.2, -2.8) | (-0.8, -0.2) |
| 11 & 11 & 5 | (2.8, 2.0) | (3.4, -2.0) | (-2.8, 2.8) | (-3.8, -3.2) | (-0.8, -1.4) |
| 12 & 12 & 5 | (3.0, 2.2) | (4.0, -2.8) | (-2.4, 3.2) | (-3.4, -3.0) | (-0.4, -1.0) |
| 13 & 13 & 5 | (3.2, 0.6) | (3.4, -2.2) | (-2.6, 3.0) | (-3.6, -3.2) | (-1.2, -0.8) |
| 14 & 14 & 5 | (3.4, 1.6) | (3.4, -1.4) | (-3.0, 2.6) | (-3.6, -3.0) | (-1.0, -1.0) |
| 15 & 15 & 5 | (2.0, 3.0) | (3.4, -2.6) | (-3.4, 3.0) | (-3.4, -2.8) | (0.2, -1.6) |
| 16 & 16 & 5 | (3.6, 0.8) | (3.6, -2.0) | (-3.0, 2.6) | (-3.6, -3.4) | (0.0, -0.6) |
| 1 & 1 & 6 | (3.2, -2.4) | (3.2, -2.4) | (3.2, -2.4) | (3.2, -2.4) | (6.0, -6.0) |
| 2 & 2 & 6 | (-4.2, 3.8) | (-4.2, 3.8) | (-4.2, 3.8) | (-4.2, 3.8) | (-5.8, -6.0) |
| 3 & 3 & 6 | (4.0, -2.6) | (4.0, -2.6) | (-4.0, 0.0) | (-4.0, 0.0) | (-6.0, -6.0) |
| 4 & 4 & 6 | (4.2, 1.4) | (4.2, -0.6) | (-3.4, 3.8) | (-3.6, -3.0) | (-4.0, -6.0) |
| 5 & 5 & 6 | (3.0, 0.2) | (3.0, 0.2) | (-3.4, 2.8) | (-3.4, -2.0) | (0.2, -6.0) |
| 6 & 6 & 6 | (2.4, 2.6) | (3.6, -0.8) | (-2.2, 2.6) | (-4.2, -3.4) | (1.0, 0.0) |
| 7 & 7 & 6 | (3.4, 1.6) | (3.4, 0.6) | (-3.2, 3.0) | (-3.6, -2.4) | (-0.4, 1.2) |
| 8 & 8 & 6 | (3.0, 1.8) | (3.0, -2.4) | (-3.4, 2.0) | (-3.6, -2.8) | (-0.4, 0.2) |
| 9 & 9 & 6 | (3.0, 2.4) | (3.2, -1.6) | (-3.2, 3.2) | (-4.0, -3.4) | (-1.2, -0.6) |
| 10 & 10 & 6 | (3.2, 0.8) | (3.4, -1.8) | (-3.2, 2.4) | (-3.8, -3.0) | (0.0, -0.2) |
| 11 & 11 & 6 | (2.6, 2.6) | (3.4, -1.8) | (-3.4, 3.6) | (-3.6, -2.8) | (0.0, -1.4) |
| 12 & 12 & 6 | (3.0, 2.4) | (3.0, -2.4) | (-2.6, 3.0) | (-3.6, -3.0) | (-0.2, -1.6) |
| 13 & 13 & 6 | (2.8, 2.4) | (3.2, -2.2) | (-2.6, 3.0) | (-3.4, -2.8) | (0.2, -1.4) |
| 14 & 14 & 6 | (3.2, 1.6) | (3.2, -2.0) | (-3.4, 3.0) | (-3.4, -3.0) | (0.6, 0.4) |
| 15 & 15 & 6 | (2.6, 2.6) | (2.8, -1.6) | (-3.0, 3.0) | (-3.4, -3.0) | (0.2, -0.8) |
| 16 & 16 & 6 | (2.8, 1.4) | (2.8, -2.0) | (-3.4, 3.0) | (-3.4, -3.0) | (-0.2, -1.6) |

The previous analysis suggests that increasing the number of trained referential values in the models could effectively improve the accuracy of the MAKER model approximations to the Himmelblau function. From Table 4.11, we can see that the extrema generated by the models in the approximations with more than five trained x-coordinate referential values and more than five trained y-coordinate referential values, which are nearest to their counterparts for the Himmelblau function among the relevant extrema of the models, are generally closer to the corresponding

extrema of the Himmelblau function than the extrema generated by the models in the approximations with five of each, which are nearest to their counterparts for the Himmelblau among the relevant extrema of the model.

Table 4.12 exhibits some of the extrema generated by the models in the approximations with more than five trained x-coordinate and y-coordinate referential values, which are nearest to their counterparts for the Himmelblau function among the relevant extrema of the models. In Table 4.12, nrvx, nrvy, and nrvz stand for number of the trained x-coordinate, y-coordinate, and z-coordinate referential values respectively.

From Table 4.12, we can clearly see that some of the extrema generated by the models in the approximations with more than five trained x-coordinate referential values and more than five trained y-coordinate referential values, which are nearest to their counterparts for the Himmelblau function among the relevant extrema of the models, are generally close to the corresponding extrema of the Himmelblau function.

**Table 4.12 Some of the extrema of the predicted output values of the models in the approximations with more than five trained x-coordinate, y-coordinate, and z-coordinate referential values, which are closest to the corresponding extrema of the Himmelblau function**

| nrvx & nrvy & nrvz\extremums | local minimum 1 at (3.0, 2.0) | local minimum 2 at about (3.6, -1.8) | local minimum 3 at about (-2.8, 3.1) | local minimum 4 at about (-3.8, -3.3) | local maximum at about (-0.3, -0.9) |
|---|---|---|---|---|---|
| 11 & 11 & 1 | (3.6, 2.4) | (3.6, -2.4) | (-2.8, 2.6) | (-3.6, -2.6) | (-0.6, -0.6) |
| 15 & 15 & 2 | (3.0, 2.4) | (3.4, -2.2) | (-3.0, 3.0) | (-3.4, -3.0) | (-0.2, -0.8) |
| 12 & 12 & 3 | (2.8, 2.2) | (2.8, -1.4) | (-3.0, 3.0) | (-3.4, -3.0) | (-0.4, -0.6) |
| 13 & 13 & 4 | (3.2, 1.6) | (3.6, -1.8) | (-2.8, 3.0) | (-3.4, -3.2) | (-0.6, -1.4) |
| 11 & 11 & 5 | (2.8, 2.0) | (3.4, -2.0) | (-2.8, 2.8) | (-3.8, -3.2) | (-0.8, -1.4) |
| 12 & 12 & 6 | (3.0, 2.4) | (3.0, -2.4) | (-2.6, 3.0) | (-3.6, -3.0) | (-0.2, -1.6) |

As the intervals between any two adjacent x-coordinate values or any two y-coordinate values of the data set for the approximations are set to 0.2, the accuracy of the MAKER model approximations to the Himmelblau function could be improved even further if the intervals were made smaller than 0.2. This is because smaller intervals would lead to more data pairs in the data set for the approximations, which could lead to more accurate approximations to the Himmelblau function. As a result, the extrema generated by the models in the approximations, which are nearest to their counterparts for the Himmelblau function among the relevant extrema of the models, could be even closer to the corresponding extrema of the Himmelblau function.

## 4.5 Stopping Criteria for the Training of the Models

Generally speaking, the higher the complexity of a model is, the more accurate its approximation will be (Coello Coello, Hernández Aguirre and Zitzler, 2005). However, overly complex models will lead to overfitting (Matignon, 2005). Although an overfitted model may have a small error (or bias), the error will be heavily dependent on the data, as an overfitted model reflects noise in data (Hanrahan, 2009). In other words, the error of an overfitted model has high variance (Hanrahan, 2009), which indicates that it might not be generalizable. On the contrary, a model with a large error will be less sensitive to the data and hence have reduced variance (Hanrahan, 2009), which means it could be more generalizable than one with a small error. However, a model with a large error cannot provide sufficiently accurate predictions for the data. Ideally, the optimal model will have a low error and a low variance, but this is often not the case for statistical models based on a finite sample of noisy data (Hanrahan, 2009). This kind of dilemma between underfitting and overfitting is referred to as the bias-variance dilemma (Geman, Bienenstock and Doursat, 1992). Therefore, we need to take the trade-off between model accuracy and model complexity into consideration in selecting a model with enough complexity to achieve the best generalization (Matignon, 2005).

In this research, taking that trade-off into account means we need to stop training the MAKER models once the training results satisfy some given criteria. In the following part of this section, we will summarize the findings for the univariate function MAKER model approximations to obtain some criteria for stopping training the models.

### 4.5.1 Exponential Function

We start from the advanced learning approximations for the exponential function $y = 6^x$. As mentioned previously, from Fig 4.5, we can see that the solid red curves representing the predictions of the MAKER models and the solid cyan curves representing the observations of the function generally match well in all subfigures, regardless of the number of trained x-coordinate and y-coordinate referential values, and the locations of these trained referential values. From Table 4.5, which shows the MSEs for the advanced learning approximations to $y = 6^x$, we can observe that there is little difference between the MSE for the approximation based on zero trained x-coordinate and zero trained y-coordinate referential values, and the MSE for the approximations with more than zero of each. Thus, we can conclude that there is no need to include extra referential values in addition to the four boundary referential values (i.e. the minimum and the maximum of the observed input values (x) of the function and the global extrema of the observed output values (y) of the function) when approximating the function $y = 6^x$ using MAKER models.

To provide a more intuitive picture of the findings from the approximations to the function $y = 6^x$, we can take advantage of the ratios between the moving averages of the MSEs for different approximations, in the advanced learning case. The moving average is simply the average of several successive data values (Jani,

2014). The series of moving averages of the original observations is smoother than the series of original observations (Wegner, 2010). A moving average removes the effect of irregular fluctuations in the original observations (Wegner, 2010). It helps the decision maker to focus more on the general trend of changes in the observations, without the obscuring effect of noise (Wegner, 2010). Thus, we can use the ratios of moving averages to describe the relative changes in the MSEs for the advanced learning approximations to $y = 6^x$, to reflect the findings from the approximations.

**Table 4.13 The 3-elements moving averages as we move along the dimension of number of trained x-coordinate referential values (nrvx) of MSEs for approximations with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values in the advanced learning for $y = 6^x$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0.001351 | 0.001878 | 0.001763 | 0.001494 | 0.000852 | 0.000916 |
| 1 | 0.000921 | 0.000901 | 0.000897 | 0.000874 | 0.000820 | 0.000792 |
| 2 | 0.000848 | 0.000841 | 0.000850 | 0.000812 | 0.000752 | 0.000723 |
| 3 | 0.000699 | 0.000815 | 0.000928 | 0.000971 | 0.000814 | 0.000751 |
| 4 | 0.000641 | 0.000603 | 0.000726 | 0.000593 | 0.000594 | 0.000595 |

**Table 4.14 The 3-elements moving averages as we move along the dimension of number of trained y-coordinate referential values (nrvy) of MSEs for approximations with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values in the advanced learning for $y = 6^x$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0.000977 | 0.001296 | 0.001896 | 0.000800 | 0.000857 | 0.000851 |
| 1 | 0.000912 | 0.001168 | 0.001539 | 0.000804 | 0.000837 | 0.000782 |
| 2 | 0.000757 | 0.000888 | 0.000911 | 0.000876 | 0.000871 | 0.000639 |
| 3 | 0.000537 | 0.000922 | 0.000801 | 0.000782 | 0.000795 | 0.000584 |
| 4 | 0.000414 | 0.000927 | 0.000788 | 0.000767 | 0.000793 | 0.000553 |

On the basis of Table 4.5, which shows the MSEs for the advanced learning approximations for $y = 6^x$, we can generate Tables 4.13 and 4.14, which show the 3-elements moving averages as we move along the dimension of number of trained x-coordinate or y-coordinate referential values (nrvx) of MSEs for approximations

with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values in the advanced learning for $y = 6^x$. It should be noted that, when there are less than three elements (values) in the window for the computation of a geometric moving average, at the endpoints of Table 4.5, we just take the geometric average over the elements (values) that are available.

With the moving averages in Tables 4.13 and 4.14, we can further calculate the ratios of moving averages along the dimension of the number of trained x-coordinate or y-coordinate referential values, to describe the trend of changes in these moving averages in detail. Tables 4.15 and 4.16 present the ratios of the moving averages of the MSEs for the advanced learning approximations based on different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values for $y = 6^x$, to those for the approximations with one less x-coordinate or y-coordinate referential value

From Table 4.15, it can be seen that the ratios in Table 4.14 as we move along the dimension of the number of trained x-coordinate referential values (nrvx) generally fluctuate between 0.8 and 1.2. From Table 4.16, we can see that the ratios of the ratios in Table 4.15 as we move along the dimension of the number of trained y-coordinate referential values (nrvy) generally lie between 0.8 and 1.2.

**Table 4.15 The ratios of the moving averages of the MSEs for the advanced learning approximations based on different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values for $y = 6^x$, to those for the approximations with one less x-coordinate referential value**

| nrvy\nrvx | 1 to 0 | 2 to 1 | 3 to 2 | 4 to 3 | 5 to 4 |
|---|---|---|---|---|---|
| 0 | 1.389835 | 0.939109 | 0.847070 | 0.570185 | 1.074951 |
| 1 | 0.977923 | 0.996299 | 0.974368 | 0.937476 | 0.965636 |
| 2 | 0.991745 | 1.010702 | 0.955686 | 0.925728 | 0.960771 |
| 3 | 1.165951 | 1.138650 | 1.046695 | 0.838023 | 0.921990 |
| 4 | 0.941238 | 1.203315 | 0.817264 | 1.001685 | 1.001122 |

**Table 4.16 The ratios of the moving averages of the MSEs for the advanced learning approximations based on different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values for $y = 6^x$, to those for the approximations with one less y-coordinate referential value**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 to 0 | 0.934289 | 0.901325 | 0.812099 | 1.004583 | 0.977622 | 0.919851 |
| 2 to 1 | 0.830106 | 0.760491 | 0.592031 | 1.090004 | 1.039809 | 0.816787 |
| 3 to 2 | 0.709067 | 1.037913 | 0.878566 | 0.892314 | 0.912711 | 0.913928 |
| 4 to 3 | 0.770019 | 1.005244 | 0.983555 | 0.981237 | 0.997273 | 0.946918 |

Combining the findings from Tables 4.15 and 4.16 with the findings from Figure 4.5 and Table 4.5 as previously mentioned, we can conclude that, if the ratios of moving averages of MSEs from Table 4.15 along the dimension of the number of trained x-coordinate or y-coordinate referential values is between 0.8 and 1.2, there is no need to include extra referential values in addition to the available referential values for the MAKER model approximations to $y = 6^x$.

For example, as can be seen from Table 4.15, the ratio of the moving average of the MSEs for the approximation with two trained x-coordinate referential values (xrv) and zero trained y-coordinate referential values (yrv) to that for the approximation with one less x-coordinate referential value is 0.939109, which is between 0.8 and 1.2. The ratio of the moving average of the MSEs for the approximation with three trained xrv and zero trained yrv to that for the

approximation with one less xrv is 0.847070, which is still between 0.8 and 1.2. Hence, we can just stop training the models along the dimension of the number of trained x-coordinate referential values at the model with one trained x-coordinate referential values and zero trained y-coordinate referential values. From Table 4.16, we can clearly see that the ratio of moving average of the MSEs for the approximation with zero xrv and one yrv to that for the approximation with one less yrv is 0.934289, which is between 0.8 and 1.2, and the ratio of moving average of the MSEs for the approximation with zero xrv and two yrv to that for the approximation with one less yrv is 0.830106, which is still between 0.8 and 1.2. Thus, we can stop training the models along the dimension of the number of trained y-coordinate referential values at the model with zero trained x-coordinate referential values and zero trained y-coordinate referential values.

### 4.5.2 Logarithmic Function

The next group of approximations we focus on is the advanced learning approximations for the logarithmic function $y = \log_6 x$. As previously stated, from Figure 4.6, it is clear that the solid red curves representing the predictions of the MAKER models generally coincide well with the solid cyan curves representing the observations of the function, in all subfigures, regardless of the number of trained x-coordinate and y-coordinate referential values, and their locations. From Table 4.6, it can be observed that there is little difference between the MSE for the approximation with zero trained x-coordinate and y-coordinate referential values and those for approximations with more than zero of them, with the MSEs of the latter being slightly smaller than the MSE of the former. However, this slight improvement in the accuracy of the models is achieved at the cost of increased model complexity. Hence, we need to consider the trade-off between the models' accuracy and complexity.

Similarly, we use the moving averages of the MSEs for the advanced learning approximations for the logarithmic function $y = \log_6 x$ to more precisely describe the trend in the changes in the MSEs of the approximations, from which we can observe the relationship between the accuracy and complexity of the models. Tables 4.17 and 4.18 show these moving averages for the advanced learning approximations for $y = \log_6 x$ as we move along the dimension of the number of trained x-coordinate or y-coordinate referential values.

**Table 4.17 The 3-elements moving averages as we move along the dimension of number of trained x-coordinate referential values (nrvx) of MSEs for approximations with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values in the advanced learning for $y = \log_6 x$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | $1.89* 10^{-5}$ | $1.62* 10^{-5}$ | $1.10* 10^{-5}$ | $8.79* 10^{-6}$ | $8.00* 10^{-6}$ | $7.78* 10^{-6}$ |
| 1 | $9.45* 10^{-6}$ | $8.68* 10^{-6}$ | $5.64* 10^{-6}$ | $5.95* 10^{-6}$ | $5.02* 10^{-6}$ | $5.84* 10^{-6}$ |
| 2 | $4.22* 10^{-6}$ | $6.00* 10^{-6}$ | $7.03* 10^{-6}$ | $7.12* 10^{-6}$ | $5.50* 10^{-6}$ | $4.84* 10^{-6}$ |
| 3 | $5.30* 10^{-6}$ | $6.01* 10^{-6}$ | $6.64* 10^{-6}$ | $7.39* 10^{-6}$ | $6.44* 10^{-6}$ | $5.57* 10^{-6}$ |
| 4 | $7.43* 10^{-6}$ | $7.68* 10^{-6}$ | $6.81* 10^{-6}$ | $5.98* 10^{-6}$ | $4.41* 10^{-6}$ | $4.13* 10^{-6}$ |

**Table 4.18 The 3-elements moving averages as we move along the dimension of number of trained y-coordinate referential values (nrvy) of MSEs for approximations with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values in the advanced learning for $y = \log_6 x$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | $1.83* 10^{-5}$ | $1.01* 10^{-5}$ | $8.92* 10^{-6}$ | $5.92* 10^{-6}$ | $7.29* 10^{-6}$ | $6.33* 10^{-6}$ |
| 1 | $1.34* 10^{-5}$ | $8.30* 10^{-6}$ | $9.13* 10^{-6}$ | $6.22* 10^{-6}$ | $6.51* 10^{-6}$ | $5.79* 10^{-6}$ |
| 2 | $7.51* 10^{-6}$ | $5.14* 10^{-6}$ | $8.04* 10^{-6}$ | $6.14* 10^{-6}$ | $6.29* 10^{-6}$ | $4.54* 10^{-6}$ |
| 3 | $5.88* 10^{-6}$ | $5.42* 10^{-6}$ | $8.39* 10^{-6}$ | $6.67* 10^{-6}$ | $5.43* 10^{-6}$ | $4.25* 10^{-6}$ |
| 4 | $6.95* 10^{-6}$ | $5.78* 10^{-6}$ | $7.81* 10^{-6}$ | $6.59* 10^{-6}$ | $5.67* 10^{-6}$ | $4.02* 10^{-6}$ |

On the basis of these ratios in Table 4.17 and Table 4.18, we can generate the ratios of moving averages of MSEs for the advanced learning approximations for $y = \log_6 x$ to describe the trend in the changes in these moving averages specifically, as shown in Tables 4.19 and 4.20. Tables 4.19 and 4.20 displays the ratios of the moving averages of the MSEs for the advanced learning approximations based on different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values

for $y = \log_6 x$, to those for the approximations with one less x-coordinate or y-coordinate referential value

From Table 4.19, we can see that the ratios of moving averages of MSEs in Table 4.17 along the dimension of the number of trained x-coordinate referential values (nrvx) generally range between 0.8 and 1.2. From Table 4.20, it can be seen that the ratios of moving averages of MSEs in Table 4.18 along the dimension of the number of trained y-coordinate referential values (nrvy) are generally between 0.8 and 1.2.

By integrating the findings from Tables 4.19 and 4.20 with the findings from Figure 4.6 and Table 4.6, it can be concluded that, if the ratios of moving averages of the MSEs from Tables 4.19 or 4.20 as we move along the dimension of the number of trained x-coordinate or y-coordinate referential values are between 0.55 and 1.2, we do not need to use extra referential values in addition to the available referential values when approximating the function $y = \log_6 x$ using the MAKER framework.

For instance, as shown in Table 4.19, the ratio of moving average of the MSEs for the approximation with one trained x-coordinate referential value (xrv) and zero trained y-coordinate referential values (yrv) to that for the approximation with one less x-coordinate referential value is 0.855379, which is between 0.55 and 1.2. The ratio of moving average of the MSEs for the approximation with two trained xrv and zero trained yrv to that for the approximation with one less yrv is 0.679175, and it is between 0.55 and 1.2. Hence, the training of the models along the dimension of the number of trained x-coordinate referential values can be stopped at the model with zero trained x-coordinate and zero trained y-coordinate referential values.

**Table 4.19 The ratios of the moving averages of the MSEs for the advanced learning approximations based on different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values for $y = \log_6 x$, to those for the approximations with one less x-coordinate referential value**

| nrvy\nrvx | 1 to 0 | 2 to 1 | 3 to 2 | 4 to 3 | 5 to 4 |
|---|---|---|---|---|---|
| 0 | 0.855379 | 0.679175 | 0.800546 | 0.909746 | 0.972280 |
| 1 | 0.918166 | 0.650019 | 1.055556 | 0.843785 | 1.162575 |
| 2 | 1.421801 | 1.171111 | 1.012808 | 0.773302 | 0.879467 |
| 3 | 1.133333 | 1.105993 | 1.112895 | 0.871055 | 0.864130 |
| 4 | 1.034096 | 0.885900 | 0.879040 | 0.737604 | 0.934668 |

**Table 4.20 The ratios of the moving averages of the MSEs for the advanced learning approximations based on different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values for $y = \log_6 x$, to those for the approximations with one less y-coordinate referential value**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 to 0 | 0.735160 | 0.821452 | 1.024117 | 1.051564 | 0.893617 | 0.915219 |
| 2 to 1 | 0.559752 | 0.619124 | 0.880248 | 0.986602 | 0.966206 | 0.783659 |
| 3 to 2 | 0.783400 | 1.054510 | 1.043965 | 1.086909 | 0.863805 | 0.936858 |
| 4 to 3 | 1.181303 | 1.067077 | 0.930274 | 0.988006 | 1.043558 | 0.945141 |

From Table 4.20, we can observe that the ratio of the moving average of the MSEs for the approximation with zero trained x-coordinate referential value (xrv) and one trained y-coordinate referential values (yrv) to that for the approximation with one less y-coordinate referential value is 0.735160, which is between 0.55 and 1.2. The ratio of the moving average of the MSEs for the approximation with zero trained xrv and two trained yrv to that for the approximation with one less trained yrv is 0.559752, which is still between 0.55 and 1.2. Thus, the training of the models along the dimension of the number of trained y-coordinate referential values can be terminated at the model with zero trained x-coordinate referential values and zero trained y-coordinate referential value.

### 4.5.3 Power Function

Now, we turn to the advanced learning approximations for the power function $y = x^{\frac{1}{6}}$. As described previously, from the analysis of Figure 4.7, we can deduce that the number and locations of trained y-coordinate referential values do not have a significant effect on the accuracy of the MAKER-based model approximations. From the analysis of Table 4.7 and Figure 4.7, we can conclude that a trained x-coordinate referential value located in the range where the function has a large mean curvature is very important in improving the accuracy of MAKER-based model approximations to functions with large mean curvature.

**Table 4.21 The 3-elements moving averages as we move along the dimension of number of trained x-coordinate referential values (nrvx) of MSEs for approximations with numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values in the advanced learning for $y = x^{\frac{1}{6}}$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | $9.89 * 10^{-4}$ | $6.86 * 10^{-4}$ | $8.69 * 10^{-5}$ | $4.85 * 10^{-5}$ | $3.08 * 10^{-5}$ | $2.68 * 10^{-5}$ |
| 1 | $9.64 * 10^{-4}$ | $6.53 * 10^{-4}$ | $4.33 * 10^{-5}$ | $2.50 * 10^{-5}$ | $1.79 * 10^{-5}$ | $1.58 * 10^{-5}$ |
| 2 | $9.47 * 10^{-4}$ | $6.45 * 10^{-4}$ | $5.12 * 10^{-5}$ | $2.21 * 10^{-5}$ | $1.21 * 10^{-5}$ | $1.11 * 10^{-5}$ |
| 3 | $9.33 * 10^{-4}$ | $6.37 * 10^{-4}$ | $4.60 * 10^{-5}$ | $2.30 * 10^{-5}$ | $1.25 * 10^{-5}$ | $1.25 * 10^{-5}$ |
| 4 | $9.85 * 10^{-4}$ | $6.65 * 10^{-4}$ | $5.08 * 10^{-5}$ | $1.90 * 10^{-5}$ | $1.33 * 10^{-5}$ | $1.23 * 10^{-5}$ |

**Table 4.22 The 3-elements moving averages of MSEs as we move along the dimension of number of trained y-coordinate referential values (nrvy) for approximations with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values in the advanced learning for $y = x^{\frac{1}{6}}$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | $1.84 * 10^{-3}$ | $1.09 * 10^{-4}$ | $5.57 * 10^{-5}$ | $3.04 * 10^{-5}$ | $2.42 * 10^{-5}$ | $1.84 * 10^{-5}$ |
| 1 | $1.83 * 10^{-3}$ | $1.05 * 10^{-4}$ | $5.10 * 10^{-5}$ | $2.50 * 10^{-5}$ | $1.96 * 10^{-5}$ | $1.62 * 10^{-5}$ |
| 2 | $1.81 * 10^{-3}$ | $8.50 * 10^{-5}$ | $3.92 * 10^{-5}$ | $1.63 * 10^{-5}$ | $1.46 * 10^{-5}$ | $1.16 * 10^{-5}$ |
| 3 | $1.81 * 10^{-3}$ | $9.65 * 10^{-5}$ | $3.75 * 10^{-5}$ | $1.40 * 10^{-5}$ | $1.27 * 10^{-5}$ | $1.12 * 10^{-5}$ |
| 4 | $1.82 * 10^{-3}$ | $9.59 * 10^{-5}$ | $3.55 * 10^{-5}$ | $1.39 * 10^{-5}$ | $1.38 * 10^{-5}$ | $1.10 * 10^{-5}$ |

In order to identify the trade-off between the accuracy and the complexity of the models, we again use the moving averages of the MSEs for the advanced learning for the power function $y = x^{\frac{1}{6}}$ to describe in detail the trend in the changes in the MSEs for the approximations. Tables 4.21 and 4.22 shows the moving averages of MSEs as we move along the dimension of number of trained x-coordinate or y-coordinate referential values of MSEs for approximations with different numbers of trained x-coordinate and y-coordinate referential values in the advanced learning for $y = x^{\frac{1}{6}}$.

**Table 4.23 The ratios of the moving averages of the MSEs for the advanced learning approximations based on different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values for $y = x^{\frac{1}{6}}$, to those for the approximations with one less x-coordinate referential value**

| nrvy\nrvx | 1 to 0 | 2 to 1 | 3 to 2 | 4 to 3 | 5 to 4 |
|---|---|---|---|---|---|
| 0 | 0.693630 | 0.126725 | 0.557899 | 0.635052 | 0.870130 |
| 1 | 0.677521 | 0.066327 | 0.577692 | 0.714647 | 0.885131 |
| 2 | 0.681305 | 0.079279 | 0.432573 | 0.548193 | 0.910714 |
| 3 | 0.682638 | 0.072285 | 0.499638 | 0.543478 | 0.996000 |
| 4 | 0.675533 | 0.076295 | 0.374918 | 0.696322 | 0.930332 |

**Table 4.24 The ratios of the moving averages of the MSEs for the advanced learning approximations based on different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values for $y = x^{\frac{1}{6}}$, to those for the approximations with one less y-coordinate referential value**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 to 0 | 0.991504 | 0.964318 | 0.915619 | 0.823465 | 0.811295 | 0.876413 |
| 2 to 1 | 0.990702 | 0.806452 | 0.769281 | 0.651132 | 0.745331 | 0.717351 |
| 3 to 2 | 1.001288 | 1.134902 | 0.955820 | 0.858896 | 0.865604 | 0.968076 |
| 4 to 3 | 1.004595 | 0.994126 | 0.945333 | 0.989286 | 1.085526 | 0.983066 |

Based on these moving averages in Table 4.21 and Table 4.22, we can generate the corresponding ratios of moving averages of MSEs as we did with the other functions. These are shown in Tables 4.23 and 4.24 respectively.

From Table 4.23, it can be observed that the ratios of the moving averages of MSEs in Table 4.21 as we move along the dimension of the number of trained x-coordinate referential values decrease from about 0.68 to about 0.10, and then increase to about 0.5 and keep increasing before reaching about 0.90 which is for the ratios of the moving averages of the MSEs for the approximations with five trained x-coordinate referential values, to those for the approximations with one less x-coordinate referential value. From Table 4.24, we can observe that the ratios of moving averages of MSEs in Table 4.22 along the dimension of the number of trained y-coordinate referential values (nrvy) generally range between 0.8 and 1.2.

By incorporating the findings from Tables 4.23 and 4.24 with the findings from Figure 4.7 and Table 4.7, we can reach the conclusion that, if a ratio of the moving averages of the MSEs from Table 4.23 or Table 4.24 and its next corresponding ratio as we move along the dimension of the number of trained x-coordinate or y-coordinate referential values (nrvx) are both between 0.6 and 1.2, we do not need extra referential values in addition to the available referential values when approximating the function $y = x^{\frac{1}{6}}$ using the MAKER framework.

For example, as can be found from Table 4.23, the ratio of the moving average of the MSEs for the approximation with four trained x-coordinate referential value (rvx) and zero trained y-coordinate referential values (rvy) to that for the approximation with one less trained x-coordinate referential values is 0.635052, and the next ratio as we move along the dimension of number of trained x-coordinate referential values is 0.870130, which are both between 0.6 and 1.2. Hence, the training of the models along the dimension of the number of trained x-coordinate referential

values can be stopped at the model with three trained x-coordinate referential values and zero trained y-coordinate referential values.

From Table 4.24, it is clear that the ratio of the moving average of the MSEs for the approximation with zero trained x-coordinate referential value (xrv) and one trained y-coordinate referential values (yrv) to that for the approximation with one less trained y-coordinate referential value is 0.991504, and the next ratio as we move along the dimension of number of trained y-coordinate referential values is 0.990702, which are both between 0.6 and 1.2. Thus, the training of the models along the dimension of the number of trained y-coordinate referential values can be terminated at the model with zero trained xrv and zero trained yrv.

### 4.5.4 Basic Non-monotonic Function

In the previous parts of this section, we have summarized the findings from the monotonic univariate functions, i.e., $\mathrm{y} = 6^x$, $y = \log_6 x$, and $y = x^{\frac{1}{6}}$. We now proceed to the advanced learning approximations for the basic non-monotonic univariate function $y = -(x - 0.5)^2 + 0.25$. As previously stated, from the analysis of Table 4.8 and Figure 4.8, we can draw the conclusion that the number and locations of trained y-coordinate referential values do not have a significant impact on the accuracy of the MAKER model approximations. From the analysis of Table 4.8 and Figure 4.8, it can be concluded that one trained x-coordinate referential value is sufficient for the MAKER-based models to accurately approximate the basic non-monotonic univariate function $y = -(x - 0.5)^2 + 0.25$ which has only one critical point and hence two monotone intervals, and that this trained x-coordinate referential value must generally be located at the x-coordinate of the critical point of the function in order to minimize the difference between the predicted output values of the model and the observed output values of the function.

In order to quantify the findings from the aforementioned tables and figures, we once again take advantage of the moving averages of the MSEs for the advanced learning approximations for the basic non-monotonic univariate function $y = -(x - 0.5)^2 + 0.25$ to describe the trend in the changes in the MSEs of the approximations. Tables 4.25 and 4.26 present these moving averages as we move along the dimension of the number of trained x-coordinate or y-coordinate referential values.

**Table 4.25 The 3-elements moving averages of MSEs as we move along the dimension of number of trained x-coordinate referential values (nrvx) for approximations with different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values in the advanced learning for $y = -(x - \mathbf{0.5})^2 + \mathbf{0.25}$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | $2.12 * 10^{-3}$ | $1.42 * 10^{-3}$ | $2.48 * 10^{-6}$ | $2.91 * 10^{-6}$ | $3.76 * 10^{-6}$ | $4.66 * 10^{-6}$ |
| 1 | $2.11 * 10^{-3}$ | $1.41 * 10^{-3}$ | $1.79 * 10^{-6}$ | $2.27 * 10^{-6}$ | $2.50 * 10^{-6}$ | $3.02 * 10^{-6}$ |
| 2 | $2.10 * 10^{-3}$ | $1.40 * 10^{-3}$ | $2.51 * 10^{-6}$ | $2.47 * 10^{-6}$ | $3.01 * 10^{-6}$ | $3.00 * 10^{-6}$ |
| 3 | $2.09 * 10^{-3}$ | $1.40 * 10^{-3}$ | $2.66 * 10^{-6}$ | $2.85 * 10^{-6}$ | $3.27 * 10^{-6}$ | $3.01 * 10^{-6}$ |
| 4 | $2.09 * 10^{-3}$ | $1.39 * 10^{-3}$ | $2.26 * 10^{-6}$ | $2.71 * 10^{-6}$ | $3.32 * 10^{-6}$ | $3.38 * 10^{-6}$ |

**Table 4.26 The 3-elements moving averages of MSEs as we move along the dimension of number of trained y-coordinate referential values (nrvy) for approximations with different combinations of number of trained x-coordinate referential values (nrvx) and number of trained y-coordinate referential values (nrvy) in the advanced learning for $y = -(x - \mathbf{0.5})^2 + \mathbf{0.25}$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | $4.23 * 10^{-3}$ | $2.32 * 10^{-6}$ | $2.38 * 10^{-6}$ | $1.71 * 10^{-6}$ | $3.67 * 10^{-6}$ | $4.01 * 10^{-6}$ |
| 1 | $4.22 * 10^{-3}$ | $2.29 * 10^{-6}$ | $2.34 * 10^{-6}$ | $2.15 * 10^{-6}$ | $3.15 * 10^{-6}$ | $3.96 * 10^{-6}$ |
| 2 | $4.20 * 10^{-3}$ | $2.18 * 10^{-6}$ | $2.02 * 10^{-6}$ | $2.76 * 10^{-6}$ | $2.81 * 10^{-6}$ | $3.21 * 10^{-6}$ |
| 3 | $4.19 * 10^{-3}$ | $2.24 * 10^{-6}$ | $1.86 * 10^{-6}$ | $3.34 * 10^{-6}$ | $2.83 * 10^{-6}$ | $3.43 * 10^{-6}$ |
| 4 | $4.18 * 10^{-3}$ | $2.24 * 10^{-6}$ | $1.66 * 10^{-6}$ | $3.50 * 10^{-6}$ | $3.19 * 10^{-6}$ | $3.20 * 10^{-6}$ |

On the basis of the moving averages of the MSEs in Tables 4.25 and 4.26, we can generate the ratios of the moving averages of the MSEs to reveal the trend in the changes in these ratios specifically. Tables 4.27 and 4.28 show these ratios as we move along the dimension of the number of trained x-coordinate or y-coordinate referential values.

From Table 4.27, it can be observed that the ratios of the moving averages of the MSEs in Table 4.25 as we move along the dimension of the number of trained x-coordinate referential values first decrease from about 0.66 to about 0.0016, and then increase to about 1 and remain stable. From Table 4.28, it can be seen that the ratios of the moving averages of the MSEs in Table 4.26 as we move along the dimension of the number of trained y-coordinate referential values (nrvy) generally fluctuate between 0.8 and 1.3.

**Table 4.27 The ratios of the moving averages of the MSEs for the advanced learning approximations based on different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values for $y = -(x - 0.5)^2 + 0.25$, to those for the approximations with one less x-coordinate referential value**

| nrvy\nrvx | 1 to 0 | 2 to 1 | 3 to 2 | 4 to 3 | 5 to 4 |
|---|---|---|---|---|---|
| 0 | 0.667138 | 0.001754 | 1.170470 | 1.293578 | 1.239362 |
| 1 | 0.666945 | 0.001272 | 1.268657 | 1.101471 | 1.207610 |
| 2 | 0.667026 | 0.001791 | 0.985372 | 1.218623 | 0.996678 |
| 3 | 0.666990 | 0.001908 | 1.068836 | 1.148712 | 0.920489 |
| 4 | 0.666871 | 0.001624 | 1.198822 | 1.223587 | 1.018072 |

**Table 4.28 The ratios of the moving averages of the MSEs for the advanced learning approximations based on different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values for $y = -(x - 0.5)^2 + 0.25$, to those for the approximations with one less y-coordinate referential value**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 to 0 | 0.997476 | 0.987761 | 0.983193 | 1.257310 | 0.859219 | 0.989596 |
| 2 to 1 | 0.995414 | 0.953353 | 0.861823 | 1.283721 | 0.890063 | 0.809924 |
| 3 to 2 | 0.997378 | 1.025994 | 0.920661 | 1.210145 | 1.009501 | 1.067497 |
| 4 to 3 | 0.998925 | 1.001490 | 0.891382 | 1.046407 | 1.125882 | 0.933852 |

Combining the findings from Tables 4.27 and 4.28 with those from Figure 4.8 and Table 4.8, we can conclude that, if a ratio of the moving averages of the MSEs from Table 4.27 or Table 4.28 and its next corresponding ratio as we move along the dimension of the number of trained x-coordinate or y-coordinate referential values (nrvx) are both between 0.8 and 1.3, we do not need extra referential values in

addition to the available referential values when approximating the function $y = -(x - 0.5)^2 + 0.25$ using the MAKER framework.

For instance, as shown in Table 4.27, the ratio of the moving average of the MSEs for the approximation with three trained x-coordinate referential value and zero trained y-coordinate referential values to that for the approximation with one less trained x-coordinate referential value is 1.170470, and the next ratio as we move along the dimension of number of trained x-coordinate referential values is 1.293578, which are both between 0.8 and 1.3. Hence, the training of the models along the dimension of the number of trained x-coordinate referential values can be stopped at the model with two trained x-coordinate referential values and zero trained y-coordinate referential values.

From Table 4.28, we can see clearly that the ratio of the moving average of the MSEs for the approximation with zero trained x-coordinate referential value and one trained y-coordinate referential values to that for the approximation with one less trained y-coordinate referential value is 0.997476, and the next ratio as we move along the dimension of number of trained y-coordinate referential values is 0.995414, which are both between 0.8 and 1.3. Thus, the training of the models along the dimension of the number of trained y-coordinate referential values can be terminated at the model with zero trained x-coordinate referential values and zero trained y-coordinate referential values.

### 4.5.5 Complex Non-monotonic Function

On the basis of the summary of the findings for the basic non-monotonic univariate function $y = -(x - 0.5)^2 + 0.25$ , we move on to the advanced learning approximations for the complex non-monotonic univariate function $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$ which has three critical points and hence four monotone intervals. As mentioned previously, from the analysis for Table 4.9, Figure 4.9, and Figure 4.10, we can conclude that there is no significant relationship between the number and the locations of the trained y-coordinate referential values and the accuracy of the MAKER model approximations. From the analysis for Table 4.9, Figure 4.9, Figure 4.10, and Figure 4.11, it can be deduced that six or seven trained x-coordinate referential values is sufficient for the MAKER-based models to accurately approximate the complex non-monotonic univariate function $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$.

For the purpose of quantifying the findings from Table 4.9, Figure 4.10, and Figure 4.11, in the same way, we take advantage of the moving averages of the MSEs for the advanced learning approximations for the function $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$ to describe the trend in the changes in the MSEs for the approximations. Tables 4.29 and 4.30 show these moving averages as we move along the dimension of the number of trained x-coordinate or y-coordinate referential values.

Based on the moving averages in Tables 4.29 and 4.30, we can calculate the ratios of the moving averages of the MSEs to reflect the trend in the changes in these ratios in detail. Tables 4.31 and 4.32 exhibit these ratios as we move along the dimension of the number of trained x-coordinate or y-coordinate referential values.

From Table 4.31, we can see that the ratios of the moving averages of the MSEs in Table 4.29 as we move along the dimension of the number of trained x-coordinate

referential values fluctuate between 0.1 and 0.8, and then increase to about 0.5 and keep increasing before reaching about 1.0 which is for the ratios of the moving averages of the MSEs for the approximations with ten trained x-coordinate referential values, to those for the approximations with one less x-coordinate referential value. From Table 4.32, it can be seen that the ratios of the moving averages of the MSEs in Table 4.30 as we move along the dimension of the number of trained y-coordinate referential values (nrvy) generally fluctuate between 0.8 and 1.2.

By integrating the findings from Tables 4.31 and 4.32 with those from Table 4.9, Figure 4.9, Figure 4.10, and Figure 4.11, we can conclude that, if a ratio of the moving averages of the MSEs from Table 4.29 or Table 4.30 and its next corresponding ratio as we move along the dimension of the number of trained x-coordinate or y-coordinate referential values are both between 0.6 and 1.2, we do not need extra referential values in addition to the available referential values when approximating the function $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$ using the MAKER framework.

For example, from Table 4.31, we can see clearly that the ratio of the moving average of the MSEs for the approximation with eight trained x-coordinate referential value and zero trained y-coordinate referential values to that for the approximation with one less trained x-coordinate referential value is 0.805792, and the next ratio as we move along the dimension of number of trained x-coordinate referential values is 0.635225, which are both between 0.6 and 1.2. Hence, the training of the models along the dimension of the number of trained x-coordinate referential values can be stopped at the model with seven trained x-coordinate referential values and zero trained y-coordinate referential values.

**Table 4.29 The 3-elements moving averages of MSEs as we move along the dimension of number of trained x-coordinate referential values (nrvx) for approximations with different combinations of number of trained x-coordinate referential values (nrvx) and number of trained y-coordinate referential values (nrvy) in the advanced learning for $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | $5.01*10^{-2}$ | $3.90*10^{-2}$ | $1.47*10^{-2}$ | $7.53*10^{-3}$ | $2.02*10^{-3}$ | $1.28*10^{-3}$ | $1.85*10^{-4}$ | $1.02*10^{-4}$ | $8.18*10^{-5}$ | $5.20*10^{-5}$ | $5.09*10^{-5}$ |
| 1 | $5.01*10^{-2}$ | $3.83*10^{-2}$ | $1.37*10^{-2}$ | $5.70*10^{-3}$ | $8.35*10^{-4}$ | $3.20*10^{-4}$ | $1.76*10^{-4}$ | $6.98*10^{-5}$ | $2.93*10^{-5}$ | $3.05*10^{-5}$ | $2.78*10^{-5}$ |
| 2 | $4.98*10^{-2}$ | $3.83*10^{-2}$ | $1.38*10^{-2}$ | $5.75*10^{-3}$ | $7.57*10^{-4}$ | $2.30*10^{-4}$ | $1.14*10^{-4}$ | $5.41*10^{-5}$ | $3.83*10^{-5}$ | $5.00*10^{-5}$ | $5.79*10^{-5}$ |
| 3 | $4.98*10^{-2}$ | $3.82*10^{-2}$ | $1.38*10^{-2}$ | $5.71*10^{-3}$ | $8.53*10^{-4}$ | $2.75*10^{-4}$ | $1.78*10^{-4}$ | $8.28*10^{-5}$ | $8.10*10^{-5}$ | $6.72*10^{-5}$ | $7.06*10^{-5}$ |
| 4 | $4.98*10^{-2}$ | $3.82*10^{-2}$ | $1.37*10^{-2}$ | $5.71*10^{-3}$ | $7.99*10^{-4}$ | $4.12*10^{-4}$ | $2.65*10^{-4}$ | $1.61*10^{-4}$ | $4.73*10^{-5}$ | $4.32*10^{-5}$ | $4.90*10^{-5}$ |
| 5 | $4.98*10^{-2}$ | $3.82*10^{-2}$ | $1.38*10^{-2}$ | $5.75*10^{-3}$ | $8.62*10^{-4}$ | $3.42*10^{-4}$ | $2.45*10^{-4}$ | $1.27*10^{-4}$ | $9.64*10^{-5}$ | $5.40*10^{-5}$ | $3.19*10^{-5}$ |
| 6 | $4.98*10^{-2}$ | $3.83*10^{-2}$ | $1.77*10^{-2}$ | $9.62*10^{-3}$ | $4.66*10^{-3}$ | $2.94*10^{-4}$ | $1.60*10^{-4}$ | $7.50*10^{-5}$ | $5.85*10^{-5}$ | $7.47*10^{-5}$ | $8.49*10^{-5}$ |

**Table 4.30 The 3-elements moving averages of MSEs as we move along the dimension of number of trained y-coordinate referential values (nrvy) for approximations with different combinations of number of trained x-coordinate referential values (nrvx) and number of trained y-coordinate referential values (nrvy) in the advanced learning for $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | $7.53*10^{-2}$ | $2.48*10^{-2}$ | $1.59*10^{-2}$ | $2.00*10^{-3}$ | $1.95*10^{-3}$ | $3.29*10^{-4}$ | $1.18*10^{-4}$ | $9.37*10^{-5}$ | $4.50*10^{-5}$ | $2.79*10^{-5}$ | $5.08*10^{-5}$ |
| 1 | $7.53*10^{-2}$ | $2.47*10^{-2}$ | $1.57*10^{-2}$ | $1.90*10^{-3}$ | $1.42*10^{-3}$ | $2.90*10^{-4}$ | $1.13*10^{-4}$ | $7.08*10^{-5}$ | $4.14*10^{-5}$ | $3.71*10^{-5}$ | $5.39*10^{-5}$ |
| 2 | $7.52*10^{-2}$ | $2.45*10^{-2}$ | $1.50*10^{-2}$ | $1.73*10^{-3}$ | $4.15*10^{-4}$ | $3.04*10^{-4}$ | $1.06*10^{-4}$ | $5.72*10^{-5}$ | $4.36*10^{-5}$ | $4.78*10^{-5}$ | $5.63*10^{-5}$ |
| 3 | $7.51*10^{-2}$ | $2.45*10^{-2}$ | $1.51*10^{-2}$ | $1.69*10^{-3}$ | $4.21*10^{-4}$ | $3.01*10^{-4}$ | $1.94*10^{-4}$ | $6.11*10^{-5}$ | $4.22*10^{-5}$ | $6.33*10^{-5}$ | $5.49*10^{-5}$ |
| 4 | $7.51*10^{-2}$ | $2.46*10^{-2}$ | $1.50*10^{-2}$ | $1.69*10^{-3}$ | $4.44*10^{-4}$ | $3.81*10^{-4}$ | $2.04*10^{-4}$ | $1.03*10^{-4}$ | $6.36*10^{-5}$ | $5.78*10^{-5}$ | $4.31*10^{-5}$ |
| 5 | $7.51*10^{-2}$ | $2.46*10^{-2}$ | $1.51*10^{-2}$ | $5.49*10^{-3}$ | $4.64*10^{-4}$ | $3.69*10^{-4}$ | $2.16*10^{-4}$ | $8.56*10^{-5}$ | $6.15*10^{-5}$ | $5.52*10^{-5}$ | $5.53*10^{-5}$ |
| 6 | $7.51*10^{-2}$ | $2.46*10^{-2}$ | $1.52*10^{-2}$ | $7.45*10^{-3}$ | $4.49*10^{-4}$ | $3.81*10^{-4}$ | $1.25*10^{-4}$ | $1.02*10^{-4}$ | $7.63*10^{-5}$ | $5.39*10^{-5}$ | $6.29*10^{-5}$ |

As can be observed from Table 4.32, the ratio of the moving average of the MSEs for the approximation with zero trained x-coordinate referential value and one trained y-coordinate referential values to that for the approximation with one less trained y-coordinate referential value is 0.999082, and the next ratio as we move along the dimension of number of trained y-coordinate referential values is 0.999628, which are both between 0.6 and 1.2. Thus, the training of the models along the dimension of the number of trained y-coordinate referential values can be terminated at the model with zero trained x-coordinate referential values and zero trained y-coordinate referential values.

**Table 4.31 The ratios of the moving averages of the MSEs for the advanced learning approximations based on different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values for $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$, to those for the approximations with one less x-coordinate referential value**

| nrvy\nrvx | 1 to 0 | 2 to 1 | 3 to 2 | 4 to 3 | 5 to 4 | 6 to 5 | 7 to 6 | 8 to 7 | 9 to 8 | 10 to 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.778770 | 0.377738 | 0.510935 | 0.267592 | 0.632768 | 0.144864 | 0.549635 | 0.805792 | 0.635225 | 0.979474 |
| 1 | 0.766181 | 0.358108 | 0.414879 | 0.146493 | 0.383786 | 0.549011 | 0.397081 | 0.419093 | 1.042141 | 0.911475 |
| 2 | 0.768257 | 0.360420 | 0.416636 | 0.131748 | 0.303257 | 0.494920 | 0.475953 | 0.707948 | 1.305483 | 1.157000 |
| 3 | 0.766526 | 0.360299 | 0.415180 | 0.149311 | 0.322400 | 0.647144 | 0.465243 | 0.979058 | 0.829700 | 1.049331 |
| 4 | 0.767606 | 0.359233 | 0.415813 | 0.139798 | 0.516277 | 0.642846 | 0.606137 | 0.294191 | 0.914669 | 1.132228 |
| 5 | 0.767741 | 0.360426 | 0.416868 | 0.150000 | 0.396365 | 0.717073 | 0.518776 | 0.758720 | 0.559972 | 0.589815 |
| 6 | 0.768296 | 0.461350 | 0.544535 | 0.484197 | 0.063198 | 0.544281 | 0.467957 | 0.780791 | 1.275626 | 1.136384 |

**Table 4.32 The ratios of the moving averages of the MSEs for the advanced learning approximations based on different numbers of trained x-coordinate (nrvx) and y-coordinate (nrvy) referential values for $y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}$, to those for the approximations with one less y-coordinate referential value**

| nrvy\nrvx | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 to 0 | 0.999082 | 0.996068 | 0.985104 | 0.947690 | 0.730559 | 0.882582 | 0.956827 | 0.755591 | 0.920741 | 1.330944 | 1.061024 |
| 2 to 1 | 0.999628 | 0.993076 | 0.958922 | 0.910535 | 0.291680 | 1.048623 | 0.935372 | 0.807815 | 1.052293 | 1.287253 | 1.045145 |
| 3 to 2 | 0.998020 | 0.999891 | 1.003219 | 0.977014 | 1.016077 | 0.990142 | 1.834120 | 1.067599 | 0.968654 | 1.324268 | 0.975148 |
| 4 to 3 | 0.999743 | 1.000530 | 0.998230 | 1.001186 | 1.054589 | 1.264381 | 1.048052 | 1.693231 | 1.505130 | 0.912586 | 0.785194 |
| 5 to 4 | 1.000000 | 1.000842 | 1.006074 | 3.249605 | 1.043511 | 0.968504 | 1.059440 | 0.827853 | 0.966964 | 0.954991 | 1.281298 |
| 6 to 5 | 1.000000 | 1.000020 | 1.001366 | 1.358076 | 0.967290 | 1.032520 | 0.577280 | 1.195093 | 1.241323 | 0.976133 | 1.137214 |

### 4.5.5 Stopping Criteria

According to Schweitzer (2002), stopping criteria are used to avoid overfitting and very slow convergence of the training of models. Once the stopping criteria are satisfied, the training of models will be terminated (Schweitzer, 2002). Taking all the summaries of findings from the univariate function approximations presented previously into consideration, we can obtain the stopping criterion for the training of the MAKER-based models for the univariate function approximations, Criterion (4.3):

If $0.8 \leq r_{i,i+1} \leq 1.2$ and $0.8 \leq r_{i+1,i+2} \leq 1.2\ (0 \leq i \leq N)$, then the training process of the MAKER-based models along the dimension of the number of trained x-coordinate or y-coordinate referential values is stopped at the MAKER-based model which has i trained x-coordinate or y-coordinate referential values. (4.3)

In Criterion (4.3), $r_{i,i+1} = \frac{MAMSE_{i+1}}{MAMSE_i}$, $r_{i,i+1}$ represents the ratio of $MAMSE_{i+1}$ to $MAMSE_i$.

$$MAMSE_i = \begin{cases} \frac{MSE_i + MSE_{i+1}}{2}, & i = 0 \\ \frac{MSE_{i-1} + MSE_i + MSE_{i+1}}{3}, & 0 < i < M \\ \frac{MSE_{i-1} + MSE_i}{2}, & i = M \end{cases}$$

$MAMSE_i$ denotes a moving average of MSEs for the approximations centred on the approximation of the MAKER-based model which has i trained x-coordinate or y-coordinate referential values. $MSE_i$ indicates the MSE for the approximation of the MAKER-based model which has i trained x-coordinate or y-coordinate referential values. As indicated in Section 4.5.1, there are several reasons to use moving averages in the stopping criteria. One is that the moving averages of the original observations are smoother than the original observations (Wegner, 2010). In addition, the moving averages remove the effect of irregular fluctuations in the original observations, and help the decision makers

focus more on the general changes in the observations than the obscuring of noise (Wegner, 2010). Hence, the ratios of moving averages can be used to describe the relative changes of the MSEs for the advanced learning approximations, from which we obtained the Criterion (4.3).

As there is a connection between continuous functions approximations and classification, which is mentioned in both Sections 4.1 and 5.6.1, Criterion (4.3) can be adapted to Criterion (4.4) applied to the classification of the data sets which will be shown in Chapters 5 and 6.

If $0.8 \leq r_{i,i+1} \leq 1.2$ and $0.8 \leq r_{i+1,i+2} \leq 1.2\ (i \geq 1)$, then the training process of the MAKER-based models is stopped at the MAKER-based models which all have i trained referential values for each of the input variables of cross-validated data sets. (4.4)

In Criterion (4.4), $r_{i,i+1} = \frac{MAMSE_{i+1}}{MAMSE_i}$. $r_{i,i+1}$ represents the ratio of $MAMSE_{i+1}$ to $MAMSE_i$. $MAMSE_i = \frac{MSE_i + MSE_{i+1}}{2}$. $MAMSE_i$ denotes a moving average of the average MSEs for the classification experiments of the MAKER-based models which have same number of trained referential values for each of the input variables. This moving average of the average MSEs is centred on the classification experiments of the MAKER-based models which all have i trained referential values for each of the input variables.

## 4.6 Summary

In this chapter, we have presented the referential-value-based data discretization techniques for transforming continuous data, one of the major contributions of this research. At the beginning of this chapter, we compared the referential-value-based data discretization method with mainstream data discretization methods, i.e.,

equal-width, equal-frequency, and multi-interval-entropy-minimization discretization to highlight the advantages of the referential-value-based data discretization method, which are related to reducing information loss and distortion, and we presented the characteristics of data. Then we used the MAKER-based models constructed by the referential-value-based data discretization method to approximate univariate functions and a bivariate function. From the MAKER-based model approximations, it was evident that the MAKER-based models approximated the functions well. From the findings, we obtained some stopping criteria for the training of the MAKER-based models to guide the experimental classification of data sets in the subsequent chapters.

# Chapter 5

# Rule-based Inferential Modelling and Prediction

## 5.1 Introduction

This chapter is dedicated to the rule-based inferential modelling and prediction approach based on the MAKER framework, which integrates statistical analysis, belief-rule-based inference, and machine learning. Interpretability is the most important characteristic of this approach. We take the classification of classical data sets as a case study to demonstrate how the MAKER-based classifier is constructed through rule-based inferential modelling and prediction, and compare the MAKER-based classifier to other modelling and prediction approaches. The rest of the chapter is divided into seven sections. Statistical analysis, belief-rule-based inference, and prediction and machine learning, which are the fundamental aspects of the rule-based inferential modelling and prediction approach, are introduced in Sections 5.2, 5.3, and 5.4 respectively. In Section 5.5, a comparative analysis is performed to identify the limitations of mainstream modelling and prediction approaches, and highlight the advantages of rule-based inferential modelling and prediction. Section 5.6 presents a case study in using rule-based inferential modelling and prediction to build a MAKER-based classifier for performing classification experiments on classical data sets, including the Banana data set, the Haberman's survival data set, and the Iris data set. Also, the classification results of the MAKER-based classifiers are compared to those of alternative classifiers, which are shown in Section 5.7. A summary of the chapter is provided in Section 5.8.

## 5.2 Statistical Analysis

According to Yang and Xu (2017), inference is a systematic process used to predict the outputs of a system from its inputs, in which correlation, dependence, and interaction among the inputs and between the inputs and outputs of the system are taken into account. If there is an explicit functional relationship between the inputs and outputs of a system, which is known a priori, inferences can be made by directly using the function to calculate the outputs for a given set of inputs (Yang and Xu, 2017). Otherwise, it is necessary to collect judgements or data about the behaviour of a system to generate inferences (Yang and Xu, 2017).

As stated by Yang and Xu (2017), the MAKER framework is a process of making data-driven inferences from inputs about outputs, under uncertainty. The MAKER framework involves two types of models, the state space model (SSM) and the evidence space model (ESM), and a conjunctive rule of evidential reasoning (Yang and Xu, 2017).

In an SSM, an output variable contains a number of states which comprise a system state space (Yang and Xu, 2017). In an SSM, it is assumed that a state space is composed of a finite number of states (Yang and Xu, 2017). The original thinking of Dempster (2008) on state spaces is the foundation of the SSM.

Here, following Yang and Xu (2017), we use $H_n$ to represent a system state and suppose that a system state space has at least N disjoint states which are denoted by $\Theta = \{H_1 \cdots H_n \cdots H_N \cdots\}$ under the condition that $H_i \cap H_j = \emptyset \; for \; any \; i \neq j$. We can assign a probability to a subset of system states. The collection of all subsets is known as the power set of Θ, which is denoted by P(Θ) or $2^{\Theta}$ (Yang and Xu, 2017). The full power set of Θ consists of the empty set $\emptyset$ and the full state space Θ (Yang and Xu, 2017).

As stated by Yang and Xu (2017), the output of a system can be modelled by a unique set function, which is referred to as a basic probability function. It is defined as an ordinary discrete probability distribution over the nonempty subsets of Θ (Yang and Xu, 2017), as shown in Definition 5.1.

**Definition 5.1 (Basic probability function):** A set function $p: 2^{\theta} \rightarrow [0,1]$ is referred to as a basic probability function, if conditions (5.1), (5.2), and (5.3) are satisfied.

$$0 \leq p(\theta) \leq 1 \; \forall \theta \subseteq \Theta \tag{5.1}$$

$$\sum_{\theta \subseteq \Theta} p(\theta) = 1 \tag{5.2}$$

$$p(\emptyset) = 0 \tag{5.3}$$

According to Yang and Xu (2017), in conditions (5.1), (5.2), and (5.3), θ is a subset of states, which is called an assertion. $p(\theta)$ is the basic probability of assertion θ being true. $p(\theta)$ is assigned exactly to assertion θ and it cannot be decomposed into pieces which are assigned to the subsets of θ (Yang and Xu, 2017).

The definition of system output is given by Definition 5.2.

**Definition 5.2 (System output):** A system output of y is defined as a probability distribution as shown in Equation (5.4).

$$y = \left\{ \left(\theta, p(\theta)\right), \forall \theta \subseteq \Theta \; and \; \sum_{\theta \sqsubseteq \Theta} p(\theta) = 1 \right\} \tag{5.4}$$

In Equation (5.4), $p(\theta)$ is generated from the inputs of the system using Equations (3.14) and (3.15). As stated by Yang and Xu (2017), θ is referred to as the focal element of y, if p(θ)>0.

Besides this, according to Dempster (2008), an assertion can be profiled by three nonnegative probabilities, i.e., $p^t, p^f, and \; p^u$, which are assigned to 'true', 'false', and 'unknown' and termed the triad of an assertion. In an SSM, as previously stated,

an output can be described by a basic probability function, while each member of the power set in the output of the system can be profiled by a triplet of $p^t, p^f, and\ p^u$ more specifically (Yang and Xu, 2017).

How likely an assertion in the state space is to be true relies on the degree to which it is supported by evidence (Yang and Xu, 2017). An evidence space is a space derived from data (Yang and Xu, 2017). In the evidence space, each piece of evidence is acquired from data and each piece of evidence can be partitioned into evidential elements (Yang and Xu, 2017). Each evidential element points exactly to an element in the power set of the system states or an assertion in the state space (Yang and Xu, 2017).

The process of evidence acquisition from data for a single input variable is established based on the likelihood principle and the Bayesian principle (Yang and Xu, 2017). According to Welsh (1996), likelihood contains all the information in the data. Hence, if two likelihoods for a parameter θ are proportional, the inferences should be identical for θ regardless of which likelihood we use (Welsh, 1996). This is formally known as the likelihood principle (Welsh, 1996). A concept highly related to the likelihood principle is the likelihood function. According to Held and Sabanés Bové (2014), the likelihood function L(θ) or simply the likelihood is the probability mass or the density function of the observed data x, which is viewed as a function of the unknown parameter θ. The Bayesian principle in this research is that the combination of the evidence and the prior distribution of the system states leads to the posterior probability (Yang and Xu, 2014).

Through evidence acquisition, we can construct a one-dimensional ESM for each input variable. For the sake of clarity, in the following part of this section, we use $e_{i,l}$ to represent the $i^{th}$ piece of evidence from the $l^{th}$ input variable $x_l$ and $e_{i,l}(\theta)$ to represent an element of $e_{i,l}$ which points exactly to assertion θ. The basic probability of evidence $e_{i,l}$ pointing to assertion θ is represented by $p_{\theta,i,l} = p_l(e_{i,l}(\theta))$ and we use $c_{\theta,i,l}$ to represent the likelihood of the $i^{th}$ value of $x_l$ pointing to

assertion θ (Yang and Xu, 2017). In order to generate probabilistic inference, $\mathrm{p}_{\theta,\mathrm{i},\mathrm{l}}$ needs to be acquired from the normalized likelihood as shown in Equation (5.5), which was illustrated in Equation (3.7).

$$p_{\theta,i,l} = \frac{c_{\theta,i,l}}{\sum_{A\subseteq\Theta} c_{A,i,l}} \; \forall\theta \subseteq \Theta \tag{5.5}$$

Given probability $p_{\theta,i,l}$ which is acquired from input variable $x_l$ for each assertion θ, we can use $e_{i,l}$ as a system input, which is defined as a probability distribution as shown in Definition 5.3.

**Definition 5.3 (System input):** A system input is a basic probability distribution as exhibited in Equation (5.6).

$$e_{i,l} = \left\{\left(e_{i,l}(\theta), p_{\theta,i,l}\right), \forall\theta \subseteq \Theta \; and \; \sum_{\theta\subseteq\Theta} p_{\theta,i,l} = 1\right\} \tag{5.6}$$

According to Yang and Xu (2017), in Equation (5.6), $p_{\theta,i,j}$ is derived from input variable $x_l$ using Equation (5.5). The evidential elements $e_{i,l}(H_n)$ for all $H_n \in \Theta$ form the subspace for the $i^{th}$ value of $x_l$. In this research, $x_l$ is continuous. Hence, we can use the method introduced in Section 3.5.1 to discretize $x_l$ and follow the above-mentioned procedure.

Based on the likelihood principle and the Bayesian principle, we can acquire the joint basic probability from the joint likelihood function to generate an interdependence index to analyse the statistical interdependence between two pieces of evidence, as was elaborated in Section 3.5.2.

## 5.3 Belief Rule-Base Inference

According to Jackson (1998), an expert system is a computer system which mimics the decision-making ability of a human expert in the field of artificial intelligence. Expert systems are designed to solve complex problems using reasoning and knowledge (Pattnaik, Swetapadma and Sarraf, 2018). Among a wide range of alternative means of knowledge representation, rules are one of the most common forms and are used to express various types of knowledge for a number of reasons (Sun, 1995). It has been argued by some researchers (Hayes, 1977; Chomsky, 1980; Nilsson, 1984) that other knowledge representation methods can be transformed into logic (rule)-based methods. As stated by Grosan and Abraham (2011), a rule-based system, in which the definitions depend almost entirely on expert systems, is a way of encoding human experts' knowledge in a narrow area into an automated system, using rules to represent and code the knowledge. The rules of a rule-based system are expressed as a set of IF-THEN rules, which are a set of facts, and some interpreters controlling the application of the rules according to the facts (Grosan and Abraham, 2011). Rule-based systems, as very simple models, can be adapted and applied to a large number of problems (Grosan and Abraham, 2011). A rule-based system is generally composed of two essential parts, namely a knowledge base and an inference engine, which are combined to infer useful conclusions from observed facts provided by users and rules established by experts (Yang et al., 2006).

According to Tang et al. (2011), a conventional rule base of a rule-based system is composed of simple IF-THEN rules. The $k^{th}$ rule can be written in the following form:

$$R_k: if\, A_1^k \wedge A_2^k \wedge \ldots \wedge A_{T_k}^k, then\, D_k \tag{5.7}$$

In (5.7), $A_i^k(i = 1,2, \dots, T_k)$ is a referential value of the $i^{th}$ antecedent attribute in the $k^{th}$ rule. The referential value can take different types of values. $T_k$ is the number of antecedent attributes used in the $k^{th}$ rule. $D_k$ is the consequence of the $k^{th}$ rule. The symbol ∧ indicates the relationship 'AND' between the antecedent attributes. The general form of rule displayed in (5.7) is simple, as it does not take account of the distribution of consequences, the relative importance of each antecedent, or the relative importance of rules in the rule base (Tang et al., 2011).

To take these aspects into account, three concepts, i.e., degree of belief in the consequence, attribute weight, and rule weight, are used.

Degree of belief in the consequence: In complex cases, it is possible that the consequence of a rule could take a number of values with different degrees of belief, in order to express the viewpoints of experts on the extent to which a certain consequence may be true (Tang et al., 2011). Suppose the consequence D of a rule has N different values, i.e., $D_1$, $D_2$, ... , $D_N$, and the degree of belief in $D_i$ is denoted by $\beta_i$ (i=1,2,...,N). Then, the consequence of a rule with that belief structure could be represented by $(D_1, \beta_1)$, $(D_2, \beta_2)$, … , $(D_N, \beta_N)$.

Attribute weight: As indicated by Yang et al. (2006), the relative importance of an attribute to the consequence of a rule is important in rule-based inference. Hence, a weight is assigned to each attribute to describe this (Yang et al., 2006).

Rule weight: The relative importance of a rule to the whole rule base plays an important role in rule-based inference (Yang et al., 2006). Therefore, a weight is assigned to each rule to describe this (Yang et al., 2006).

On the basis of these concepts, according to Yang et al. (2006), the simple rule displayed in (5.7) can be extended to the following:

$$R_k: if\ A_1^k \wedge A_2^k \wedge \ldots \wedge A_{T_k}^k, then\ \{(D_1, \beta_{1k}), (D_2, \beta_{2k}), \ldots, (D_N, \beta_{Nk})\},$$

$$\sum_{i=1}^{N} \beta_{ik} \leq 1,\ with\ a\ rule\ weight\ \theta_k\ and\ attribute\ weight\ \delta_{k1}, \delta_{k2}, \ldots, \delta_{kT_k},$$

$$k \in \{1, 2, \ldots, L\} \tag{5.8}$$

As stated by Yang et al. (2006), in (5.8), $A_i^k (i = 1, 2, \ldots, T_k)$ indicates the referential value of the $i^{th}$ antecedent attribute in the $k^{th}$ rule and $T_k$ is the number of antecedent attributes used in the $k^{th}$ rule. $\beta_{ik} (i \in \{1, 2, \ldots, T_k\})$ is the degree of belief that $D_i$ is the consequence, given that, in the $k^{th}$ rule, the input satisfies the packet antecedents $A^k = \{A_1^k, A_2^k, \ldots, A_{T_k}^k\}$. L is the number of rules in the rule base. $\theta_k$ is the relative weight of the $k^{th}$ rule and $\delta_{k1}, \delta_{k2}, \ldots, \delta_{kT_k}$ represent the relative weights of the antecedent attributes used in the $k^{th}$ rule (Yang et al., 2006). If $\sum_{i=1}^{N} \beta_{ik} = 1$, the $k^{th}$ rule is said to be complete; otherwise, it is said to be incomplete (Yang et al., 2006). In addition, according to Yang et al. (2006), $\sum_{i=1}^{N} \beta_{ik} = 0$ indicates total ignorance about the output, given the input in the $k^{th}$ rule. If a rule is expressed in the form displayed in (5.2), it is referred to as a belief rule. If a rule base is composed of belief rules, the rule base is referred to as belief rule base.

In this research, we use the methods introduced in Section 3.5 to acquire evidence from data and combine evidence from different input variables to generate the probability of each class for each combination of pieces of evidence from different input variables. Here, the probability of each class and each combination of pieces of evidence from different input variables correspond to the degree of belief in each value of the consequence of a rule and the 'if' part of a belief rule. In this way, we can generate a belief rule for each combination of pieces of evidence from different input variables. Thus, we can generate a belief rule base for inference.

In a BRB, the consequence of a rule takes the form of a distribution (Yang et al., 2006). Thus, any difference in antecedent attributes can be clearly reflected in the consequence (Yang et al., 2006). As a comparison, different antecedents may lead

to the same consequence in a conventional rule base (Yang et al., 2006). Due to the introduction of attribute weights and rule weights, expert knowledge can be modelled more precisely and the BRB model can be brought closer to reality (Yang et al., 2006). Overall, a belief rule as previously defined can represent a functional mapping between antecedent inputs and outputs, possibly with uncertainties (Chen et al., 2013). A belief rule can provide a more informative and realistic scheme than a conventional IF-THEN rule (Chen et al., 2013). Once a belief rule base is set up, the knowledge embedded in all the belief rules can be applied to infer something for a specific input vector (Chen et al., 2013).

## 5.4 Prediction and Machine Learning

According to Yang and Xu (2017), we need to assign values to the parameters of Equation (3.14), i.e., $r_{\theta,i,l}, w_{\theta,i,l}, r_{\theta,j,m}, w_{\theta,j,m}, and\ \gamma_{A,B,i,j}$ in order to make an inference. We can use the adapted genetic algorithm introduced in Section 3.6 to train the values of these parameters based on input-output data sets. Then, we can use the model with the trained parameters to predict system outputs from given system inputs.

In order to train the parameters of the model, we need to establish a general least squares optimization model, which is shown in Equation (5.9).

$$\min \delta = \frac{1}{2S} \sum_{s=1}^{S} \sum_{\theta \subseteq \Theta} \left( p(\theta) - \hat{p}^{(s)}(\theta) \right)^2 \tag{5.9}$$

$$s.t.\ r_{\theta,i,l}, w_{\theta,i,l}, \gamma_{A,B,i,j} \in \Omega$$

In Equation (5.9), $\hat{p}^{(s)}(\theta)$ is the probability that assertion θ is true for the $s^{th}$ observation. The objective of the optimization model is to minimise the difference

between the observed outputs of a system and the corresponding predicted outputs of a model, which is measured by mean squared error (MSE). Ω is the feasible space of parameters, which refers to the constraints on the parameters of the optimization model (Yang and Xu, 2017).

Based on the minimum mean squared error, we can use the single-level adapted genetic algorithm presented in Section 3.6 to train the parameters of the optimization model.

Based on the illustration of Chapter 3 and Sections 5.2 through 5.4, we can use the flow charts in Figure 5.1 to visualise all the steps of the approach of rule-based inferential modelling and prediction, those of the adapted single-level genetic algorithm, and the connections between the approach and the algorithm.

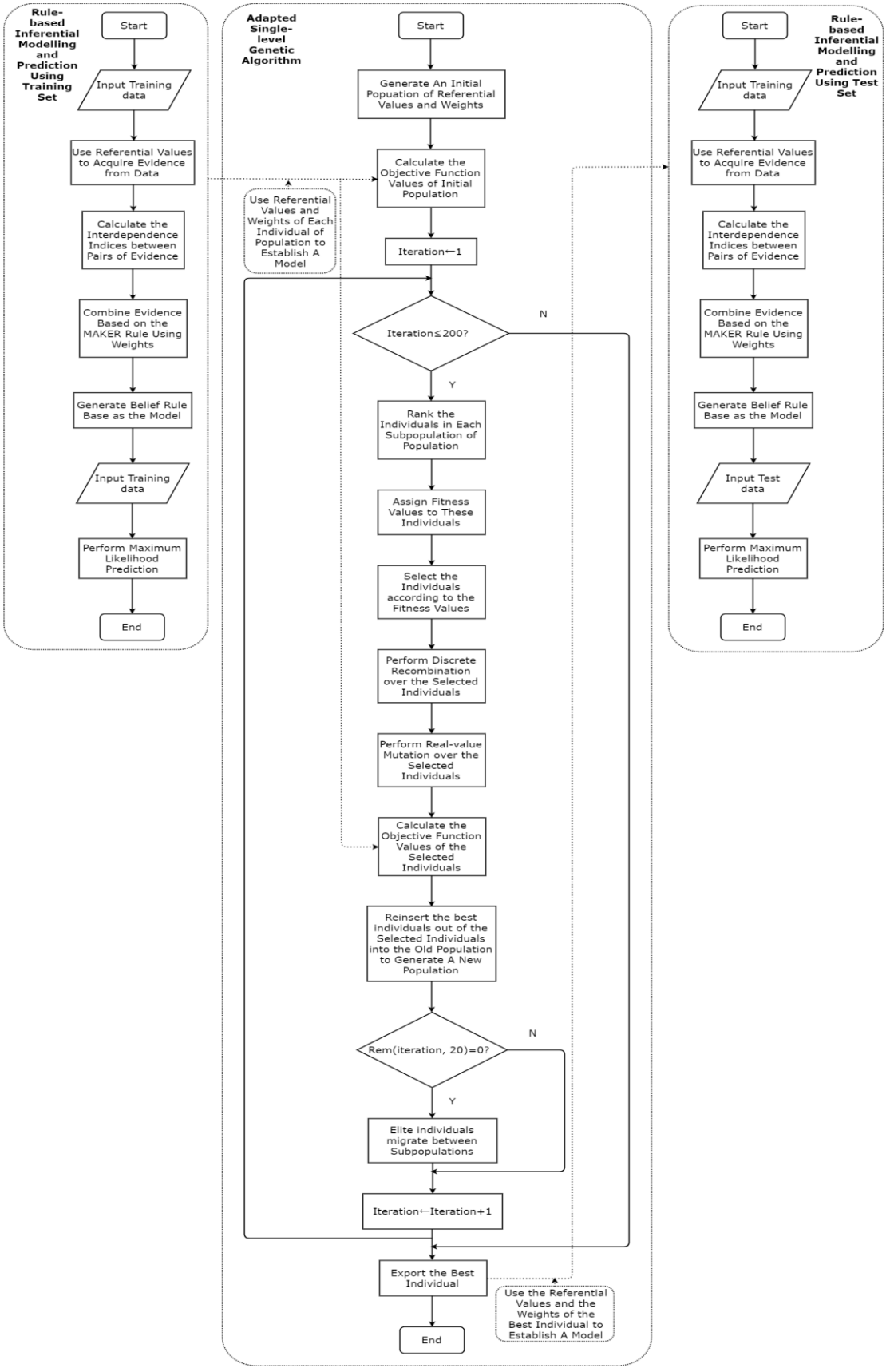



**Figure 5.1 The Flow Diagrams of the Approach of Rule-based Inferential Modelling and Prediction and the Adapted Single-level Genetic Algorithm**

## 5.5 Comparative Analysis of Modelling and Prediction Approach

In this section, the modelling kernel and the inference mechanisms of the rule-based inferential modelling and prediction approach based on the MAKER framework is presented analytically and graphically, first of all. Then this approach is compared to alternative modelling and prediction approaches, to highlight the advantages of the former.

Based on the method introduced in Section 3.5.1, the multi-model decomposition of input space based on the referential-value-based discretization is applied in the rule-based inferential modelling and prediction approach. This method of decomposition discretizes continuous data from an input space to generate evidence for inference. It helps decision makers acquire evidence directly from data using sample statistics and it requires few assumptions about the specific statistical distributions of the input data and the relationships between the input variables and the output variable.

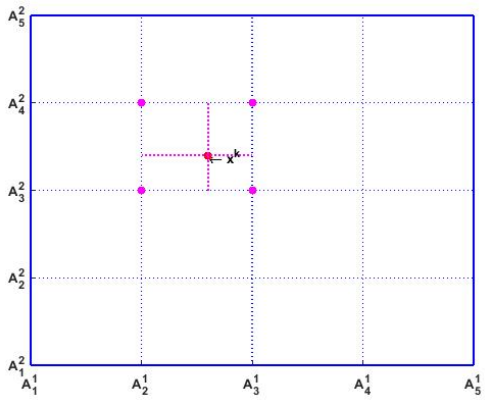



**Figure 5.2 Decomposition of 2-D Input Space**

To construct an inference system in the rule-based inferential modelling and prediction approach, as stated in Section 3.5.1, we need to define a number of referential values, indicated by $A_j^i$ $(\mathrm{i} = 1, \ldots, \mathrm{M}; \mathrm{j} = 1, \ldots, J_i)$, for each of the input variables. With the referential values, the input space U can be decomposed into a number of local regions, which are represented by the hyperspace $\left[A_1^1, A_{J_1}^1\right] \times \cdots \times \left[A_1^M, A_{J_M}^M\right]$. For example, a two-dimensional input space $x_1 \times x_2$ can be decomposed

into ($J_1$-1) × ($J_2$-1) = 4 * 4 = 16 rectangular local regions as displayed in Figure 5.2, while a three-dimensional input space $x_1$ × $x_2$ × $x_3$ can be decomposed into ($J_1$-1) × ($J_2$-1) × ($J_3$-1) = 2 * 2 * 2 = 8 cubic local regions as presented in Figure 5.3. Each data point of the input vector i.e., $x_n = \{x_{n,i_1}, \ldots, x_{n,i_M}\}$ for the input variables can then be located within a local region determined by the intersections of the referential values.

Through the calculations displayed in Section 3.5.1, each intersection of referential values will be given a degree of belief or probability indicating how likely it is that the input vector composed of these referential values would be present under a certain class of output variable. This is the process used to generate a belief rule base for inference, as mentioned in Section 5.3. Meanwhile, the similarity degrees mentioned in Equation (3.4) can be used to measure the proximity of the data point of an input vector to the intersections of the referential values. Thus, based on the referential values and similarity degrees, we can provide a complete description of the relative location of the data point of an input vector $x_n$ in the input space U. Naturally, the granularity and interpretability of local regions are generally decided by the number of referential values. It is evident that the greater the number of referential values, the more accurate the location of the data point of an input vector $x_n$ will be. In most cases, inputs which lead to high volatility of outputs require more referential values than others.

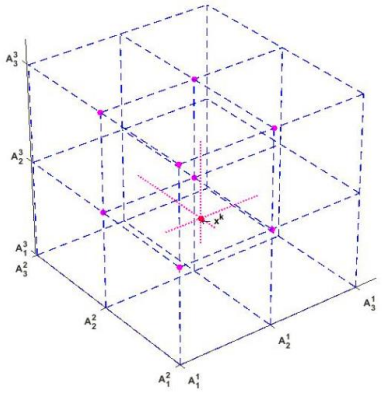


**Figure 5.3 Decomposition of 3-D Input Space**

Under the above-described structure, the inference system of this referential-value-based inferential modelling and prediction approach is essentially a multi-model approximator combining decomposed sub-models represented by local regions to

describe the general pattern of a numerical system. For each sub-model, i.e., local region, we can formulate the relationship between the input variables and the output variable using the inference system demonstrated in Section 3.5.1.

On the basis of the intersections of the referential values and their corresponding probabilities, i.e., the belief rule base, and the degrees of similarity between the data point of the input vector and the relevant intersections of the referential values, we can activate relevant belief rules from the belief rule base and then use Equations (3.14) to (3.19) to combine these belief rules to generate a combined probability for an input vector pointing to each class of output variable. As previously stated, we can use the adapted genetic algorithm presented in Section 3.4 to train the parameters of the optimization model to maximize the likelihood of the true state. Thus, from the above analysis, we can see that the rule-based inferential modelling and prediction approach is uniquely interpretable, making it an objective, rigorous, and reliable inference method for a data-driven system.

The above analysis shows that the application of the rule-based inferential modelling and prediction approach can acquire evidence directly from data using statistical analysis. Combining multiple pieces of evidence from different input variables of data generates a belief rule base for inference. With the belief rule base, the relationship between input and output can be formulated by the unified inference scheme described in Section 3.5. For any given inputs, an inference can be made about the corresponding output using the belief rule base and maximum likelihood prediction. With the algorithm of machine learning, the parameters of the inference model can be optimised to make the predicted probability of the output become as close as possible to the probability of the true state of the output. The inference process, based on the rule-based inferential modelling and prediction approach, is totally transparent and interpretable. Such a unique interpretability makes the rule-based inference modelling and prediction approach an effective,

rigorous, and reliable inference method for a numerical system. This kind of unique interpretability is illustrated by the example shown in Sections 6.3.4 and 5.6.6.

Interpretability is very important in science. According to Zhou and Chen (2018), it is defined as 'explaining or presenting in understandable terms'. One of the hallmarks of good science is understanding and trusting models and their results (Hall and Gill, 2018). The models and their results are essentially the representation of knowledge. Scientists want to know what kind of knowledge, learned by models from data, leads to the output the models generate. This may lead to possible associations between inputs and output, which can guide further research.

Researchers, engineers, and medical experts generally have a need to understand and trust models and their results (Hall and Gill, 2018). It is particularly important to understand interpretability in disease diagnosis using machine learning methods. With machine learning methods, medical experts can establish a model to predict patients' risk for diseases. In addition to the predicted results, the medical experts may also want to know the possible relationship between the patients' features and their diagnoses. This would allow medical experts to judge whether the predicted results of the models are meaningful, based on medical knowledge and experience, and allow the use of effective models to improve the accuracy and efficiency of disease diagnosis.

If a model is uninterpretable, the application of the model in different domains may be restricted by insufficient information provided by the model. For example, if neural networks are used to classify two images which are almost the same, yet slightly different, completely different classifications can result for these two images. However, it is very difficult to tell what difference between the two images leads to the two different classifications. Unlike uninterpretable models, we can use interpretable models to track and locate the difference in inputs which lead to different classification results. For example, the difference in the parameters of

inputs of linear regression models, which lead to different classification results, can be found easily.

Generally speaking, there is a compromise between accuracy and interpretability of models (Hall and Gill, 2018). Higher accuracy always comes at the expense of interpretability (Hall and Gill, 2018). Some classifiers (e.g. artificial neural networks (ANN), ensembles, and random forests) may have a very high accuracy for the classification of data sets, but what makes them accurate is what makes their predictions hard to understand (Hall and Gill, 2018). On the one hand, the complex and inscrutable inner-workings enable these models to have a tremendous capability for classification but, on the other hand, this also renders these models difficult to understand.

Conversely, compared to the classifiers which are hard to understand, other classifiers (e.g. decision tree, logistic regression, and naïve Bayes) generally have a lower accuracy for the classification of data sets, but they are easier to understand.

According to Rokach and Maimon (2015), decision tree classifiers divide the input space into a number of mutually exclusive regions to represent one concept. All the data points of inputs in the same region are assigned a same output, which is simplistic. Similar to the decision tree, MAKER-based models divide the input space into mutually exclusive regions as well. However, as already shown in this section, MAKER-based models use the probabilities of the intersections of the referential values (splits) and the similarities between the data points of inputs and the intersections of the referential values to generate the outputs of the given data point of inputs, which is closer to reality.

Another disadvantage to the decision tree is over-sensitivity (Rokach and Maimon, 2015) to the training set, which makes decision tree classifiers very unstable. A

small change in one split which is close to the root can change the entire subtree (Rokach and Maimon, 2015). Compared to decision tree classifiers, MAKER-based classifiers are generally more stable. A small change in the referential values (splits) will not significantly change the outputs of MAKER-based classifiers for the inputs, as each input will activate a number of rules and these belief rules and relevant similarities can be used to generate the output for a given input.

As stated by Moreira, Carvalho and Horváth (2018), logistic regression classifiers are restricted to linearly separable binary classification tasks. MAKER-based classifiers are not restricted to linearly separable binary classification tasks. MAKER-based classifiers can be applied to nonlinear separable binary classification tasks and multiple classification tasks. In addition, logistic regression classifiers are sensitive to correlative input variables and outliers (Moreira, Carvalho and Horváth, 2018). As introduced in Section 3.5, MAKER-based classifiers take into consideration the interdependence between input variables, as MAKER-based classifiers use an interdependence index to measure the interdependence between input variables. In addition, MAKER-based classifiers use sample statistics to generate evidence for inference. Hence, MAKER-based classifiers are less sensitive to the outliers than logistic regression classifiers.

Naïve Bayes classifiers assume that all input variables are independent from each other (Nicolas, 2015) and naïve Bayes classifiers are generally dependent on the prior distribution. According to Yang and Xu (2017), MAKER-based classifiers do not depend on prior distribution but admits unknown prior by default. The prior distribution is treated as an independent piece of evidence which is added to the evidence set of the evidence space model, as mentioned in Section 5.2 (Yang and Xu, 2017).

## 5.6 A Case Study of Classification of the Iris Data Set

### 5.6.1 Correlation between Classification and Functions Approximations

There is a correlation between classification and continuous functions approximations. Indeed, classification can be regarded as a simplified version of continuous functions approximation. This is due to the fact that the observed outputs of data sets in continuous functions approximation are continuous values and the observed outputs of the data sets in classification are generally categorical. If the categorical observed outputs of the data sets in the classification are represented by discretized values, the observed outputs of data sets in classification can just be considered as a number of values of the observed outputs of data sets in continuous functions approximation. Equation (5.10) exhibits a continuous univariate function that can be used to describe the relationship between the inputs and the outputs of a data set for continuous function approximation and Equation (5.11) displays a piecewise function that can be used to describe the relationship between the inputs and the outputs of a data set that has only 1 input variable for classification.

$$f(x) = x^{\frac{1}{2}}, \quad 0 \leq x \leq 9 \tag{5.10}$$

$$g(x) = \begin{cases} 1, & 0 \leq x < 3 \\ 2, & 3 \leq x < 8 \\ 3, & 8 \leq x \leq 9 \end{cases} \tag{5.11}$$

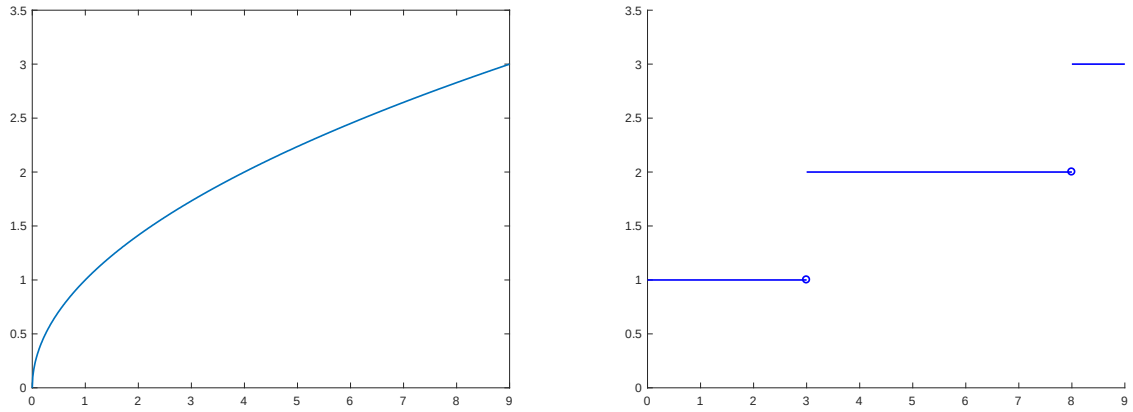



**Figure 5.4 A continuous function about continuous functions approximation and a piecewise function about classification**

Figure 5.4 visualizes the functions exhibited in both Equation (5.10) and Equation (5.11), which provides a more intuitive display of the relationship between continuous functions approximation and classification. It is noteworthy that in the approximations for continuous functions presented in Chapter 4, the predicted outputs of the models based on the system of MAKER framework are actually the probabilities for the referential values of observed output values of functions of the data sets. Then, the sums of the products generated by the referential values of the observed output values of the functions and their corresponding probabilities yielded by the models can be used as the predicted values for the observed output values of the functions in the approximations for continuous functions. In the classification of classical data sets presented in this chapter, the predicted outputs of the models based on the system of the MAKER framework are just the probabilities for different classes of the observed outputs of the data sets for classification. Then the class with the highest probability would be taken as the predicted class of the model for each observation of a data set in classification.

### 5.6.2 Data Sets

The functions approximation presented in Chapter 4 has shown the remarkable approximation capability of the models based on the system of the MAKER framework, which provides a solid foundation to apply the rule-based inferential modelling and prediction approach based on the system of the MAKER framework to the classification of data sets. In this section, as mentioned previously, we would take the classification of classical data sets as a case study to compare the rule-based inferential modelling and prediction approach based on the system of the MAKER framework and other alternative modelling and prediction approaches. The classical data sets as stated above include the Banana data set, the Haberman's survival data set, and the Iris data set, which were briefly introduced in Section 3.3. Each of these data sets had already been divided into five folds using

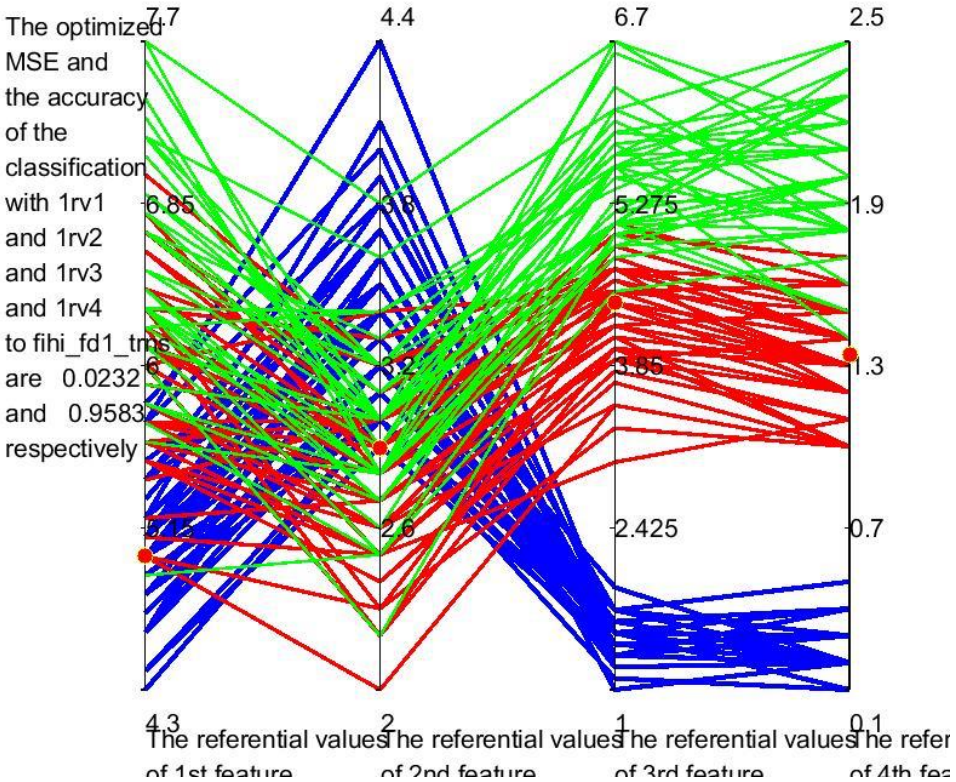



**Figure 5.5 Parallel coordinates plot of distribution of observations of different classes across four input variables of the training set of the first fold of the Iris data set and the trained referential values of the input variables of the Iris data set**

distribution optimally balanced stratified cross-validation (Alcalá-Fdez et al., 2011) before they were downloaded from the KEEL-dataset repository, so that the class distribution of the whole data set is reflected in separate folds (Aggarwal, 2015).

In order to illustrate how the rule-based inferential modelling and prediction approach can be used to build a MAKER-based classifier for the classification of a data set, a numerical study using the iris data set is presented in the remainder of this section. As previously mentioned, the Iris data set has four input variables (i.e., sepal length, sepal width, petal length, and petal width) and a categorical output variable (i.e., species with three classes, i.e., *Iris setosa*, *Iris versicolor*, and *Iris virginica*). As stated previously, the Iris data set has been divided into five folds for cross-validated classification and the class distribution of the whole data set is reflected in these separate folds, which means the observations of all these separate folds have similar class distributions. Thus, we can just take the training set of the first fold of the Iris data set as an example to demonstrate how to use the rule-based inferential modelling and prediction approach to establish a MAKER-based classifier for the classification of the Iris data set.

Figure 5.5 exhibits the parallel coordinates plot about the general distribution of the observations of different classes across four input variables of the training set of the first fold of the sepsis data set and the locations of the trained referential values of each input variable of the data set. As is indicated in Figure 5.5, the blue solid lines, the red solid lines, and the green solid lines represent the observations of *Iris setosa*, *Iris versicolor*, and *Iris virginica* respectively. The four vertical axes in Figure 5.5 from left to right represent the input variables: sepal length, sepal width, petal length, and petal width, respectively. The red nodes with yellow edges on the axes of Figure 5.5 denote the trained referential values of the input variables of the Iris data set.

From Figure 5.5, it can be observed that the distributions of observations of different classes in the different input variables of the Iris data set can have different patterns. Specifically, in the input variable of sepal length, the observed input values of Iris virginica represented by the green solid lines are generally the largest, and the observed input values of Iris versicolor represented by the red solid lines are generally smaller than those of Iris virginica, and the observed input values of Iris setosa represented by blue solid lines are generally the smallest. In the input variable of sepal width, the observed input values of Iris virginica and those of Iris versicolor are generally smaller than those of Iris setosa, and the observed input values of Iris versicolor are generally smaller than those of Iris virginica. The distribution of observations of different classes in the input variable of petal length is similar to that in the input variable of petal width. In the input variables: petal length and petal width, the observed input values of Iris virginica are generally the largest, and the observed input values of Iris versicolor are generally smaller than those of Iris virginica, and the observed input values of Iris setosa are generally the smallest.

### 5.6.3 The Optimised Referential Values of the Model

As introduced in Chapter 3, we can use the adapted genetic algorithm to optimize the parameters, i.e., the referential values and the weights of the models based on the system of the MAKER framework for functions approximation or classification. In the adapted genetic algorithm for the optimization of the MAKER-based models of sepsis diagnosis, the initial population of individuals (chromosomes) includes 10 subpopulations, each of which comprises 20 individuals (chromosomes). Each individual of a population contains both the referential values of the observed values of the input variables of a training set of the Iris data set and the weights (reliabilities) of these referential values under different classes of the output variable of the data set. After the initial population has been generated, the

objective value, i.e., MSE is calculated for each individual (solution) of a population. Then, on the basis of a population of the individuals as stated above, a group of genetic algorithm operations, e.g., selection, recombination, mutation, reinsertion, and migration, is performed for 200 iterations to obtain an optimized solution for referential values and weights.

The target of the optimization of a MAKER-based model of sepsis diagnosis is to maximize the predicted outputs, i.e., predicted probabilities, of the model for the true observed outputs of a training set of the Iris data set to minimize the MSE that is used to measure the difference between the observed outputs and the predicted outputs. Thus, it can reasonably be inferred that, through optimization, the trained referential values of each input variable will generally be located around critical observed values that divide the observed values of this input variable into several parts, in each of which the observations of a given class are in the majority.

This is supported by Figures 5.6 and 5.7, which present the distribution of trained referential values obtained from the optimization of the MAKER-based models for the classification of the Iris data set. The MAKER-based models of Figure 5.6 all have one trained referential value for each input variable of the training set, and those of Figure 5.7 all have two trained referential values for each input variable of the training set. It is worth noting that the 'trained referential values' refer to the trained referential values of observed values of an input variable of a data set between the minimum and maximum of those observed values. From Figures 5.6 and 5.7, we can clearly see that the trained referential values for the input variables obtained from the optimization of the MAKER-based models are generally located around the separation point between the clusters of species (*Iris setosa*, *Iris versicolor*, and *Iris virginica*). For example, in the first subfigure of Figure 5.5, the trained referential value for the input variable: petal width (the rightmost vertical axis of the subfigure) is located around the separation point between the cluster of *Iris versicolor* and the cluster of *Iris virginica*.

As already mentioned in this section, the training set of the first fold of the Iris data set (henceforth the 'training set') is taken as an example to illustrate how a MAKER-based classifier is established through the rule-based inferential modelling and prediction approach for the classification of the Iris data set. Hence, we just use the optimized solution of the MAKER-based classifier for the training set to construct a classifier for the purpose of illustration. The optimized solution includes optimized referential values and optimized weights obtained from the optimization of the MAKER-based model, which has one optimized referential value for each of the input variables of the training set. The 'optimized referential values' refer to the optimized referential values of the observed values of an input variable of a data set, which lie between the minimum and maximum of the observed values of this input variable.

In the following part of this section, we will show how to apply the rule-based inferential modelling and prediction approach to build the MAKER-based classifier for the training set from four aspects: evidence acquisition from data, analysis of evidence independence, belief rule-base inference, and maximum likelihood prediction and machine learning.

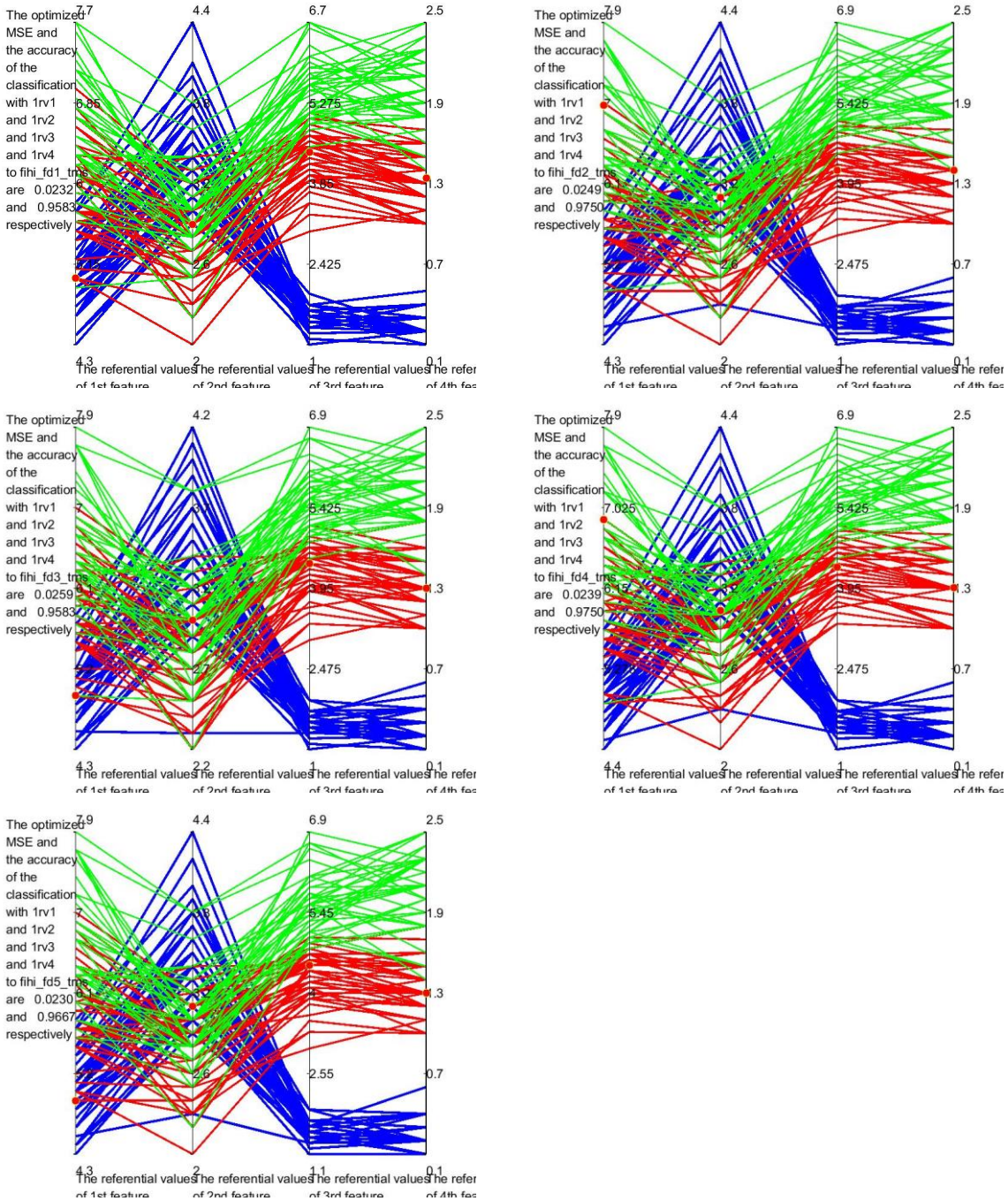



**Figure 5.6 The distribution of trained referential values obtained from the optimization of MAKER-based models for the classification of the Iris data set which all have one Trained referential value for each of the input variables of the training set**

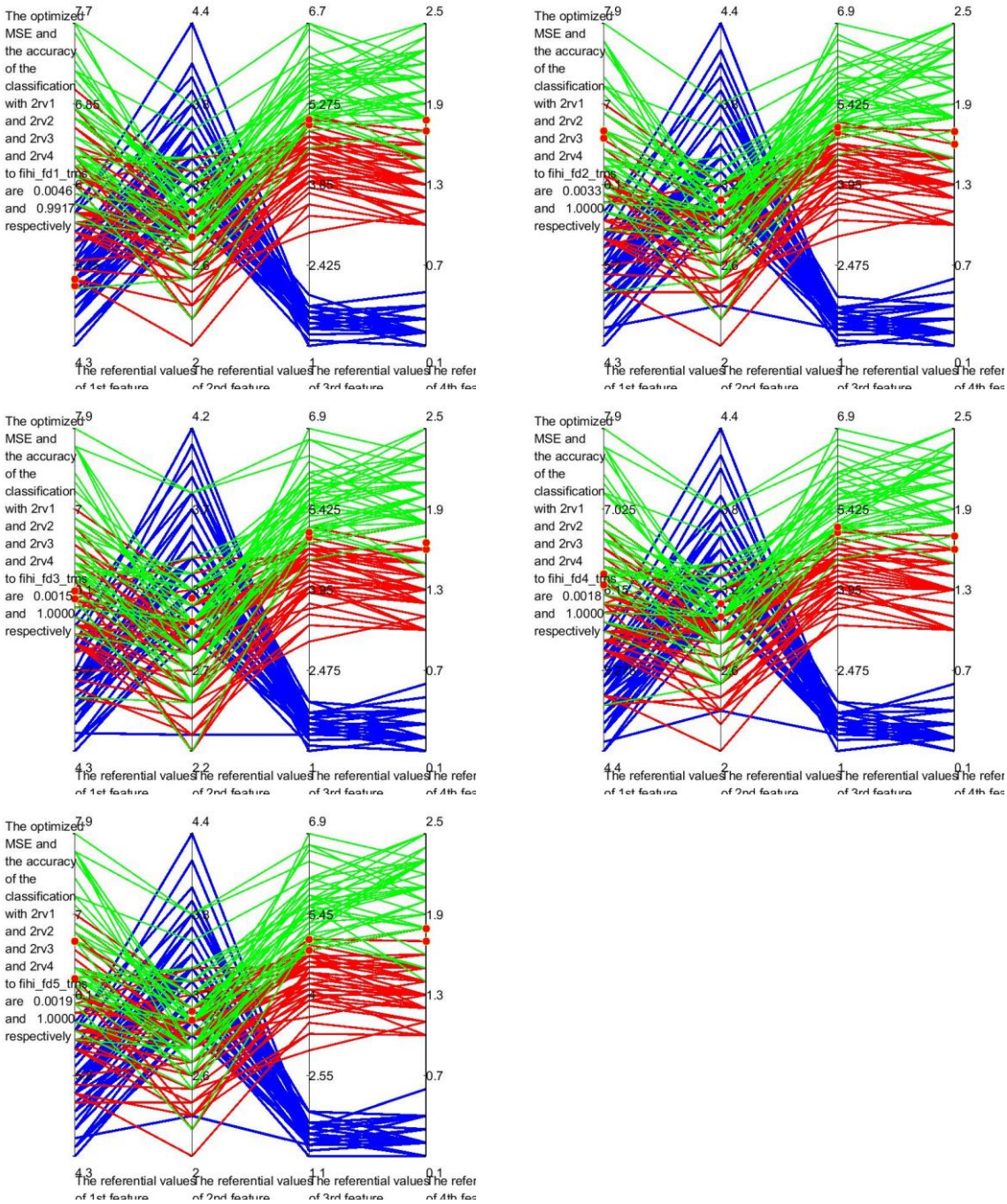



**Figure 5.7 The distribution of trained referential values obtained from the optimization of MAKER-based models for the classification of the Iris data set which all have two trained referential value for each of the input variables of the training set**

### 5.6.4 Evidence Acquisition from Data

As introduced in Section 3.5, the first step in establishing a MAKER-based classifier is to acquire evidence from data. In order to acquire evidence from a data set, we need to decide on referential values for each of the input variables of the data set. Referential values, as adjustable parameters, can initially be determined by expertise or random rule without prior knowledge and can subsequently be trained using an input-output data set under a certain optimization objective (Xu et al., 2017).

As previously stated, in this section, for the purpose of illustration, we use the optimized solution of the MAKER-based classifier for the training set. This includes optimized referential values and optimized weights obtained from the optimization of the MAKER-based classifier, which has one optimized referential value for each of the input variables of the training set. Thus, we just use the optimized referential values obtained from the above-mentioned optimization of the MAKER-based classifier to acquire evidence from the training set.

**Table 5.1 The referential values obtained from the optimization of the MAKER-based classifier for the training set of the first fold of the Iris data set**

| Input variables | sepal length | sepal width | petal length | petal width |
|---|---|---|---|---|
| Boundary referential values (minima) | 4.3000 | 2 | 1 | 0.1000 |
| Optimized referential values | 4.9991 | 2.8969 | 4.4044 | 1.3389 |
| Boundary referential values (maxima) | 7.7000 | 4.4000 | 6.7000 | 2.5000 |

Table 5.1 displays the referential values including the boundary referential values and the optimized referential values used to acquire evidence from data. The boundary referential values are the minima and maxima of the observed values of the input variables of the data set. On the basis of the optimized referential values displayed in Table 5.1 and the boundary referential values as stated above, we can use Equation (3.4) from Section 3.5.1 to transform each observed value of each input variable of the training set into the belief distributions of the two adjacent referential values between which this observed value is located. After all the observed values of the input variables are transformed into belief distributions of referential values, we will use Equation (3.5) to aggregate similarity degrees of belief distributions according to the referential values under different classes of the output variable of the training set. In this way, we will generate the frequencies of the referential values under different classes of the output variable displayed in the form of Table 3.3. Table 5.2 shows the frequencies of the referential values: 4.3, 4.9991, and 7.7 of the input variable: sepal length.

**Table 5.2 The frequencies of the referential values of the input variable of sepal length of the training set of the first fold of the Iris data set under different species**

| class/referential value | 4.3000 | 4.9991 | 7.7000 |
|---|---|---|---|
| Iris setosa | 7.5588 | 30.3599 | 2.0812 |
| Iris versicolor | 0 | 26.2507 | 13.7493 |
| Iris virginica | 0.1417 | 17.0017 | 22.8566 |

Next, using Equation (3.6), we calculate the likelihood of a referential value of an input variable being true given that a class of the output variable of the training set is true, for all the referential values of the input variables under different classes of the output variable, displayed in the form of Table 3.4. Table 5.3 presents the likelihoods of the referential values: 4.3, 4.9991, and 7.7 of the input variable of sepal length being different species. With these likelihoods, the probability of a referential value of an input variable pointing to a class of the output variable of

the training set can be obtained from Equation (3.7) for all the referential values of the input variables under different classes of output variable, shown in the form of Table 3.5.

**Table 5.3 The likelihoods of the referential values of the input variable of sepal length of the training set of the first fold of the Iris data set being different species**

| class/referential value | 4.3000 | 4.9991 | 7.7000 |
|---|---|---|---|
| Iris setosa | 0.1890 | 0.7590 | 0.0520 |
| Iris versicolor | 0 | 0.6563 | 0.3437 |
| Iris virginica | 0.0035 | 0.4250 | 0.5714 |

Tables 5.4 and 5.5 exhibit the probabilities of the referential values of the observed values of the input variables of sepal length and sepal width, respectively, pointing to different classes of the output variable of the training set. It is worth noting that the referential values displayed in Tables 5.4 and 5.5 include not only the previously defined boundary referential values, i.e., 4.3000, 7.7000, 2, and 4.4000, but also the previously defined optimized referential values, i.e., 4.9991 and 2.8969.

**Table 5.4 The probabilities with which the referential values of the observed values of the input variable of sepal length point to different classes of the output variable of the training set of the first fold of the Iris data set**

| class/referential value | 4.3000 | 4.9991 | 7.7000 |
|---|---|---|---|
| Iris setosa | 0.9816 | 0.4124 | 0.0538 |
| Iris versicolor | 0 | 0.3566 | 0.3554 |
| Iris virginica | 0.0184 | 0.2310 | 0.5908 |

**Table 5.5 The probabilities with which the referential values of the observed values of the input variable of sepal width point to different classes of the output variable of the training set of the first fold of the Iris data set**

| class/referential value | 2 | 2.8969 | 4.4000 |
|---|---|---|---|
| Iris setosa | 0 | 0.3019 | 0.7055 |
| Iris versicolor | 0.7011 | 0.3324 | 0.0847 |
| Iris virginica | 0.2989 | 0.3657 | 0.2098 |

From the probabilities of the referential values of the observed values of the input variables of the training set pointing to different classes of the output variable of the training set, partly shown in Tables 5.4 and 5.5, we can acquire a number of pieces of evidence. For example, as shown in Table 5.9, the probabilities: 0.9816 and 0.0184, under boundary referential values: 4.3000, indicate that if the sepal length is 4.3000 cm, the probability of this flower being Iris setosa is 0.9816, and the probability of this flower being Iris versicolor is 0, and the probability of this flower being Iris virginica is 0.0184. Thus, we can acquire a piece of evidence from the sepal length of 4.3000 cm, in that it points to the Iris setosa with a probability of 0.9816, and points to the Iris versicolor with a probability of 0, and points to the Iris virginica with a probability of 0.0184.

### 5.6.5 Analysis of Evidence Independence

As stated previously, there are four input variables: sepal length, sepal width, petal length, and petal width in the Iris data set, and the output variable, i.e., *species* of the Iris data set contains three classes: *Iris setosa*, *Iris versicolor*, and *Iris virginica*. Without doubt, the predictive power of a single piece of evidence is limited. In order to achieve greater predictive power, it is necessary to combine multiple pieces of evidence to make a prediction for a patient. In the original evidential reasoning (ER) rule, any two pieces of evidence to be combined are assumed to be independent from each other, which is simplistic. Under the MAKER framework, the interdependence between two pieces of evidence is taken into consideration through the introduction of an interdependence index 'α' which is defined in Equation (3.13). To generate the interdependence index between each pair of evidential elements, we need to estimate the joint probabilities for these two pieces of evidence according to Equation (3.12) in advance. Table 5.11 displays the joint probabilities for all the combinations of the referential values of pieces of evidence

from the input variables: sepal length and sepal width, each of which points to different classes of output variable of the training set.

In Table 5.6, the first and second numbers of each combination of referential values represent the referential value of a piece of evidence from the input variable of CRP of the training set and that from the input variable of IL6 respectively. On the basis of the probabilities displayed in Table 5.4, Table 5.5, and Table 5.6, we can use Equation (3.13) to generate the interdependence indices between a piece of evidence from the input variable of CRP and a piece from the input variable of IL6 which are exhibited in Table 5.7. From Table 5.7, it is evident that the sepal length and sepal width are generally moderately independent from each other, as the majority of the independence indices displayed in Table 5.7 are between 1 and 7, although there are several exceptional independence indices, e.g., 0 and 181.8097. For example, the interdependence between the sepal length: 4.9991 cm and the sepal width: 4.4 cm is moderate, as the interdependence index between the sepal length: 4.9991 cm and the sepal width: 4.4 cm under *Iris setosa* is 2.8891, and that under *Iris versicolor* is 2.1496, and that under *Iris virginica* is 1.9491. The sepal length: 4.3 cm and the sepal width: 2 cm are highly independent from each other under *Iris virginica*, as the corresponding interdependence index is 181.8097.

**Table 5.6 The joint probabilities with which different combinations of the referential values of pieces of evidence from the input variables: sepal length and sepal width point to different classes of the output variable of the training set of the first fold of the sepsis data set**

| class/combination of referential values | {4.3, 2} | {4.3, 2.8969} | {4.3, 4.4} | {4.9991, 2} | {4.9991, 2.8969} | {4.9991, 4.4} | {7.7, 2} | {7.7, 2.8969} | {7.7, 4.4} |
|---|---|---|---|---|---|---|---|---|---|
| Iris setosa | 0 | 0.9876 | 1 | 0 | 0.3777 | 0.8406 | 0 | 0.0284 | 0.2558 |
| Iris versicolor | 0 | 0 | 0 | 0.7462 | 0.3602 | 0.0649 | 0.6003 | 0.3556 | 0.1615 |
| Iris virginica | 1 | 0.0124 | 0 | 0.2538 | 0.2621 | 0.0945 | 0.3997 | 0.6160 | 0.5826 |

**Table 5.7 The interdependence indices between a piece of evidence from the input variable of sepal length of the training set of the first fold of the Iris data set and that from the input variable of sepal width of the training set**

| referential value of sepal length | 4.3 | 4.3 | 4.3 | 4.9991 | 4.9991 | 4.9991 | 7.7 | 7.7 | 7.7 |
|---|---|---|---|---|---|---|---|---|---|
| class/referential value of sepal length | 2 | 2.8969 | 4.4 | 2 | 2.8969 | 4.4 | 2 | 2.8969 | 4.4 |
| Iris setosa | 0 | 3.3328 | 1.4441 | 0 | 3.0340 | 2.8891 | 0 | 1.7463 | 6.7411 |
| Iris versicolor | 0 | 0 | 0 | 2.9846 | 3.0388 | 2.1496 | 2.4091 | 3.0103 | 5.3646 |
| Iris virginica | 181.8097 | 1.8426 | 0 | 3.6764 | 3.1024 | 1.9491 | 2.2636 | 2.8510 | 4.7003 |

### 5.6.6 Belief Rule-base Inference

As we have acquired a number of pieces of evidence from the input variables of the training set, and we have analysed the interdependence between two pieces of evidence, we are now in a position to construct a belief rule base for inferring the likelihood of a flower being Iris setosa, Iris versicolor, or Iris virginica on the basis of their sepal length, sepal width, petal length, and petal width. According to the belief rule described in (5.8) of Section 5.3, the antecedent of the belief rule, which is expressed in the form of '$if\, A_1^k \wedge A_2^k \wedge \ldots \wedge A_{T_k}^k$' in (5.8), should be expressed in this case study of the Iris data set as 'if the value of each feature of a flower is just equal to a referential value of the observed values of this feature'. The corresponding consequent of the belief rule, which is expressed in the form of '$then\, \{(D_1, \beta_{1k}), (D_2, \beta_{2k}), \ldots, (D_N, \beta_{Nk})\}$' in (5.8), should then be expressed in this case study of the Iris data set as 'the probability of the flower being each of the *species*: *Iris setosa*, *Iris versicolor*, or *Iris virginica* is equal to a value.

To obtain the probability of a flower being each of the *species*: *Iris setosa*, *Iris versicolor*, and *Iris virginica* in the consequent of a belief rule of this case study, we need to combine four pieces of evidence from the different features using the MAKER rule as described in Section 3.5.3. As stated previously, the optimized solution used in this case study, including optimized referential values and optimized weights, is obtained from the optimization of the MAKER-based classifier for the training set, which has one optimized referential value for each of the features. We use this optimized solution to construct a MAKER-based classifier on the training set for the purpose of illustration. Thus, there are a total of three referential values, namely the boundary referential values as stated previously and

an optimized referential value, for each of the four input variables of the training set in this case study.

Further, each piece of evidence contains a referential value from different input variables of the training set. There are three referential values in each of the four input variables of the training set for this case study. Hence, there are altogether 81 combinations of four pieces of evidence that can be used to construct the belief rule base in this case study of the Iris data set. In other words, there are a total of 81 combinations of four pieces of evidence, i.e., 81 belief rules for the belief rule base in this case study. On the basis of the previously mentioned optimized weights obtained from the optimization, we use Equation (3.14) to combine four pieces of evidence from different input variables of the training set to generate the probabilities of the *species* (*Iris setosa*, *Iris versicolor*, and *Iris virginica*) for each of the 81 combinations of four pieces of evidence. For instance, through calculation, we can obtain a probability: 0.0066 of *Iris setosa*, and a probability: 0.0015 of *Iris versicolor*, and a probability: 0.9919 of *Iris virginica* for the following combination of pieces of evidence: {7.7, 4.4, 6.7, 2.5}.

Each observation of the training set will activate 16 belief rules out of the total 81 belief rules of the belief rule base in this case study. This is because each observed value of each observation of the training set will activate two adjacent referential values, each of which belongs to a piece of evidence, of observed values of an input variable of the training set between which this observed value is located, and there are four input variables: sepal length, sepal width, petal length, and petal width in the training set. Hence, each observation of the training set activates $2^4 = 16$ combinations of four pieces of evidence, i.e., belief rules of the belief rule base in this case study.

Here we take an observation: {5, 2.3, 3.3, 1} from the training set as an example to demonstrate how an observation activates 16 belief rules from this case study's belief rule base for inference. Based on Table 5.1, the referential values activated by this observation in terms of the input variables, are displayed in Table 5.8. Then, these activated referential values are used to generate the combinations of referential values for the 16 belief rules from this case study's belief rule base that are activated by the same observation. These combinations of activated referential values are presented in Table 5.9. According to the 16 belief rules activated by each observation of the training set, we can find the corresponding probability of each class of the output variable of the training set that follows from each of these 16 belief rules. This can be used to predict the probability of each class of each observation.

**Table 5.8 The referential values of the input variables of the training set of the first fold of the Iris data set activated by the observation: {5, 2.3, 3.3, 1}**

| Input variable | sepal length | sepal width | petal length | petal width |
|---|---|---|---|---|
| Activated referential values | $4.9991, 7.7$ | $2, 2.8969$ | $1, 4.4044$ | $0.1, 1.3389$ |

### 5.6.7 Maximum Likelihood Prediction and Machine Learning

As already mentioned, each observation of the training set will activate 16 belief rules from this case study's belief rule base that can be used to predict the probabilities of the classes of this observation. On the basis of these 16 belief rules, we need to calculate the similarity degree between each observed value of this observation and each of the referential values between which this observed value is located, using Equation (3.16) from Section 3.5.4. The similarity degree indicates how closely each observed value matches each of those referential values. For

example, 5 is an observed value of the observation: {5, 2.3, 3.3, 1}, and according to Table 5.11, the referential values of the input variable: sepal length activated by the observed value: 5 are 4.9991 and 7.7. On the basis of Equation (3.16), the similarity degree between 5 and 4.9991 is calculated as $\frac{7.7-5}{7.7-4.9991} \approx 0.9997$ and that between 5 and 7.7 is calculated as $1-\frac{7.7-5}{7.7-4.9991} \approx 0.0003$, which indicates that the observed value: 5 matches the referential value: 4.9991 to a high degree and referential value: 7.7 to a low degree. In this way, we can calculate the similarity degree between each observed value of the observation: {5, 2.3, 3.3, 1} and each of the referential values activated by this observed value. With these similarity degrees, we can use Equation (3.17) to calculate the joint similarity degree between the observation: {5, 2.3, 3.3, 1} and the combination of referential values of each belief rule activated by this observation. The joint similarity degree indicates the degree to which we should invoke the belief rules activated by an observation to predict the probability of each class of the output variable for this observation.

Having generated this joint similarity degree, we are now in a position to combine these belief rules activated by the observation: {5, 2.3, 3.3, 1} to predict the probability of each class of the output variable, i.e., the species for this observation. In order to combine these belief rules, we need their weights. Using Equation (3.19), we can generate their weights from the probability mass $m_{\theta,e(L)}$ ($\theta \subseteq \Theta$, $\theta \neq \emptyset$) and the probability $p_{\theta,e(L)}$ ($\theta \subseteq \Theta$, $\theta \neq \emptyset$) or just the probability mass $m_{P(\Theta),e(L)}$, which are shown in both Equations (3.18) and (3.19).

As we use the weights of four pieces of evidence to combine four pieces of evidence from different input variables of the training set to generate the probability mass $m_{\theta,e(L)}$ ($\theta \subseteq \Theta$, $\theta \neq \emptyset$), the probability $p_{\theta,e(L)}$ ($\theta \subseteq \Theta$, $\theta \neq \emptyset$), and the probability mass $m_{P(\Theta),e(L)}$ for each belief rule in this case study's belief rule base, and we use the

relevant probability masses and relevant probability to generate the weight for each belief rule activated by the observation: {5, 2.3, 3.3, 1}, we can conclude that the weights of the four pieces of evidence have an effect on the weights of each belief rule activated by this observation.

On the basis of the joint similarity degrees between the observation and activated belief rules, and the weight of each activated belief rule, we can generate the updated weight of each activated belief rule for the observation, which considers the degree to which we should invoke these activated belief rules in predicting the probability of each class of the observation. As the weights of the four pieces of evidence from the different input variables have an effect on the weight of each belief rule activated by this observation, and since we use the joint similarity degrees between the observation and the activated belief rules, and the weight for each activated belief rule to generate the updated weight of each activated belief rule for the observation, we can draw the conclusion that the weights of the four pieces of evidence from the different input variables have an impact on the updated weight of each activated belief rule for the observation. With the belief rules activated by the observation: {5, 2.3, 3.3, 1} and the updated weight for each of these activated belief rules, we can combine these belief rules activated by the observation to predict the probability of each class of the output variable of the observation, using the conjunctive MAKER rule which is shown in Equations (3.14) and (3.15).

**Table 5.9 The combinations of referential values of the input variables of the training set of the first fold of the Iris data set activated by the observation: {5, 2.3, 3.3, 1}**

| combination of activated referential values/input variable | sepal length | sepal width | petal length | petal width |
|---|---|---|---|---|
| combination 1 | 4.9991 | 2 | 1 | 0.1 |
| combination 2 | 4.9991 | 2 | 1 | 1.3389 |
| combination 3 | 4.9991 | 2 | 4.4044 | 0.1 |
| combination 4 | 4.9991 | 2 | 4.4044 | 1.3389 |
| combination 5 | 4.9991 | 2.8969 | 1 | 0.1 |
| combination 6 | 4.9991 | 2.8969 | 1 | 1.3389 |
| combination 7 | 4.9991 | 2.8969 | 4.4044 | 0.1 |
| combination 8 | 4.9991 | 2.8969 | 4.4044 | 1.3389 |
| combination 9 | 7.7 | 2 | 1 | 0.1 |
| combination 10 | 7.7 | 2 | 1 | 1.3389 |
| combination 11 | 7.7 | 2 | 4.4044 | 0.1 |
| combination 12 | 7.7 | 2 | 4.4044 | 1.3389 |
| combination 13 | 7.7 | 2.8969 | 1 | 0.1 |
| combination 14 | 7.7 | 2.8969 | 1 | 1.3389 |
| combination 15 | 7.7 | 2.8969 | 4.4044 | 0.1 |
| combination 16 | 7.7 | 2.8969 | 4.4044 | 1.3389 |

On the basis of the predictions for all the observations of the training set, we can generate the MSE for the set of parameters including the referential values and the weights used to establish a classifier for the training set. This set of parameters is referred to as an individual of the population in the adapted genetic algorithm described in Section 3.6. The MSE is calculated for all the individuals of the population. The individuals of the population and their MSEs are used in the adapted genetic algorithm, which helps to achieve the target of optimizing a MAKER-based classifier for the Iris data set, which in turn maximizes the predicted outputs, i.e., predicted probabilities, of the classifier for the true observed outputs of the training set to minimize the MSE for the training set.

As stated previously, an individual of the population in the adapted genetic algorithm is composed of the referential values of the four input variables and the weights of the evidential elements of all pieces of evidence, each of which contains a referential value, of the four input variables. Among the weights of the individuals of population in the adapted genetic algorithm, the weights of evidential elements of the four pieces of evidence from the different input variables have an impact on the updated weight of each activated belief rule for an observation used to predict the probabilities of the classes of this observation. Thus, we can maximize the predicted outputs, i.e., predicted probabilities, of the classifier for the true observed outputs of the training set to minimize the MSE for the training set by optimizing the referential values of the four input variables and the weights of the evidential elements. The optimal individual (solution) of the population acquired from the optimization based on the adapted genetic algorithm of a MAKER-based classifier for the Iris data set can make the predicted outputs, i.e., predicted probabilities, of the classifier as close to the true observed outputs of the training set as possible.

In this case study of the Iris data set, as mentioned previously, the optimized individual (solution) of the population used in this case study of the Iris data set, is obtained from the optimization of the MAKER-based classifier, which has one optimized referential value for each of the input variables of the training set. We use this optimized individual (solution) to construct a MAKER-based classifier for the purpose of illustration. Based on the referential values and weights acquired from this optimized individual (solution), we can use the MAKER-based classifier established by the process described earlier in this section to make a prediction about the observation: {5, 2.3, 3.3, 1}. The predicted probability of the class of *Iris setosa* for this observation is 0.1079, and that of the class of *Iris versicolor* for this observation is 0.8288, and that of the class of *Iris virginica* is 0.0632. In other words, if the sepal length, sepal width, petal length, and petal width of a flower are

5, 2.3, 3.3, and 1 respectively, the probability of this flower being *Iris setosa* is 0.1079, and that of this flower being *Iris versicolor* is 0.8288, and that of this flower being *Iris virginica* is 0.0632. From the process used to establish the MAKER-based classifier for this case study of the Iris data set, it is evident that the rule-based inferential modelling and prediction approach used to do this is an interpretable approach integrating statistical analysis, belief rule-base inference, and maximum likelihood prediction and machine learning.

## 5.7 Performance Comparative Analysis for Classical Data Sets

In this section, we will carry out a performance comparative analysis on the three classical datasets: the Bananas data set, the Haberman's survival dataset, and the Iris data set. On one side of the comparison is the MAKER-based classifier constructed using a rule-based, inferential modelling and prediction approach; on the other side are the traditional classifiers including classification trees, discriminant analysis, logistic regression, support vector machines (SVM), k-nearest neighbors (KNN), ensembles, and Naïve Bayes.

As mentioned in Section 3.3, before being downloaded from the KEEL-dataset repository, each of the three classical data sets had already been divided into five folds using optimally balanced stratified cross-validation (Alcalá-Fdez, et al., 2011). This ensured all the separate folds have a similar class distribution, equal to that of the entire dataset (Aggarwal, 2015). As usual in cross validation, the data in each successive fold represents test data for the training data in the other folds.

**Table 5.10 The alternative variants of the classifiers except the MAKER-based classifiers for the classical datasets**

| Classifier | Variants of Classifier | Selected Variant of Classifier |
|---|---|---|
| decision trees | simple tree, medium tree, and complex tree | complex tree |
| discriminant analysis | linear discriminant and quadratic discriminant | quadratic discriminant |
| logistic regression | logistic regression | logistic regression |
| support vector machine (SVM) | linear SVM, quadratic SVM, cubic SVM, fine Gaussian SVM, medium Gaussian SVM, and coarse Gaussian SVM | fine Gaussian SVM |
| k-nearest neighbour (KNN) | fine KNN, medium KNN, coarse KNN, cosine KNN, cubic KNN, and weighted KNN | fine KNN and weighted KNN |
| ensembles | boosted trees, bagged trees, subspace discriminant, subspace KNN, and RUSBoosted trees | bagged trees and subspace KNN |
| Naïve Bayes | Naïve Bayes | Naïve Bayes |

For each classifier variant listed in the central column of Table 5.10, a model was constructed using each training set of the three classical datasets: Bananas, Haberman's and Iris. We selected the classifier variant with the highest average accuracy against the training sets as the chosen representative of this classifier type. Table 5.10 lists the alternative variants and the selected variant of each alternative classifier for each classical dataset in this case study.

In addition, according to the stopping criterion introduced in Section 4.5, all the training for the MAKER-based classifiers was stopped when there were five trained referential values for each of the input variables of the training sets. For the Banana data set, we selected the MAKER-based classifiers with five trained referential

values for each input variable. For the Haberman's survival data set and the Iris data set, we selected the MAKER-based classifiers with one trained referential value for each input variable. Each of these selected MAKER-based classifiers are compared with other alternative classifiers for the corresponding classical dataset, as each of these MAKER-based classifiers with the same number of trained referential values for each input variable have the highest average diagnostic accuracy among all the trained MAKER-based classifiers with different numbers trained referential values for each input variable for the corresponding classical data set.

To compare the MAKER-based classifiers with other alternative classifiers, we need measures of performance. These can include sensitivity, specificity, diagnostic accuracy, and the area under the receiver operating characteristic curve (AUC). According to Ling, Huang and Zhang (2003), accuracy is a widely-used measure for comparing the predictive capability of different classifiers. Most classifiers generate probability estimations of the classification, but they are completely ignored in the accuracy (Ling, Huang and Zhang, 2003). Ling, Huang and Zhang (2003) present arguments emphasising that AUC provides a better measure than the accuracy, with higher numbers up to a maximum of 1.0 being best.

In the following part of this section, we will present the Receiver Operating Characteristics (ROC) curve of the MAKER-based classifier in comparison with the alternative classifier with the optimal AUC. Figure 5.9 shows the comparisons for the Banana dataset; Figure 5.10 shows the equivalent results for Haberman's survival dataset, and Figure 5.11 shows the Iris dataset. We will display the AUC of each classifier for each test set of the Banana data set in Table 5.11, for that of the Haberman's survival data set in Table 5.12, and for that of the Iris data set in

Table 5.13. The average AUCs of classifiers over the five test sets will be calculated for each classical data set and we will present them in the above-mentioned tables.

**Table 5.11 The area under the receiver operating characteristic (ROC) curve (AUC) of each of the classifiers for the Banana dataset**

| Classifiers/Measures | AUC | | | | | |
|---|---|---|---|---|---|---|
| | Test1 | Test2 | Test3 | Test4 | Test5 | Avg. |
| MAKER | 0.95158 | 0.96072 | 0.96254 | 0.9606 | 0.96296 | **0.95968** |
| Complex tree | 0.93494 | 0.93956 | 0.93735 | 0.94741 | 0.94917 | 0.941686 |
| Quadratic discriminant | 0.64853 | 0.6471 | 0.65017 | 0.6516 | 0.65386 | 0.650252 |
| Logistic regression | 0.54892 | 0.54909 | 0.54775 | 0.5495 | 0.55027 | 0.549106 |
| Fine Gaussian SVM | 0.94302 | 0.95581 | 0.96107 | 0.95938 | 0.96611 | 0.957078 |
| Fine KNN | 0.87788 | 0.89556 | 0.89293 | 0.86254 | 0.86118 | 0.878018 |
| Weighted KNN | 0.95471 | 0.96592 | 0.96757 | 0.959 | 0.96363 | 0.962166 |
| Ensemble: bagged trees | 0.94867 | 0.96139 | 0.96088 | 0.9591 | 0.96396 | 0.9588 |
| Ensemble: subspace KNN | 0.63064 | 0.60058 | 0.62289 | 0.60923 | 0.62424 | 0.617516 |
| Naïve Bayes | 0.66185 | 0.66148 | 0.66561 | 0.6677 | 0.67017 | 0.665362 |

From Table 5.11, we can observe that the average AUC of the MAKER-based classifiers over the five test sets is 0.95968, highlighted in bold, the second largest AUC among all the AUCs of the ten classifiers for the Banana data set. According to Smithson and Merkle (2014), a general rule of thumb for using AUC to judge the classification capability of a classifier is that an AUC between 0.7 and 0.8 is considered acceptable, between 0.8 and 0.9 indicates excellent discrimination, and larger than 0.9 implies outstanding discrimination. As the average AUC of the MAKER-based classifiers is 0.95968, the MAKER-based classifiers for the Banana dataset can be considered outstanding.

The average AUC of the MAKER-based classifiers over the five test sets is surpassed only by that of the weighted KNN over five test sets. In addition, both the logistic

regression classifiers and the naïve Bayes classifiers are capable of meaningful interpretation, but their average AUCs over the five test sets are much lower than the MAKER-based classifiers. Furthermore, the average AUC of the complex tree classifiers over the five test sets is slightly lower than that of the MAKER-based classifiers. This suggests that the simple interpretable classifiers (e.g. logistic regression and Naïve Bayes) cannot work as well in the Banana data set as the complex interpretable classifiers, e.g. complex tree, whose performance is similar to that of the MAKER-based classifiers. This is because the Banana data set is complex (as shown graphically in Figure 5.8).

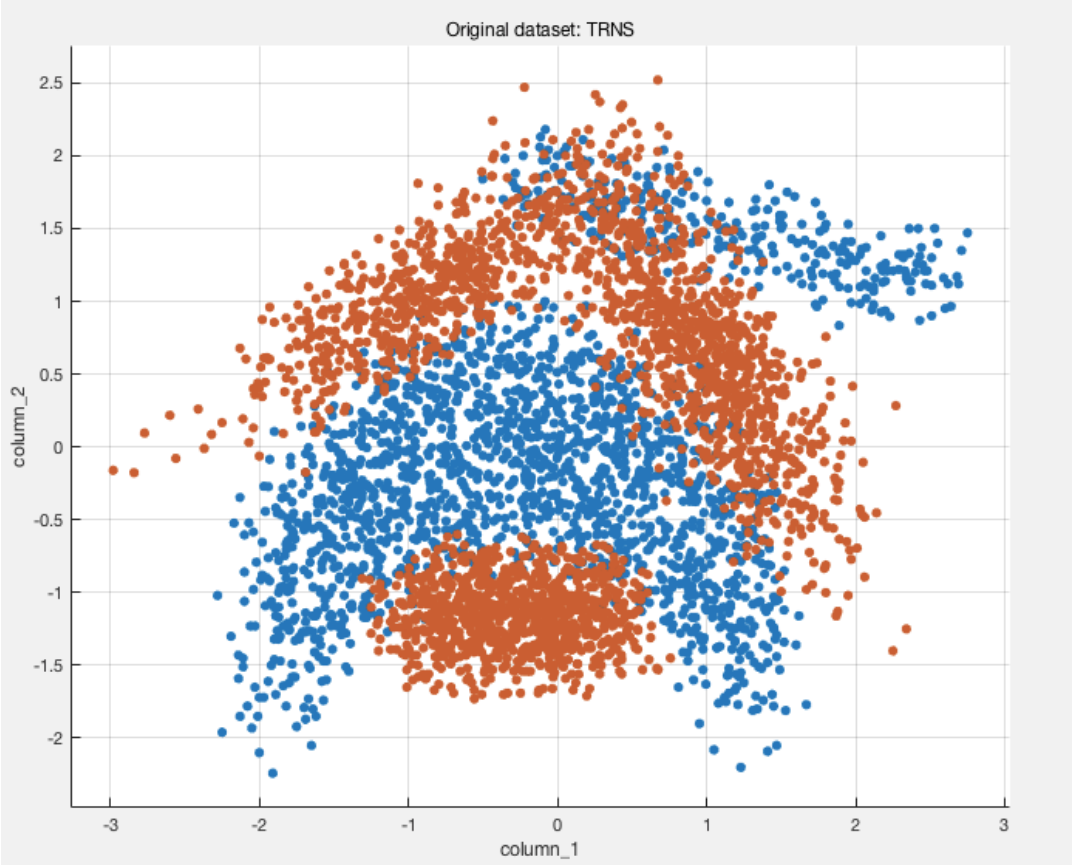
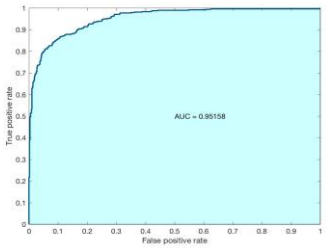
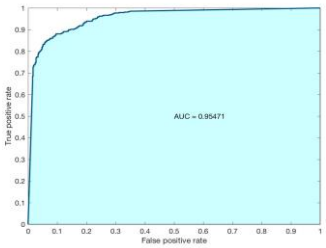


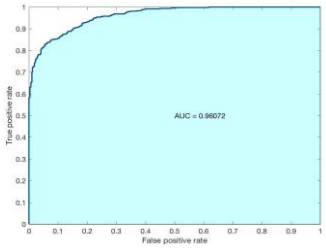

**Figure 5.8 The scatter plot of the Banana data set**



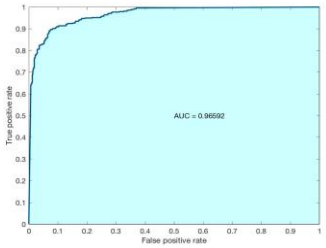

MAKER for the 1st test set



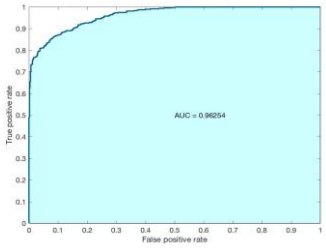

Weighted KNN for the 1st test set



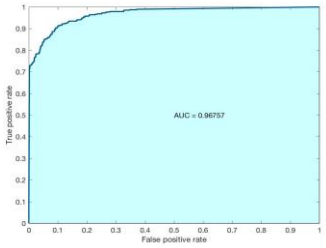

MAKER for the 2nd test set



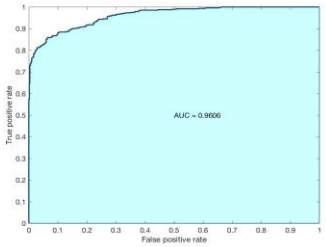

Weighted KNN for the 2nd test set



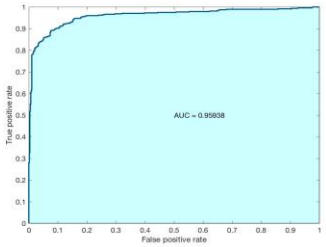

MAKER for the 3rd test set



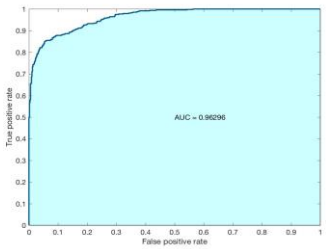

Weighted KNN for the 3rd test set



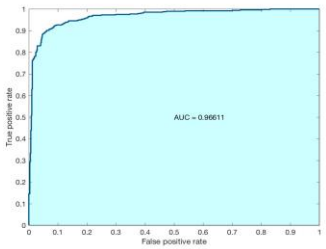

MAKER for the 4th test set




Fine Gaussian SVM for the 4th test set

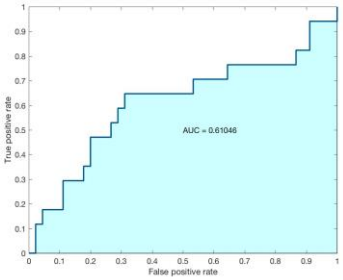



MAKER for the 5th test set

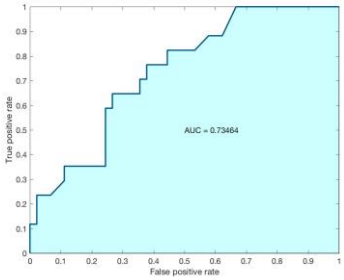



Fine Gaussian SVM for the 5th test set

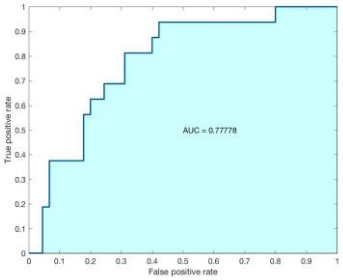
**Figure 5.9 The ROC curve of the Maker-based classifier and that of the classifier with the optimal AUC among all the alternative classifiers for the test sets of the Banana dataset**



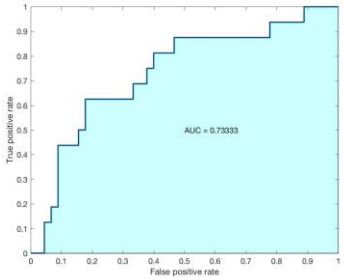

MAKER for the 1st test set



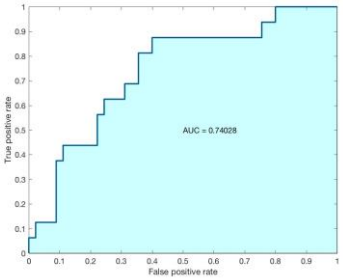

Ensemble: bagged trees for the 1st



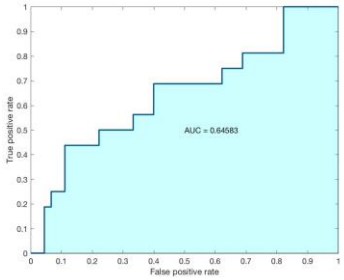

MAKER for the 2nd test set



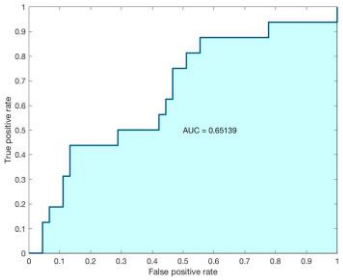

Quadratic discriminant for the 2nd



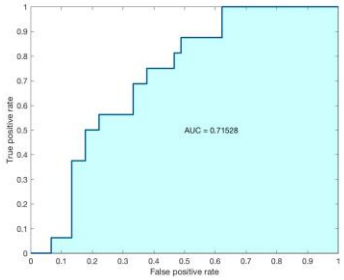

MAKER for the 3rd test set



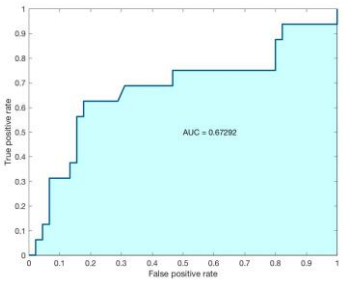

Logistic regression for the 3rd test set



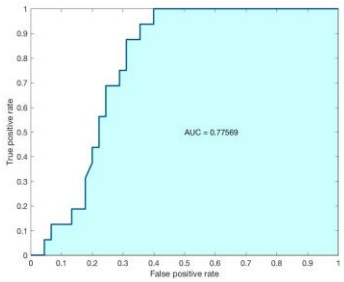

MAKER for the 4th test set




Fine Gaussian SVM for the 4th test set




MAKER for the 5th test set




Weighted KNN for the 5th test set

**Figure 5.10 The ROC curve of the MAKER-based classifier and that of the classifier with the optimal AUC among all the alternative classifiers for the test sets of the Haberman's survival dataset**

**Table 5.12 The area under the receiver operating characteristic (ROC) curve (AUC) of each of the classifiers for the Haberman's survival dataset**

| Classifiers/Measures | AUC | | | | | |
|---|---|---|---|---|---|---|
| | Test1 | Test2 | Test3 | Test4 | Test5 | Avg. |
| MAKER | 0.61046 | 0.77778 | 0.74028 | 0.65139 | 0.67292 | **0.690566** |
| Complex tree | 0.53464 | 0.62361 | 0.59931 | 0.50069 | 0.56597 | 0.564844 |
| Quadratic discriminant | 0.71634 | 0.73333 | 0.62222 | 0.6875 | 0.80069 | 0.712016 |
| Logistic regression | 0.67843 | 0.71806 | 0.64583 | 0.6375 | 0.73542 | 0.683048 |
| Fine Gaussian SVM | 0.71503 | 0.65694 | 0.56528 | 0.71528 | 0.71736 | 0.673978 |
| Fine KNN | 0.59869 | 0.57639 | 0.56528 | 0.62986 | 0.56528 | 0.5871 |
| Weighted KNN | 0.69673 | 0.64236 | 0.62778 | 0.69306 | 0.77569 | 0.687124 |
| Ensemble: bagged trees | 0.73464 | 0.67014 | 0.62361 | 0.66806 | 0.68819 | 0.676928 |
| Ensemble: subspace KNN | 0.55948 | 0.66806 | 0.63611 | 0.625 | 0.58333 | 0.614396 |
| Naïve Bayes | 0.69542 | 0.70139 | 0.62222 | 0.58333 | 0.69097 | 0.658666 |

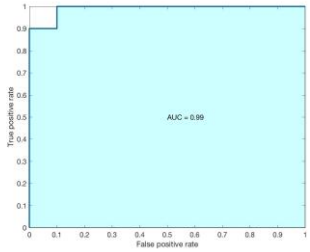
From Table 5.12, we can find that the average AUC of the MAKER-based classifiers for the Haberman's survival dataset is 0.690566, highlighted in bold, the second largest AUC among all the AUCs of the ten classifiers. As the Haberman's survival dataset is an imbalanced dataset of which the ratio of the number of positive observations to that of negative observations is about 1:3, the performance of the MAKER-based classifiers is generally acceptable. In addition to this, the MAKER-based classifier performs better in terms of the average AUC over the five test sets than other interpretable classifiers, the complex tree and logistic regression classifiers. From what has been analysed above, we can conclude that the MAKER-based classifier is a good classifier for the Haberman's survival dataset.

As mentioned in Section 3.3, the output variable of the Iris dataset is composed of three classes, the three species of Iris. Hence, the logistic regression classifier is not applicable to the Iris dataset since it can only split data into one of two classes. In the Iris dataset, we can take the class *Iris Versicolor* as the positive class and take both *Iris Setosa* and *Iris Virginica* as the combined negative class to plot the

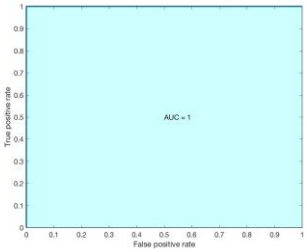
figure of ROC curve and calculate the AUC for each classifier. From Table 5.13, we can see that the average AUC of the MAKER-based classifiers over the five test sets of the Iris data set is 0.9955, highlighted in bold, which is very close to 1, i.e., the AUC of the perfect classifier and is the second largest AUC among all the AUCs of the ten classifiers for the Iris data set. The average AUCs of other alternative classifiers are also very close to 1. Thus, it can be concluded that the MAKER-based classifier is an outstanding classifier for the Iris data set. Overall, taking all the above-mentioned results of AUC into consideration, we can conclude that the MAKER-based classifier is an outstanding classifier for the classical data sets: the Banana data set, the Haberman's survival data set, and the Iris data set, and it generally performs better than other interpretable classifiers, e.g. complex tree, logistic regression, and Naïve Bayes.

**Table 5.13 The area under the receiver operating characteristic (ROC) curve (AUC) of each of the classifiers for the Iris dataset**

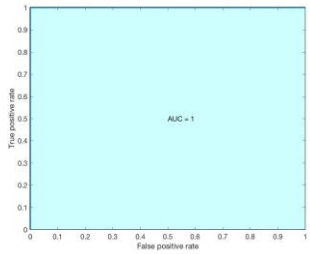
| Classifiers/Measures | AUC | | | | | |
|---|---|---|---|---|---|---|
| | Test1 | Test2 | Test3 | Test4 | Test5 | Avg. |
| MAKER | 0.99 | 1 | 0.9925 | 1 | 0.995 | **0.9955** |
| Complex tree | 0.9 | 0.975 | 0.9725 | 0.975 | 0.9175 | 0.948 |
| Quadratic discriminant | 1 | 1 | 1 | 1 | 1 | 1 |
| Fine Gaussian SVM | 0.995 | 0.99 | 0.995 | 1 | 0.95 | 0.986 |
| Fine KNN | 0.95 | 0.975 | 0.95 | 0.95 | 0.875 | 0.94 |
| Weighted KNN | 1 | 0.995 | 1 | 1 | 0.985 | 0.996 |
| Ensemble: bagged trees | 0.995 | 1 | 0.995 | 1 | 0.965 | 0.991 |
| Ensemble: subspace KNN | 1 | 1 | 0.9925 | 0.9875 | 0.9725 | 0.9905 |
| Naïve Bayes | 0.995 | 0.995 | 0.995 | 0.99 | 0.99 | 0.993 |



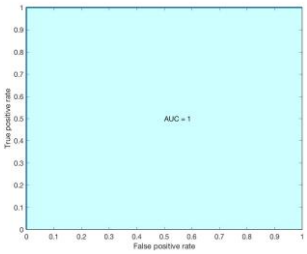

MAKER for the 1st test set



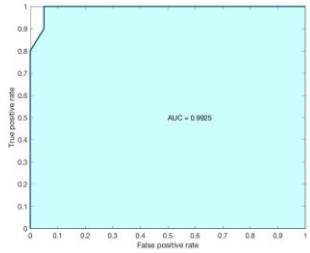

Quadratic discriminant for the 1st test



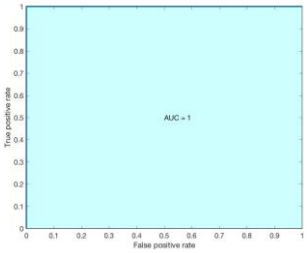

MAKER for the 2nd test set



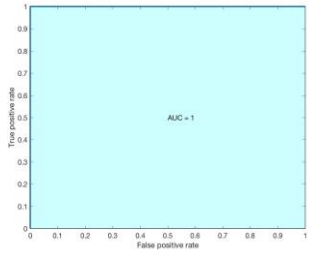

Quadratic discriminant for the 2nd test



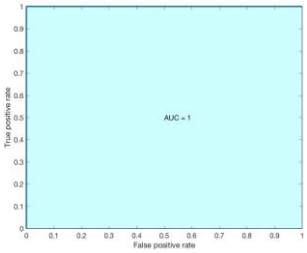

MAKER for the 3rd test set



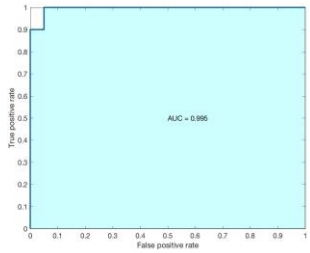

Quadratic discriminant for the 3rd test



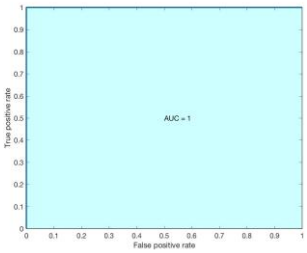

MAKER for the 4th test set




Quadratic discriminant for the 4th test




MAKER for the 5th test set




Quadratic discriminant for the 5th test

**Figure 5.11 The ROC curve of the MAKER-based classifier and that of the classifier with the optimal AUC among all the alternative classifiers for the test sets of the Iris dataset**

## 5.8 Summary

This chapter presents the rule-based inferential modelling and prediction approach from the perspectives of: fundamental knowledge, theoretical comparative analysis, case study, and performance comparative analysis. It begins by describing the fundamental knowledge of this approach from the perspectives of: statistical analysis, belief rule-base inference, and maximum likelihood prediction and machine learning. Next, the modelling kernel and the inference mechanisms of the rule-based inferential modelling and prediction approach are presented, emphasising the unique interpretability of this approach. Subsequently, the importance of interpretability is highlighted, and MAKER-based classifiers constructed by this approach are compared theoretically to other alternative classifiers. MAKER-based classifiers have the unique interpretability that the complex classifiers (e.g. ensembles, artificial neural networks, and random forests) do not have. In addition, MAKER-based classifiers are generally better than other interpretable classifiers (e.g. decision tree, logistic regression, and naïve Bayes). Next, a case study is used to demonstrate how to build a MAKER-based classifier with the rule-based inferential modelling and prediction approach, from the perspectives of: evidence acquisition from data, analysis of evidence independence, belief rule-base inference, and maximum likelihood prediction and machine learning. Finally, a performance comparative analysis is conducted between the MAKER-based classifier and other alternative classifiers for classical data sets, including: the Banana data set, the Haberman's survival data set, and the Iris data set. The performance comparative analysis shows that the MAKER-based classifier is an outstanding classifier for the classical data sets and it generally performs better than other interpretable classifiers (e.g. complex tree, logistic regression, and Naïve Bayes).

# Chapter 6 Application to Sepsis Diagnosis

## 6.1 Introduction

This chapter reports on the application of the rule-based inferential modelling and prediction approach based on the MAKER framework to sepsis diagnosis. The rest of this chapter is organized as follows. Section 6.2 presents the data preparation for this application of the rule-based inferential and modelling approach, which mainly comprises data cleaning, data transformation, and data partitioning. Section 6.3 describes how the classifier based on the MAKER framework is built on the basis of the rule-based inferential modelling and prediction approach. In Section 6.4, a performance comparative analysis is performed between the classification results of the MAKER-based classifier and those of alternative classifiers. Section 6.5 provides a summary of Chapter 6.

## 6.2 Data Preparation

Data preparation comprises the techniques used to transform raw data into quality usable data, including data integration, data transformation, data cleaning, data reduction and others (Zhang, Zhang and Yang, 2003). Data in the real world may be incomplete, noisy, and inconsistent, which could disguise useful patterns (Zhang, Zhang and Yang, 2003). Hence, it is necessary to prepare the raw data before beginning modelling and prediction.

For this research, the data preparation mainly consisted of data cleaning, data transformation, and data partitioning. Data cleaning is a process that is used to identify imprecise, incomplete, and unreasonable data and then to improve the quality of the data by deleting any errors and omissions (Chapman, 2005). In the

original sepsis data set, values of '99999' are considered as missing values and any observations with these values were completely deleted prior to modelling and prediction.

Data transformation, which is a key concept of data preparation, consists of transforming or merging data into a form in which learning can be applied (Dua and Chowriappa, 2013). Data generalization is one of the strategies of data transformation and is applied when abstraction of data is needed (Dua and Chowriappa, 2013).

As is indicated in Section 3.3, the output variable of the original sepsis data set, i.e., the patient groups, include five classes i.e., sepsis-1, sepsis-2, sepsis-3, unknown, and non-sepsis. Among these classes, sepsis-1, sepsis-2, and sepsis-3 indicate that the patient has sepsis, while non-sepsis indicates that the patient does not. Unknown implies that the patient may or may not have sepsis.

The sepsis data set used in this research does not include the class of 'unknown', and in the future study, on the basis of the data set containing the class of 'unknown', we will use the approach of rule-based inferential modelling and prediction to predict how likely the patients with the records of 'unknown' will get sepsis. In this future study, the classes of 'unknown', 'sepsis', and 'non-sepsis' are treated equally. We will follow the similar research steps to acquire evidence from the data set containing the class of 'unknown', analyse interdependence between pairs of evidence, combine multiple pieces of evidence, perform maximum likelihood prediction, and etc. The difference is in the step of evidence combination, and we will use a slightly different way to calculate the joint support of two pieces of evidence pointing to different classes in Equation (3.14) or (3.15). We will use the training data set containing the class of 'unknown' to train a MAKER-based model. This model will be used to predict the probabilities of the patients with records of 'unknown' in the test data set having sepsis or not.

Due to the fact that there are only a few observations in sepsis-1 and sepsis-2, and sepsis-1, sepsis-2, and sepsis-3 all indicate that patients have sepsis, we can just consolidate sepsis-1, sepsis-2, and sepsis-3 into one class, i.e., sepsis, which indicates that the patient has sepsis, so that there are plenty of observations in this class. We can then divide the sepsis class into a number of folds for cross-validation of models of sepsis diagnosis.

In data partitioning, the data are divided into training sets and test sets (Olson and Delen, 2008). Data partitioning can be as complicated as partitioning data into k disjoint subsets (a.k.a. k-fold cross-validation) of which each subset has one or more variables used as strata to maintain the proportional representations of different subgroups for stratified random sampling. Each subset is used as the test set and the remaining subsets are used as training sets (Olson and Delen, 2008). In this method of data partitioning, we can train and test classifiers repeatedly on different subsets of data to make near-optimal use of the available data (Molinaro, Simon and Pfeiffer, 2005).

In this research, we employed a method based on stratified random sampling to partition the sepsis data set into four folds for cross-validation of models of sepsis diagnosis. Specifically, we partitioned the observations in each class of the sepsis data set into a number of bins according to breakpoints in each input variable of the sepsis data set. The breakpoints for the partitions were optimized so as to make each bin with more than three observations contain as many observations as possible. Then we performed random sampling in each bin with more than three observations to obtain four sets of observations. In this way, the sepsis data set was partitioned into four folds, all having similar class distributions, and with observations belonging to the same class having similar probability density functions.

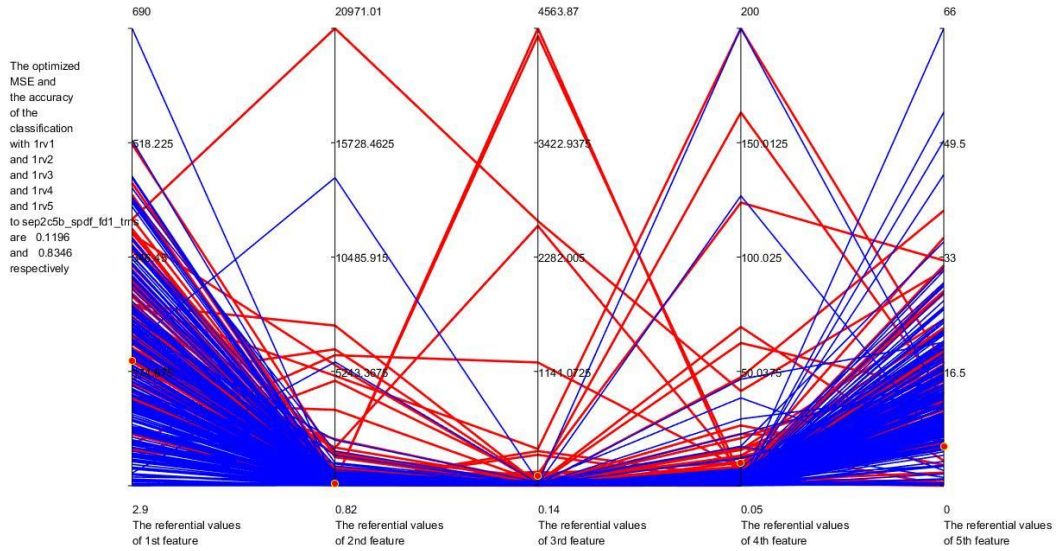



**Figure 6.1 Parallel Coordinates Plot of Distribution of Observations of Different Classes across Five Input Variables of the Training Set of the First Fold of the Sepsis Data Set and the Trained Referential Values of the Input Variables of the Data Set**

## 6.3 New Models for Classification of the Sepsis Data Sets

In order to demonstrate how the rule-based inferential modelling and prediction approach can be used to build classifiers based on the system of the MAKER framework for the classification of sepsis data sets, a numerical study using the sepsis data set is presented in the remainder of this section. As previously stated, there are five input variables, CRP, IL6, IL10, PCT, and WCC, in the sepsis data set. The output variable, diagnosis, contains two classes, sepsis and non-sepsis. In addition, the sepsis data set has been partitioned into four folds, all having similar class distributions and with observations belonging to the same class having similar probability density functions. Hence, we can take the training set of the first fold of the sepsis data set as an example to illustrate how we use the rule-based inferential modelling and prediction approach to establish the classifiers based on the system of the MAKER framework to classify the data sets generated from the sepsis data set.

Figure 6.1 exhibits the parallel coordinates plot of the general distribution of observations of different classes across five input variables of the training set of the first fold of the sepsis data set and the locations of the trained referential values of each input variable of the data set. The red solid lines indicate the sepsis observations and the blue solid lines the non-sepsis observations. The five vertical axes in Figure 6.1 from left to right represent CRP, IL6, IL10, PCT, and WCC, respectively, which are all the input variables of the data set. The red nodes with yellow edges on the axes of Figure 6.1 denote the trained referential values of the input variables. It can be found from Figure 6.1 that the distributions of the observations of different classes across the five input variables have different patterns. For CRP and WCC, we can see that the observations of both the sepsis and the non-sepsis class are generally distributed in the same range. For IL6, IL10, and PCT, the majority of the observations of the non-sepsis class are concentrated close to the respective minimum values and the observations of the sepsis class are generally distributed over a larger range of values than those of the non-sepsis class.

### 6.3.1 The Optimised Referential Values of the Model

As introduced in Chapter 3, we can use the adapted genetic algorithm to optimize the parameters, i.e., the referential values and weights of the models based on the system of the MAKER framework for function approximation or classification. In the adapted genetic algorithm for the optimization of the MAKER-based models of sepsis diagnosis, the initial population of individuals (chromosomes) includes 10 subpopulations, each of which comprises 20 individuals (chromosomes). Each individual of a population contains both the referential values of observed values of input variables of a training set generated from the sepsis data set, and the weight

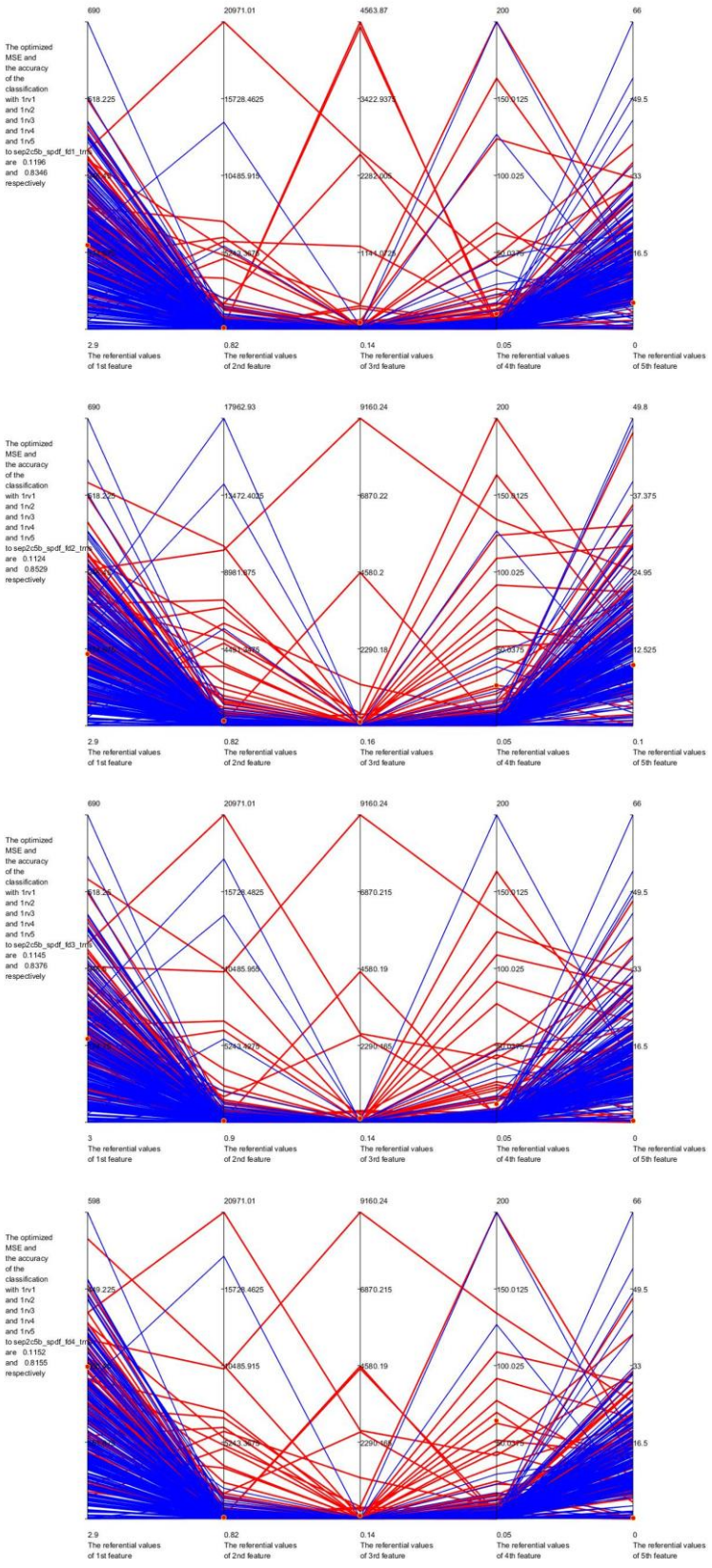



**Figure 6.2**

**The Distribution of Trained Referential Values Obtained from the Optimization of MAKER-based Models on Sepsis Diagnosis Which All Have 1 Trained Referential Value in Each of the Input Variables of A Training Set Generated from the Sepsis Data Set**

(reliabilities) of these referential values under different classes of the output variable of the data set. After the initial population has been generated, the objective value, i.e., mean squared error (MSE) is calculated for each individual (solution) of a population. Then, on the basis of a population of the individuals as stated above, a group of genetic algorithm operations, e.g., selection, recombination, mutation, reinsertion, and migration, is performed for 200 iterations to obtain an optimized solution for referential values and weights.

The target of the optimization of a MAKER-based model of sepsis diagnosis is to maximize the predicted outputs, i.e., predicted degrees of belief, of the model for the true observed outputs of a training set generated from the sepsis data set, so as to minimize the MSE, which is used to measure the difference between the observed outputs and the predicted outputs. Thus, it can reasonably be inferred that, through optimization, trained referential values of each input variable will generally be located around critical observed values that divide the observed values of this input variable into several parts, in each of which the observations of a given class are in the majority.

This is supported by Figure 6.2, which presents the distribution of trained referential values obtained from the optimization of the MAKER-based models of sepsis diagnosis, all of which have one trained referential value for each input variable of a training set generated from the sepsis data set. It is worth noting that the 'trained referential values' refer to the observed values of an input variable of a data set, between the minimum and maximum of those observed values. From Figure 6.2, we can see clearly that each of the trained referential values obtained from optimization for the input variables IL6, IL10, and PCT is generally located around the respective critical value that divides the observed values of the corresponding input variable into two parts, where in one part sepsis observations are in the majority and in the other part non-sepsis observations are in the majority.

As stated at the beginning of Section 6.3, the training set of the first fold of the sepsis data set is taken as an example to illustrate how the MAKER-based classifiers are established for the sepsis data set. Hence, we just use the optimized solution obtained from the optimization of the MAKER-based classifier (model) for the training set of the first fold of the sepsis data set to establish a MAKER-based classifier on the training set for the purpose of illustration. The optimized solution includes optimized referential values and optimized weights, and the optimization generating this optimized solution has one optimized referential value for each of the input variables of the training set. The 'optimized referential values' have the same meaning with the 'trained referential values' stated in Page 230.

In the following part of this section, we will show how to apply the rule-based inferential modelling and prediction approach to build the MAKER-based classifier for the training set of the first fold of the sepsis data set (henceforth the 'training set') from four aspects: evidence acquisition from data, analysis of evidence independence, belief rule-base inference, and maximum likelihood prediction and machine learning. The complete above-mentioned training set is provided in Appendix A.

### 6.3.2 Evidence Acquisition from Data

As introduced in Section 3.3, the first step in establishing a MAKER-based classifier is to acquire evidence from data. In order to acquire evidence from a data set, we need to decide on referential values for each of the input variables of the data set. Referential values, as adjustable parameters, can initially be determined by expertise, or by a random rule without prior knowledge, and can subsequently be trained using an input-output data set under a certain optimization objective (Xu et al., 2017).

**Table 6.1 The referential values obtained from the optimization of the MAKER-based classifier for the training set of the first fold of the sepsis data set**

| Input variables | CRP | IL6 | IL10 | PCT | WCC |
|---|---|---|---|---|---|
| Boundary referential values (minima) | 2.9000 | 0.8200 | 0.1400 | 0.0500 | 0 |
| Optimized referential values | 190.2799 | 101.9637 | 97.6625 | 9.8923 | 5.7066 |
| Boundary referential values (maxima) | 690.0000 | 20971.0100 | 4563.8700 | 200.0000 | 66.0000 |

Based on what has been mentioned in Page 231 of Section 6.3.1, we just use the optimized referential values obtained from the optimization of the MAKER-based classifier for the 'training set' of the sepsis data set, to acquire evidence from that training set.

Table 6.1 displays the referential values, including the boundary referential values and the optimized referential values, used to acquire evidence from the data. The boundary referential values are the minima and maxima of the observed values of the input variables of the data set. On the basis of the optimized referential values displayed in Table 6.1 and the boundary referential values as stated above, we can use Equation (3.4) from Section 3.5.1 to transform each observed value of each input variable of the training set into the belief distributions of the two adjacent referential values between which this observed value is located. After all the observed values of the input variables are transformed into belief distributions of referential values, we will use Equation (3.5) to aggregate similarity degrees of belief distributions according to the referential values under different classes of the output variable of the training set. In this way, we will generate the frequencies of the referential values under different classes of the output variable, displayed in the form of Table 3.3.

**Table 6.2 The probabilities with which the referential values of the observed values of the input variable of CRP point to different classes of the output variable of the training set of the first fold of the sepsis data set**

| class/referential value | 2.9000 | 190.2799 | 690.0000 |
|---|---|---|---|
| sepsis | 0.2274 | 0.5657 | 0.6488 |
| non-sepsis | 0.7726 | 0.4343 | 0.3512 |

**Table 6.3 The probabilities with which the referential values of the observed values of the input variable of IL6 point to different classes of the output variable of the training set of the first fold of the sepsis data set**

| class/referential value | 0.8200 | 101.9637 | 20971.0100 |
|---|---|---|---|
| sepsis | 0.1270 | 0.6150 | 0.8432 |
| non-sepsis | 0.8730 | 0.3850 | 0.1568 |

Then, using Equation (3.6), we calculate the likelihood of a referential value of an input variable being true given that a class of the output variable of the training set is true, for all the referential values of input variables under different classes of output variable, displayed in the form of Table 3.4. With these likelihoods, the

probability of a referential value of an input variable pointing to a class of the output variable of the training set can be obtained from Equation (3.7) for all the referential values of input variables under different classes of output variable, shown in the form of Table 3.5.

Tables 6.2 and 6.3 exhibit the probabilities of the referential values of the observed values of the input variables CRP and IL6, respectively, pointing to different classes of the output variable of the training set. It is worth noting that the referential values displayed in Tables 6.2 and 6.3 include not only the boundary referential values as previously defined, i.e., 2.9000, 690.0000, 0.8200, and 20971.0100, but also the optimized referential values as previously defined, i.e., 190.2799 and 101.9637. From the probabilities of the referential values of the observed values of the input variables of the training set pointing to different classes of the output variable of the training set, partly shown in Tables 6.2 and 6.3, we can acquire a number of pieces of evidence. For example, as shown in Table 6.2, the probabilities 0.2274 and 0.7726, under boundary referential value 2.9000, indicate that, if the CRP test result for a patient is 2.9000, the probability of this patient having sepsis is 0.2274 and the probability of this patient not having sepsis is 0.7726. Thus, we can acquire a piece of evidence from the CRP test result of 2.9000, in that it points to the sepsis class with a probability of 0.2274 and points to the non-sepsis class with a probability of 0.7726.

### 6.3.3 Analysis of Evidence Interdependence

As stated previously, there are five input variables, i.e., CRP, IL6, IL10, PCT, and WCC, in the sepsis data set and the output variable, diagnosis, contains two classes, sepsis and non-sepsis. Obviously, the predictive power of a single piece of evidence is limited. In order to achieve greater predictive power, it is necessary to combine

multiple pieces of evidence to make a prediction for a patient. In the original evidential reasoning (ER) rule, any two pieces of evidence to be combined are assumed to be independent from each other, which is simplistic. Under the MAKER framework, the interdependence between two pieces of evidence is taken into consideration through the introduction of an interdependence index 'α' which is defined in Equation (3.13). To generate the interdependence index between each pair of pieces of evidence, we need to estimate the joint probabilities for these two pieces of evidence according to Equation (3.12) in advance. Table 6.4 displays the joint probabilities for all combinations of referential values of pieces of evidence from the input variables CRP and IL6, each of which points to different classes of output variable of the training set.

In Table 6.4, the first and second numbers of each combination of referential values represent the referential value of a piece of evidence from CRP and IL6 input variables of the training set respectively. On the basis of the probabilities displayed in Tables 6.2, 6.3, and 6.4, we can use Equation (3.13) to generate the interdependence indices between a piece of evidence from the input variable CRP and a piece from the input variable IL6, which are exhibited in Table 6.5. From Table 6.5, it is clear that the test results for CRP and IL6 are moderately independent from each other, as the independence indices displayed in Table 6.5 are between 1 and 5. For instance, the interdependence between CRP test result of 190.2799 and the IL6 test result of 101.9637 is moderate, as the interdependence index under the class of sepsis is 1.8679 and that under the class of non-sepsis is 2.0941.

**Table 6.4 The joint probabilities of different combinations of the referential value of a piece of evidence from the CRP input variable and that from the IL6 input variable pointing to different classes of the output variable of the training set of the first fold of the sepsis data set**

| class/combination of referential values | {2.9000, 0.8200} | {2.9000, 101.9637} | {2.9000, 20971.01} | {190.2799, 0.8200} | {190.2799, 101.9637} | {190.2799, 20971.01} | {690.0000, 0.8200} | {690.0000, 101.9637} | {690.0000, 20971.01} |
|---|---|---|---|---|---|---|---|---|---|
| sepsis | 0.0596 | 0.3922 | 0.8109 | 0.1810 | 0.6498 | 0.8312 | 0.2615 | 0.6614 | 0.9089 |
| non-sepsis | 0.9404 | 0.6078 | 0.1891 | 0.8190 | 0.3502 | 0.1688 | 0.7385 | 0.3386 | 0.0911 |

**Table 6.5 The interdependence indices between pieces of evidence from the CRP and IL6 input variables of the training set of the first fold of the sepsis data set**

| referential value of CRP | 2.9000 | 2.9000 | 2.9000 | 190.2799 | 190.2799 | 190.2799 | 690.0000 | 690.0000 | 690.0000 |
|---|---|---|---|---|---|---|---|---|---|
| class/referential value of IL6 | 0.8200 | 101.9637 | 20971.0100 | 0.8200 | 101.9637 | 20971.0100 | 0.8200 | 101.9637 | 20971.0100 |
| sepsis | 2.0634 | 2.8052 | 4.2298 | 2.5186 | 1.8679 | 1.7425 | 3.1723 | 1.6578 | 1.6614 |
| non-sepsis | 1.3943 | 2.0430 | 1.5610 | 2.1602 | 2.0941 | 2.4792 | 2.4087 | 2.5038 | 1.6542 |

**Table 6.6 The referential values of the input variables of the training set of the first fold of the sepsis data set activated by the observation {158.0000, 619.4500, 120.1000, 123.8600, 32.5000}**

| input variable | CRP | IL6 | IL10 | PCT | WCC |
|---|---|---|---|---|---|
| activated referential values | 2.9000, 190.2799 | 101.9637, 20971.0100 | 97.6625, 4563.8700 | 9.8923, 200.0000 | 5.7066, 66.0000 |

### 6.3.4 Belief Rule-base Inference

As we have acquired a number of pieces of evidence from the input variables of the training set, and we have analysed the interdependence between two pieces of evidence, we are now in a position to construct a belief rule base for inferring the likelihood of a patient having sepsis on the basis of their test results for the biomarkers CRP, IL6, IL10, PCT, and WCC. According to the belief rule described in (5.2) of Section 5.3, the antecedent of the belief rule, which is expressed in the form of '$if\, A_1^k \wedge A_2^k \wedge \ldots \wedge A_{T_k}^k$' in (5.2), in this case study of sepsis diagnosis should be expressed as 'if the test result of each biomarker of a patient is just equal to a referential value of the observed values of this biomarker'. The corresponding consequent of the belief rule, which is expressed in the form of '$then\, \{(D_1, \beta_{1k}), (D_2, \beta_{2k}), \ldots, (D_N, \beta_{Nk})\}$' in (5.2), in this case study of sepsis diagnosis should then be expressed as 'the probability of this patient having sepsis is equal to a value and that of this patient not having sepsis is equal to a value'.

To obtain the probability of a patient having sepsis and that of a patient not having sepsis in the consequent of a belief rule for this case study, we need to combine five pieces of evidence from the different biomarkers using the MAKER rule as described in Section 3.5.3. As stated previously, we use the optimized solution, including optimized referential values and optimized weights, with one optimized referential value for each of the input variables (biomarkers), to construct a MAKER-based classifier on the training set for the purpose of illustration. Thus, there are a total of three referential values, namely the boundary referential values as stated previously and an optimized referential value, for each of the five input variables of the training set.

Further, there are altogether 243 combinations of five pieces of evidence, each of which contains a referential value from different input variables of the training set,

that can be used to construct the belief rule base in this case study of sepsis diagnosis, as there are three referential values for each of the five input variables of the training set for this case study. In other words, there are 243 possible belief rules in the belief rule-base for the numerical example in this section. This complete belief rule-base is provided in Appendix B. On the basis of the previously mentioned optimized weights obtained from the optimization, we use Equation (3.14) to combine five pieces of evidence from different input variables of the training set to generate the probabilities of the sepsis and non-sepsis classes for each of the 243 combinations of five pieces of evidence. For example, through calculation, we can obtain a probability 0.0086 of sepsis and a probability 0.9914 of non-sepsis for the following combination of pieces of evidence:

**Table 6.7 The combinations of referential values of the input variables of the training set of the first fold of the sepsis data set activated by the observation {158.0000, 619.4500, 120.1000, 123.8600, 32.5000}**

| combination of activated referential values/input variable | CRP | IL6 | IL10 | PCT | WCC |
|---|---|---|---|---|---|
| combination 1 | 2.9000 | 101.9637 | 97.6625 | 9.8923 | 5.7066 |
| combination 2 | 2.9000 | 101.9637 | 97.6625 | 9.8923 | 66.0000 |
| combination 3 | 2.9000 | 101.9637 | 97.6625 | 200.0000 | 5.7066 |
| combination 4 | 2.9000 | 101.9637 | 97.6625 | 200.0000 | 66.0000 |
| combination 5 | 2.9000 | 101.9637 | 4563.8700 | 9.8923 | 5.7066 |
| combination 6 | 2.9000 | 101.9637 | 4563.8700 | 9.8923 | 66.0000 |
| combination 7 | 2.9000 | 101.9637 | 4563.8700 | 200.0000 | 5.7066 |
| combination 8 | 2.9000 | 101.9637 | 4563.8700 | 200.0000 | 66.0000 |
| combination 9 | 2.9000 | 20971.0100 | 97.6625 | 9.8923 | 5.7066 |
| combination 10 | 2.9000 | 20971.0100 | 97.6625 | 9.8923 | 66.0000 |
| combination 11 | 2.9000 | 20971.0100 | 97.6625 | 200.0000 | 5.7066 |
| combination 12 | 2.9000 | 20971.0100 | 97.6625 | 200.0000 | 66.0000 |
| combination 13 | 2.9000 | 20971.0100 | 4563.8700 | 9.8923 | 5.7066 |
| combination 14 | 2.9000 | 20971.0100 | 4563.8700 | 9.8923 | 66.0000 |
| combination 15 | 2.9000 | 20971.0100 | 4563.8700 | 200.0000 | 5.7066 |
| combination 16 | 2.9000 | 20971.0100 | 4563.8700 | 200.0000 | 66.0000 |
| combination 17 | 190.2799 | 101.9637 | 97.6625 | 9.8923 | 5.7066 |
| combination 18 | 190.2799 | 101.9637 | 97.6625 | 9.8923 | 66.0000 |

Continued on the next page

Continued from the previous page

| combination of activated referential values/input variable | CRP | IL6 | IL10 | PCT | WCC |
|---|---|---|---|---|---|
| combination 19 | 190.2799 | 101.9637 | 97.6625 | 200.0000 | 5.7066 |
| combination 20 | 190.2799 | 101.9637 | 97.6625 | 200.0000 | 66.0000 |
| combination 21 | 190.2799 | 101.9637 | 4563.8700 | 9.8923 | 5.7066 |
| combination 22 | 190.2799 | 101.9637 | 4563.8700 | 9.8923 | 66.0000 |
| combination 23 | 190.2799 | 101.9637 | 4563.8700 | 200.0000 | 5.7066 |
| combination 24 | 190.2799 | 101.9637 | 4563.8700 | 200.0000 | 66.0000 |
| combination 25 | 190.2799 | 20971.0100 | 97.6625 | 9.8923 | 5.7066 |
| combination 26 | 190.2799 | 20971.0100 | 97.6625 | 9.8923 | 66.0000 |
| combination 27 | 190.2799 | 20971.0100 | 97.6625 | 200.0000 | 5.7066 |
| combination 28 | 190.2799 | 20971.0100 | 97.6625 | 200.0000 | 66.0000 |
| combination 29 | 190.2799 | 20971.0100 | 4563.8700 | 9.8923 | 5.7066 |
| combination 30 | 190.2799 | 20971.0100 | 4563.8700 | 9.8923 | 66.0000 |
| combination 31 | 190.2799 | 20971.0100 | 4563.8700 | 200.0000 | 5.7066 |
| combination 32 | 190.2799 | 20971.0100 | 4563.8700 | 200.0000 | 66.0000 |

{2.9000, 0.8200, 0.1400, 0.0500, 66.0000}. Each observation of the training set will activate 32 belief rules out of the entire 243 belief rules in this case study. This is due to the fact that each observed value of the training set will activate two adjacent referential values, each of which belongs to a piece of evidence, of observed values of an input variable of the training set, between which this observed value is located, and there are five input variables, i.e., CRP, IL6, IL10, PCT, and WCC, in the training set. Hence, each observation of the training set activates $2^5 = 32$ combinations of five pieces of evidence, i.e., belief rules of the rule base in this case study.

Here, we take an observation {158.0000, 619.4500, 120.1000, 123.8600, 32.5000} from the training set as an example to demonstrate how an observation activates 32 rules from this case study's belief rule base for inference. Based on Table 6.1, the referential values activated by this observation, in terms of the input variables, are displayed in Table 6.6. Then, these activated referential values are used to generate the combinations of referential values for the 32 rules from this case

study's belief rule base that are activated by this same observation. These combinations of activated referential values are presented in Table 6.7. According to the 32 belief rules activated by each observation of the training set, we could find the corresponding probability of each class of the output variable of the training set that follows from each of these 32 belief rules. This could be used for the predicted probabilities of each observation.

### 6.3.5 Maximum Likelihood Prediction and Machine Learning

As already mentioned, each observation of the training set will activate 32 belief rules from this sepsis diagnosis case study's belief rule base that can be used to predict this observation. On the basis of these 32 belief rules, we need to calculate the degree of similarity between each observed value of this observation and each of the referential values between which this observed value is located, using Equation (3.16) presented in Section 3.5.4. The degree of similarity indicates how closely each observed value matches each of those referential values. For example, 158.0000 is an observed value of the observation {158.0000, 619.4500, 120.1000, 123.8600, 32.5000}, and according to Table 6.6, the referential values of the input variable CRP activated by the observed value 158.0000 are 2.9000 and 190.2799. On the basis of Equation (3.16), the degree of similarity between 158.0000 and 2.9000 is calculated as $\frac{190.2799-158.0000}{190.2799-2.9000} \approx 0.1723$ and that between 158.0000 and 190.2799 is calculated as $1-\frac{190.2799-158.0000}{190.2799-2.9000} \approx 0.8277$, which indicates that the observed value 158.0000 matches referential value 190.2799 to a high degree and referential value 2.9000 to a low degree. In this way, we can calculate the degree of similarity between each observed value of the observation {158.0000, 619.4500, 120.1000, 123.8600, 32.5000} and each of the referential values activated by this observed value. With these degrees of similarity, we can use Equation (3.17) to calculate the joint degree of similarity between the observation {158.0000,

619.4500, 120.1000, 123.8600, 32.5000} and the combination of referential values of each belief rule activated by this observation. This joint degree of similarity indicates the degree to which we should invoke the belief rules activated by an observation to predict the probability of each class of the output variable for this observation.

Having generated this joint degree of similarity, we are now in a position to combine the belief rules activated by the observation {158.0000, 619.4500, 120.1000, 123.8600, 32.5000} to predict the probability of each class of the output variable, i.e., the diagnosis for this observation. In order to combine these belief rules, we need their weights. Using Equation (3.19), we can generate their weights from the probability mass $m_{\theta,e(L)}$ ($\theta \subseteq \Theta, \theta \neq \emptyset$) and the probability $p_{\theta,e(L)}$ ($\theta \subseteq \Theta, \theta \neq \emptyset$) or just the probability mass $m_{P(\Theta),e(L)}$, which are shown in Equations (3.18) and (3.19).

As we use the weights of five pieces of evidence from different input variables of the training set to generate the probability mass $m_{\theta,e(L)}$ ($\theta \subseteq \Theta, \theta \neq \emptyset$), the probability $p_{\theta,e(L)}$ ($\theta \subseteq \Theta, \theta \neq \emptyset$), and the probability mass $m_{P(\Theta),e(L)}$ for each belief rule in this case study's belief rule base, and we use the relevant probability masses and relevant probability to generate the weight for each belief rule activated by the observation {158.0000, 619.4500, 120.1000, 123.8600, 32.5000}, we can conclude that the weights of the five pieces of evidence have an effect on the weights of each belief rule activated by this observation. On the basis of the joint degree of similarity between the observation and the activated belief rules, and the weight of each activated rule, we can generate the updated weight of each activated belief rule for the observation, which considers the degree to which we should invoke these activated belief rules in predicting the probability of each class of the observation. As the weights of the five pieces of evidence from the different input variables have an effect on the weight of each belief rule activated by this observation, and since we use the joint degree of similarity between the observation and the activated belief rules, and the weight for each activated rule

to generate the updated weight of each activated belief rule for the observation, we can draw the conclusion that the weights of the five pieces of evidence from the different input variables have an impact on the updated weight of each activated belief rule for the observation. With the belief rules activated by the observation {158.0000, 619.4500, 120.1000, 123.8600, 32.5000} and the updated weight for each of these activated rules, we can combine these belief rules activated by the observation to predict the probability of each class of the output variable of the observation, using the conjunctive MAKER rule which is shown in Equations (3.14) and (3.15).

On the basis of the predictions for all the observations of the training set, we can generate the MSE for the set of parameters including the referential values and the weights used to establish a classifier for the training set. This set of parameters is referred to as an individual of the population in the adapted genetic algorithm described in Section 3.4. The MSE is calculated for all individuals of the population. The individuals of the population and their MSEs are used in the adapted genetic algorithm, which helps to achieve the target of optimizing a MAKER-based classifier for sepsis diagnosis, which maximizes the predicted outputs, i.e., predicted probabilities, of the classifier for the true observed outputs of the training set to minimize the MSE for the training set.

As stated previously, an individual of the population in the adapted genetic algorithm is composed of the referential values of five input variables and the weights of the evidential elements of all pieces of evidence, each of which contains a referential value, of the five input variables. Among the weights of the individuals of the population in the adapted genetic algorithm, the weights of the evidential elements of the five pieces of evidence from the different input variables have an impact on the updated weight of each activated belief rule for an observation used to predict this observation. Thus, we can maximize the predicted outputs, i.e., predicted probabilities, of the classifier for the true observed outputs of the training

set to minimize the MSE for the training set by optimizing the referential values of the five input variables and the weights of the evidential elements. The optimal individual (solution) of the population acquired from the optimization based on the adapted genetic algorithm of a MAKER-based classifier for the sepsis diagnosis can make the predicted outputs, i.e., predicted probabilities, of the classifier as close to the true observed outputs of the training set as possible.

In this case study of sepsis diagnosis, as mentioned previously, we have used the optimized individual (solution) of the population obtained from the optimization, which has one optimized referential value. On the basis of the referential values and weights acquired from this optimized individual (solution), we can use the MAKER-based classifier established by the process described earlier in this section to make a prediction about the observation {158.0000, 619.4500, 120.1000, 123.8600, 32.5000}. The predicted probability of the sepsis class for this observation is 0.9553 and that of the non-sepsis class is 0.0447. In other words, if a patient's test results for CRP, IL6, IL10, PCT, and WCC are 158.0000, 619.4500, 120.1000, 123.8600, and 32.5000 respectively, the probability this patient has sepsis is 0.9553 and the probability this patient does not have sepsis is 0.0447. From the process used to establish the MAKER-based classifier for this case study of sepsis diagnosis, it is evident the rule-based inferential modelling and prediction approach used to do this is an interpretable approach integrating statistical analysis, belief rule-based inference, and machine learning.

## 6.4 Performance Comparative Analysis

In this section, a performance comparative analysis on the sepsis data set is carried out between the MAKER-based classifier constructed using the rule-based inferential modelling and prediction approach, and alternative classifiers including classification trees, discriminant analysis, logistic regression, the support vector machine (SVM), k-nearest neighbours (KNN), ensembles, naïve Bayes, and artificial

neural networks (ANN). As mentioned previously, the sepsis data set was partitioned into four folds for cross-validation. All of the four folds of the sepsis data set have similar class distributions, and observations belonging to the same class have similar probability density functions. Four training sets and their corresponding test sets were generated based on the four folds of the sepsis data set. We calculated the performance measures for each of the classifiers constructed on each training set as stated above. The performance measures include sensitivity (SEN), specificity (SPC), diagnostic accuracy (ACC), and the area under the receiver operating characteristic (ROC) curve (AUC). The SEN, SPC, and ACC were determined from the threshold value 0.5, if the predicted outputs of the classifier were probabilities. The SEN, SPC, and ACC were determined from the threshold value 0, if the predicted outputs were generated by the SVM classifiers.

**Table 6.8 Alternative variants of the classifiers other than the MAKER-based classifiers for the sepsis diagnosis**

| Classifier | Variants of classifier | Selected variant of classifier |
|---|---|---|
| decision trees | simple tree, medium tree, and complex tree | complex tree |
| discriminant analysis | linear discriminant and quadratic discriminant | quadratic discriminant |
| logistic regression | logistic regression | logistic regression |
| support vector machine (SVM) | linear SVM, quadratic SVM, cubic SVM, fine Gaussian SVM, medium Gaussian SVM, and coarse Gaussian SVM | fine Gaussian SVM |
| k-nearest neighbor (KNN) | fine KNN, medium KNN, coarse KNN, cosine KNN, cubic KNN, and weighted KNN | fine KNN and weighted KNN |
| ensembles | boosted trees, bagged trees, subspace discriminant, subspace KNN, and RUSBoosted trees | bagged trees and subspace KNN |
| naïve Bayes | naïve Bayes | naïve Bayes |
| artificial neural networks (ANN) | feed-forward backpropagation | feed-forward backpropagation |

The reason for using these threshold values was that all the classifiers for the sepsis diagnosis in this case study are binary classifiers.

On the basis of these performance measures calculated for each test set of the sepsis data set, we generated the average measures for each of the above classifiers to provide a comprehensive evaluation of each one's performance. It is worth noting that a number of alternative variants of each classifier were constructed based on each training set of the sepsis data set. According to the training accuracies of the variants, we selected the variant with the highest average accuracy for the classification of the training sets to represent each classifier.

Table 6.8 lists the alternative variants and the selected variant for each classifier except the MAKER-based classifiers, used for the sepsis diagnosis in this case study. In addition, according to the stopping criterion introduced in Section 4.5, the training for the MAKER-based classifier was stopped when there were five trained referential values for each of the input variables of the training sets generated from the sepsis data set. Among the trained MAKER-based classifiers used for the sepsis diagnosis, we selected the MAKER-based classifiers with one trained referential value for each of the input variables to represent the MAKER-based classifiers in the comparison with the alternative classifiers, as these MAKER-based classifiers had the highest average diagnostic accuracy among all the trained MAKER-based classifiers, for sepsis diagnosis. The performance measures of each of the classifiers selected are reported in Table 6.9.

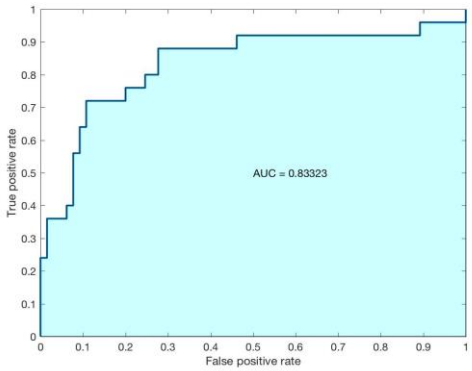
From Table 6.9, it is clear that, although the MAKER-based classifiers do not produce the optimal performance measures among the alternative classifiers for the sepsis diagnosis, they are generally near-optimal or near to the average (across all alternatives) performance measure. Specifically, the average sensitivity of the MAKER-based classifiers over the four test sets generated from the sepsis data set is 45.12%, which is above the average sensitivity of 41.29% across all the

classifiers and over the four test sets. The average specificity of the MAKER-based classifiers over the four test sets is 90%, which is very close to the average specificity of 90.87% across all classifiers over the four test sets. The average accuracy of the MAKER-based classifiers over the four test sets is 77.34%, which is close to the optimal accuracy of 80.67%. This suggests that the performance of the MAKER-based classifiers for the sepsis diagnosis is similar to that of alternative classifiers if we take the threshold values of 0.5 and 0 to generate the performance measures for the classifiers. The sensitivities, specificities, and accuracies presented in Table 6.9 were determined from those threshold values, and we could use other threshold values instead.

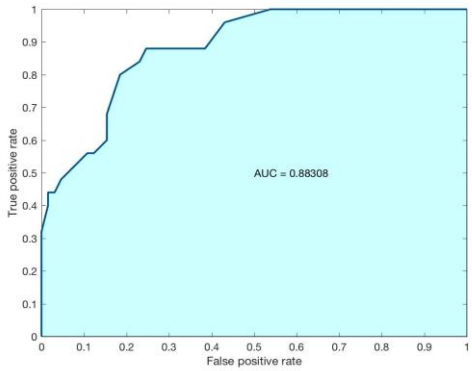
According to Ling, Huang and Zhang (2003), accuracy is a widely used measure for comparing the predictive capability of different classifiers. Most classifiers generate probability estimations of the classification, but they are completely ignored in the calculation of the accuracy (Ling, Huang and Zhang, 2003). Ling, Huang and Zhang (2003) present arguments emphasizing that the AUC provides a better measure than the accuracy. In the following part of this section, we present the ROC curve of the MAKER-based classifier and that of the classifier with the optimal AUC among all the alternative classifiers, for each of the test sets generated from the sepsis data set in Figure 6.3. We display the AUC of each classifier for each of the test sets, and the average AUC of each classifier over the four test sets, in Table 6.10. From Table 6.10, it can be observed that the average AUC of the MAKER-based classifiers over the four test sets is 0.8127, which is the fourth largest AUC among all the AUCs of the 11 classifiers for the sepsis diagnosis.

**Table 6.9 Performance measures of the classifiers for the sepsis diagnosis**

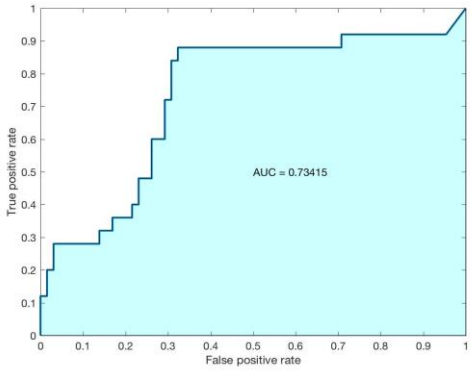
| classifiers/measures | SEN (%) | | | | | SPC (%) | | | | | ACC (%) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | tsts1 | tsts2 | tsts3 | tsts4 | avg. | tsts1 | tsts2 | tsts3 | tsts4 | avg. | tsts1 | tsts2 | tsts3 | tsts4 | avg. |
| MAKER | 56 | 36 | 50 | 38.46 | **45.12** | 92.31 | 84.62 | 87.69 | 95.38 | **90** | 82.22 | 71.11 | 76.92 | 79.12 | **77.34** |
| complex tree | 72 | 60 | 53.85 | 65.38 | 62.81 | 83.08 | 83.08 | 83.08 | 83.08 | 83.08 | 80 | 76.67 | 74.73 | 78.02 | 77.36 |
| quadratic discriminant | 36 | 28 | 42.31 | 30.77 | 34.27 | 95.38 | 98.46 | 93.85 | 96.92 | 96.15 | 78.89 | 78.89 | 79.12 | 78.02 | 78.73 |
| logistic regression | 36 | 28 | 50 | 42.31 | 39.08 | 93.85 | 95.38 | 92.31 | 95.38 | 94.23 | 77.78 | 76.67 | 80.22 | 80.22 | 78.72 |
| fine Gaussian SVM | 8 | 36 | 11.54 | 30.77 | 21.58 | 96.92 | 90.77 | 98.46 | 95.38 | 95.38 | 72.22 | 75.56 | 73.63 | 76.92 | 74.58 |
| fine KNN | 52 | 52 | 34.62 | 50 | 47.16 | 83.08 | 86.15 | 84.62 | 81.54 | 83.85 | 74.44 | 76.67 | 70.33 | 72.53 | 73.49 |
| weighted KNN | 36 | 40 | 34.62 | 30.77 | 35.35 | 89.23 | 93.85 | 92.31 | 87.69 | 90.77 | 74.44 | 78.89 | 75.82 | 71.43 | 75.15 |
| ensemble: bagged trees | 56 | 52 | 46.15 | 50 | 51.04 | 89.23 | 86.15 | 90.77 | 93.85 | 90 | 80 | 76.67 | 78.02 | 81.32 | 79.00 |
| ensemble: subspace KNN | 64 | 36 | 50 | 50 | 50 | 87.69 | 87.69 | 81.54 | 84.62 | 85.39 | 81.11 | 73.33 | 72.53 | 74.73 | 75.43 |
| naïve Bayes | 44 | 28 | 42.31 | 30.77 | 36.27 | 98.46 | 98.46 | 98.46 | 96.92 | 98.08 | 83.33 | 78.89 | 82.42 | 78.02 | 80.67 |
| ANN: feed-forward backpropagation | 48 | 32 | 19.23 | 26.92 | 31.54 | 92.31 | 86.15 | 96.92 | 95.38 | 92.69 | 80 | 71.11 | 74.73 | 75.82 | 75.42 |



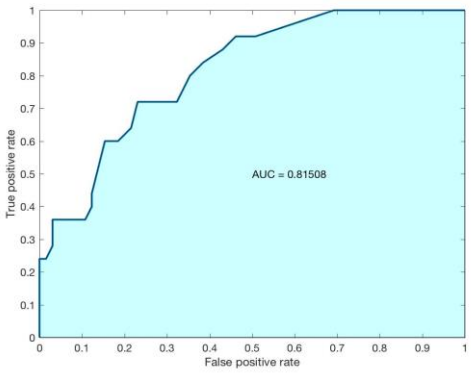

The ROC curve of MAKER classifier for the 1st test set



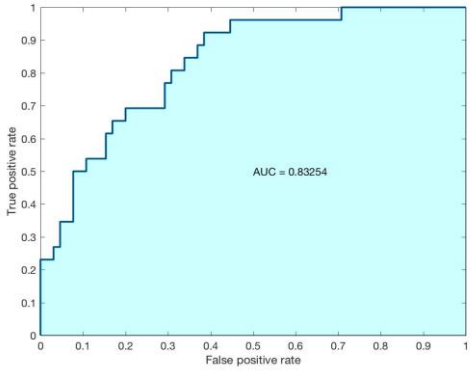

The ROC curve of the classifier of ensemble: bagged trees for the 1st test set



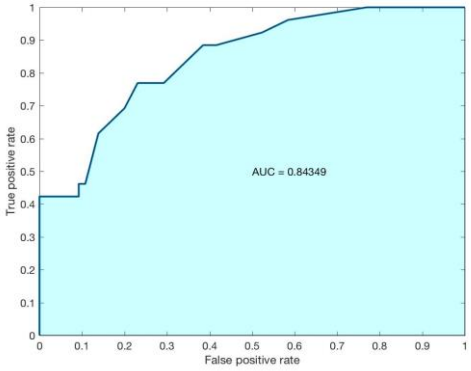

The ROC curve of MAKER classifier for the 2nd test set



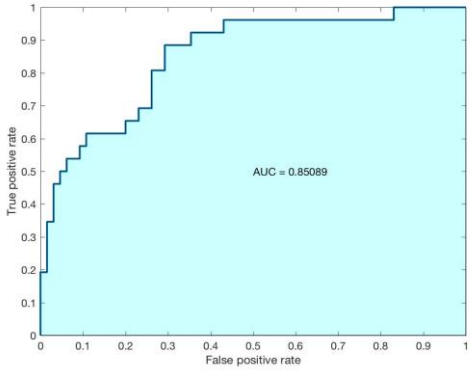

The ROC curve of the classifier of ensemble: bagged trees for the 2nd test set



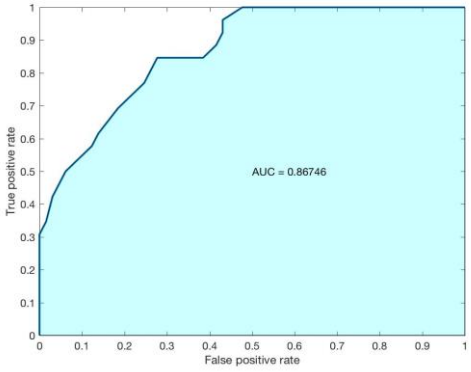

The ROC curve of MAKER classifier for the 3rd test set




The ROC curve of the classifier of ensemble: bagged trees for the 3rd test set




The ROC curve of MAKER classifier for the 4th test set




The ROC curve of the classifier of ensemble: bagged trees for the 4th test set

**Figure 6.3 The ROC curve of the MAKER-based classifier and that of the classifier which has the optimal AUC among all of the classifiers except the MAKER-based classifier for each of the test sets generated from the sepsis data set**

**Table 6.10 The area under the receiver operating characteristic (ROC) curve (AUC) of each of the classifiers for the sepsis diagnosis**

| classifiers/measures | AUC | | | | |
|---|---|---|---|---|---|
| | tsts1 | tsts2 | tsts3 | tsts4 | avg. |
| MAKER | 0.8332 | 0.7342 | 0.8325 | 0.8509 | **0.8127** |
| complex tree | 0.8129 | 0.7674 | 0.6962 | 0.7997 | 0.7690 |
| quadratic discriminant | 0.8486 | 0.7905 | 0.8284 | 0.7894 | 0.8142 |
| logistic regression | 0.8295 | 0.8462 | 0.8396 | 0.8089 | 0.8311 |
| fine Gaussian SVM | 0.7655 | 0.7508 | 0.7781 | 0.8284 | 0.7807 |
| fine KNN | 0.6754 | 0.6908 | 0.5962 | 0.6577 | 0.6550 |
| weighted KNN | 0.7828 | 0.7443 | 0.7651 | 0.7101 | 0.7506 |
| ensemble: bagged trees | 0.8831 | 0.8151 | 0.8435 | 0.8675 | 0.8523 |
| ensemble: subspace KNN | 0.8323 | 0.7120 | 0.8284 | 0.7902 | 0.7907 |
| naïve Bayes | 0.8452 | 0.7923 | 0.8249 | 0.7852 | 0.8119 |
| ANN: feed-forward backpropagation | 0.8782 | 0.7840 | 0.8095 | 0.7675 | 0.8098 |

According to Smithson and Merkle (2014), a general rule of thumb for using AUC to judge the classification capability of a classifier is that an AUC between 0.7 and 0.8 is considered acceptable, an AUC between 0.8 and 0.9 indicates excellent discrimination, and an AUC larger than 0.9 implies outstanding discrimination. As the average AUC of the MAKER-based classifiers as stated above is 0.8127, the MAKER-based classifiers for sepsis diagnosis can be considered excellent. The average AUC of the quadratic discriminant classifier over the four test sets is slightly larger than that of the MAKER-based classifier over the four test sets. In addition, the average AUC of the MAKER-based classifier is close to the average AUC of 0.8311 of the logistic regression classifiers over the four test sets and the average AUC of 0.8523 of the ensembles of bagged trees over the four test sets.

Thus, we can conclude that the performance of the MAKER-based classifiers for sepsis diagnosis is similar to that of the mainstream classifiers for sepsis diagnosis, e.g., naïve Bayes, quadratic discriminant, logistic regression, and ensemble: bagged trees, and the MAKER-based classifiers perform better than the complex tree, fine Gaussian SVM, fine KNN, weighted KNN, ensemble: subspace KNN, and

ANN: feed-forward backpropagation. As the MAKER-based classifiers are based on the belief rule-based inference and integrate statistical analysis, belief rule-based inference, and machine learning to generate predictions for sepsis diagnosis, they are essentially interpretable and hence a recommended tool to help doctors make reasonable diagnostic decisions about sepsis in their patients.

## 6.5 Summary

In summary, this chapter presents the application of the rule-based inferential modelling and prediction approach based on the system of the MAKER framework to sepsis diagnosis. At the beginning of this chapter, we introduced the data preparation, including data cleaning, data transformation, and data partition, for the cross-validation of the alternative classifiers on the sepsis data set. Then, from the perspectives of evidence acquisition from data, analysis of evidence independence, belief rule-based inference, and maximum likelihood prediction and machine learning, we described how the rule-based inferential modelling and prediction approach can be used to construct MAKER-based classifiers for sepsis diagnosis. Afterwards, we calculated the performance measures of sensitivity (SEN), specificity (SPC), diagnostic accuracy (ACC), and the area under the receiver operating characteristic (ROC) curve (AUC) for each classifier, for sepsis diagnosis, and we used these performance measures to compare the performance of the different classifiers in sepsis diagnosis. The results of the comparison show that the MAKER-based classifiers outperform 7 out of 10 alternative classifiers in sepsis diagnosis, and the performance of the MAKER-based classifiers is near-optimal, as the average AUC of the MAKER-based classifiers over the four test sets generated from the sepsis data set is close to that for the ensemble: bagged trees classifier over the four test sets, which is the optimal average AUC among all of the average AUCs of the alternative classifiers. Based on the analysis in this chapter, the MAKER-based classifier constructed using the rule-based inferential modelling and prediction approach, as an interpretable classifier, is a recommended tool for sepsis diagnosis.

# Chapter 7 Conclusions and Further Study

## 7.1 Conclusions

Sepsis is a serious disease that can cause death. It is important to evaluate patients' sepsis risk during diagnostic decisions within the early stages after the detection of the presence of symptoms that suggest sepsis. With the assessment of patients' sepsis risk, targeted antibiotic therapy can be used in the early stages to prevent sepsis from becoming worse and to effectively improve patients' prospects of survival. The conventional approach to sepsis diagnosis is blood culture, which may take several days. The approaches based on statistics and machine learning for sepsis diagnosis can be cheap, fast, and non-invasive. Hence, we can use these approaches to evaluate patients' sepsis risk in the early stages after the presence of suspicious symptoms of sepsis has been established. There are a wide variety of approaches based on statistics and machine learning that can be used for sepsis diagnosis, but these approaches have some issues, e.g. interpretability and overfitting, which may affect their performance in sepsis diagnosis.

In this research, we proposed a new approach, i.e. rule-based inferential modelling and prediction, to address some of the issues in the popular approaches to disease diagnosis. By applying the rule-based inferential modelling and prediction approach, we can acquire evidence directly from the data using statistical analysis, and combine multiple pieces of evidence from different input variables within the data to generate a belief rule base for inference. With the belief rule base, we can formulate the relationship between inputs and outputs using a unified inference scheme. For any given inputs, we can make an inference about the corresponding output of the inputs using the belief rule base and maximum likelihood prediction. With the algorithm of machine learning, we can optimise the parameters of the

inference model to ensure that the predicted probability of the output is as close to the probability of the true state of the output as possible. The findings for this approach are summarised as follows:

- By comparing the alternative approaches to disease diagnosis theoretically, we achieved research objective 3. The belief rule-base inference of the rule-based inferential modelling and prediction approach is totally transparent. Compared to the complex classifiers for disease diagnosis, e.g. ensemble, ANN, and random forest, the MAKER-based classifier established by the rule-based inferential modelling and prediction approach is more interpretable. It is essentially a white-box model in which the relationship between system inputs and outputs can be analysed clearly. In addition, the MAKER-based classifier is generally better than other interpretable classifiers, e.g. decision tree, logistic regression, and naïve Bayes.

- By comparing the performance of alternative approaches to sepsis diagnosis, we achieved research objective 4. The performance of the MAKER-based classifiers constructed by the rule-based inferential modelling and prediction approach for sepsis diagnosis is generally better than the majority of alternative models for sepsis diagnosis, and similar to the performance of ensemble: bagged trees, which is a complex model. The MAKER-based classifier is an outstanding classifier for classical data sets: the Banana data set, Haberman's survival data set, and the Iris data set, and it generally performs better than other interpretable classifiers, e.g. complex tree, logistic regression, and naïve Bayes. In addition, when it comes to the implications and research insights in the field of healthcare, as shown in Section 6.3.3, we can calculate the interdependence indices to find how input variables, i.e., CRP, IL6, IL10, PCT, and WCC, in the sepsis data set are dependent from each other. It has been shown in Section 6.3.5 that we can evaluate patients' sepsis risk based on the predicted probabilities generated from the MAKER-based models.

- By applying a referential-value-based data discretisation technique in the rule-based inferential modelling and prediction approach, we achieved research objective 5. Compared to other data-processing techniques, the referential-value-based data discretisation technique is closer to reality and better at reducing information loss and distortion, as well as better at presenting the characteristics of the data. The advantages of this data discretisation technique were shown in the function approximations (Chapter 4) and the classification experiments (Chapters 5 and 6).

- By applying the interdependence index in the rule-based inferential modelling and prediction approach, we achieved research objective 6. The MAKER-based models use an interdependence index to quantify the interdependence between input variables, while the naïve Bayes models assume that all input variables are independent of each other.

- By proposing an adapted genetic algorithm for the bilevel optimisation of the MAKER-based models, we achieved research objective 7. The function approximations in Chapter 4 and the classification experiments in Chapters 5 and 6 showed that this adapted genetic algorithm can work effectively to find the optimised solutions for the referential values and weights (reliabilities) of the MAKER-based models.

- By proposing the stopping criteria for the training process of the MAKER-based models, we achieved research objective 8. The functions approximations in Chapter 4 and the classification experiments in Chapters 5 and 6 showed that these stopping criteria can help us to find the optimal structure of the models based on the MAKER framework, which generally achieves balance between accuracy and complexity.

## 7.2 Further Study

Suggested directions of further study are summarised as follows:

The belief rule-base inference ensures that the MAKER-based models established by the rule-based inferential modelling and prediction approach are totally transparent and interpretable. However, the belief rule-base inference has a problem in terms of the high multiplicative complexity of the number of referential values of input variables in the belief rule base (Xu et al., 2017). In other words, the number of rules increases exponentially with the number of input variables and the number of referential values of each input variable (Yang and Xu, 2017). Thus, the number of parameters needed to training the model will increase exponentially (Yang and Xu, 2017), which will make the model extremely complex. A potential direction for further study on this issue could be to use hierarchical rule-base inference based on a hierarchical knowledge base composed of sub-rule bases.

As mentioned in section 4.4, in the bivariate function approximations of the MAKER-based models, the intervals between any two adjacent x-coordinate referential values or any two y-coordinate referential values in the data set for the approximations are set to 0.2. To improve the accuracy of the approximations of the MAKER-based models to the Himmelblau function, the abovementioned intervals can be narrowed to less than 0.2.

The results of the MAKER-based models for the classification experiments of the imbalanced data sets, i.e. the sepsis data set and Haberman's survival data set in this research, show that the sensitivity values of classification are generally much less than the specificity values of classification. This is partly due to the fact that the use of global performance measures guiding the learning process, such as the mean squared error (MSE), may provide an advantage to the majority class. One of the potential directions for further study on this issue could be to improve the

global performance measures, e.g. MSE guiding the learning process so as to treat all classes equally in the learning process.

In terms of the implications and research insights in the field of healthcare, further research will be focused on the combination of the approach of the rule-based inferential modelling and prediction and the medical knowledge. For example, we may extract the most frequent belief rules activated by the patients' records from the belief rule-base, and associate these activated belief rules with relevant medical knowledge to help healthcare professionals acquire a deeper understanding and more specific knowledge of disease diagnosis.

# References


Abe, S. (2005). Support Vector Machines for Pattern Classification. London: Springer, p.40.

Aggarwal, C. (2015). Data Classification: Algorithms and Applications. CRC Press, p.635.

Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L. and Herrera, F. (2011). KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing, 17, pp.255-287.

Alpaydin, E. (2010). Introduction to machine learning. Cambridge, Mass.: MIT Press.

Archdeacon, T. (1994). Correlation and Regression Analysis: A Historian's Guide. Madison, Wis.: University of Wisconsin Press, p.274.

Arikawa, S. and Motoda, H. (1998). Discovery Science: First international conference, DS'98, Fukuoka, Japan. New York: Springer.

Auria, L. and Moro, R. (2008). Support Vector Machines (SVM) as a Technique for Solvency Analysis. SSRN Electronic Journal.

Bousquet, O., Boucheron, S. and Lugosi, G. (2004). Introduction to Statistical Learning Theory. In: O. Bousquet, U. von Luxburg and G. Rätsch, ed., Advanced Lectures on Machine Learning. Berlin, Heidelberg: Springer, pp.169-207.

Brannen, J. (2005). Mixing Methods: The Entry of Qualitative and Quantitative Approaches into the Research Process. International Journal of Social Research Methodology, 8(3), pp.173-184.

Braspenning, P., Thuijsman, F. and Weijters, A. (1995). Artificial neural networks: An Introduction to ANN Theory and Practice. Berlin [etc.]: Springer, p.235.

Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. Lecture Notes in Computer Science, pp.164-178.

Chang, L., Zhou, Z., You, Y., Yang, L. and Zhou, Z. (2016). Belief rule based expert system for classification problems with new rule activation and weight calculation procedures. Information Sciences, 336, pp.75-91.

Chapman, A. (2005). Principles and methods of data cleaning. [Copenhagen]: GBIF, p.1.

Chen, Y., Chen, Y., Xu, X., Pan, C., Yang, J. and Yang, G. (2015). A data-driven approximate causal inference model using the evidential reasoning rule. Knowledge-Based Systems, 88, pp.264-272.

Chen, Y., Yang, J., Xu, D. and Yang, S. (2013). On the inference and approximation properties of belief rule based systems. Information Sciences, 234, pp.121-135.

Chen, Y., Yang, J., Xu, D. and Yang, S. (2013). On the inference and approximation properties of belief rule based systems. Information Sciences, 234, pp.121-135.

Chen, Y., Yang, J., Xu, D., Zhou, Z. and Tang, D. (2011). Inference analysis and adaptive training for belief rule based systems. Expert Systems with Applications, 38(10), pp.12845-12860.

Chomsky, N. (1980). Rules and representations. Behavioral and Brain Sciences, 3(01), p.1.

Coello Coello, C., Hernández Aguirre, A. and Zitzler, E. (2005). Evolutionary multi-criterion optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005, Proceedings. Berlin: Springer, p.754.

Cole, M. (2018). Hands-On Neural Network Programming with C#: Add powerful neural network capabilities to your C# enterprise applications. Packt Publishing Ltd, p.53.

Cooper, J. and Sartorius, N. (2013). A Companion to the Classification of Mental Disorders. OUP Oxford, p.55.

Cord, M. and Cunningham, P. (2008). Machine learning techniques for multimedia. Berlin: Springer, p.39.

Daniels, R. and Nutbeam, T. (2010). ABC of sepsis. Chichester, West Sussex: BMJ/ Wiley-Blackwell.

Dark, P., Blackwood, B., Gates, S., McAuley, D., Perkins, G., McMullan, R., Wilson, C., Graham, D., Timms, K. and Warhurst, G. (2014). Accuracy of LightCycler® SeptiFast for the detection and identification of pathogens in the blood of patients with suspected sepsis: a systematic review and meta-analysis. Intensive Care Medicine, 41(1), pp.21-33.

Della Riccia, G., Kruse, R. and Lenz, H. (2014). Computational Intelligence in Data Mining. Vienna: Springer Wien, p.60.

Dempster, A. (2008). The Dempster–Shafer calculus for statisticians. International Journal of Approximate Reasoning, 48(2), pp.365-377.

Denoeux, T. (1995). A k-nearest neighbor classification rule based on Dempster-Shafer theory. IEEE Transactions on Systems, Man, and Cybernetics, 25(5), pp.804-813.

Denoeux, T. (2000). A neural network classifier based on Dempster-Shafer theory. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 30(2), pp.131-150.

Dougherty, J., Kohavi, R. and Sahami, M. (1995). Supervised and Unsupervised Discretization of Continuous Features. Machine Learning Proceedings 1995, pp.194-202.

Dowla, F. and Rogers, L. (1995). Solving Problems in Environmental Engineering and Geosciences With Artificial Neural Networks. MIT Press, p.5.

Dua, S. and Chowriappa, P. (2013). Data mining for bioinformatics. Boca Raton, Fla.: CRC Press, p.115.

Dymova, L., Sevastianov, P. and Bartosiewicz, P. (2010). A new approach to the rule-base evidential reasoning: Stock trading expert system application. Expert Systems with Applications, 37(8), pp.5564-5576.

El-Gayar, N., Kittler, J. and Roli, F. (2010). Multiple Classifier Systems: 9th International Workshop, MCS 2010, Cairo, Egypt, April 7-9, 2010, Proceedings. Berlin: Springer Science & Business Media, p.104.

Fayyad, U. and Irani, K. (1993). Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In: The 13th International Joint Conference on Artificial Intelligence. pp.1022-1029.

Fisher, R. (1936). THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS. Annals of Eugenics, 7(2), pp.179-188.

Fisher, R. (1950). Contributions to mathematical statistics. New York: Wiley.

Freitas, A. and Lavington, S. (1996). Speeding up knowledge discovery in large relational databases by means of a new discretization algorithm. Lecture Notes in Computer Science, pp.124-133.

Geman, S., Bienenstock, E. and Doursat, R. (1992). Neural Networks and the Bias/Variance Dilemma. Neural Computation, 4(1), pp.1-58.

Graham, I. and Milne, R. (1991). Research and development in expert systems VIII: Proceedings of 11th Annual Technical Conference of the BCS Specialist Group, September 1991. Cambridge: Cambridge University Press on behalf of the British Computer Society, p.104.

Grosan, C. and Abraham, A. (2011). Rule-Based Expert Systems. Intelligent Systems Reference Library, pp.149-185.

Gupta, D. (2018). Applied Analytics through Case Studies Using SAS and R: Implementing Predictive Models and Machine Learning Techniques. Apress, p.235.

Haberman, S. (1976). Generalized Residuals for Log-Linear Models, Proceedings of the 9th International Biometrics Conference. In: 9th International Biometrics Conference. pp.104-122.

Hackeling, G. (2014). Mastering machine learning with scikit-learn. Birmingham, U.K.: Packt Publishing.

Hall, P. and Gill, N. (2018). An introduction to machine learning interpretability: An Applied Perspective on Fairness, Accountability, Transparency, and Explainable AI. 1st ed. Sebastopol, CA: O'Reilly Media.

Hand, D. (1992). Statistical methods in diagnosis. Statistical Methods in Medical Research, 1(1), pp.49-67.

Hanrahan, G. (2009). Modelling of pollutants in complex environmental systems. St. Albans, UK: ILM Publications, p.45.

Hastie, T., Friedman, J. and Tibshirani, R. (2013). The elements of statistical learning. New York [u.a.]: Springer.

Hayes, P. (1977). In defence of logic. In: Int. Joint Conf. Artificial Intelligence (IJCAI). pp.559–565.

Hodeghatta, U. and Nayak, U. (2016). Business Analytics Using R - A Practical Approach. Berkeley, CA: Apress, p.153.

Homenda, W. and Pedrycz, W. (2018). Pattern Recognition: A Quality of Data Perspective. Hoboken, NJ: John Wiley & Sons, p.78.

Holmes, D. and Jain, L. (2008). Innovations in Bayesian Networks: Theory and Applications. Berlin: Springer Science & Business Media, p.128.

Jackson, P. (1998). Introduction to expert systems. 3rd ed. Harlow: Addison-Wesley, p.2.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2017). An introduction to statistical learning. New York: Springer.

Jani, P. (2014). Business statistics: Theory and Applications. Delhi: PHI Learning, p.432.

Jiao, L., Pan, Q., Denœux, T., Liang, Y. and Feng, X. (2015). Belief rule-based classification system: Extension of FRBCS in belief functions framework. Information Sciences, 309, pp.26-49.

Kam, H. and Kim, H. (2017). Learning representations for the early detection of sepsis with deep neural networks. Computers in Biology and Medicine, 89, pp.248-255.

Karimi, H. (2014). Big Data: Techniques and Technologies in Geoinformatics.

Kerber, R. (1992). ChiMerge: discretization of numeric attributes. In: The Tenth National Conference on Artificial Intelligence. AAAI Press, pp.123-128.

Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. Artificial Intelligence in Medicine, 23(1), pp.89-109.

Kononenko, I. and Kukar, M. (2007). Machine learning and data mining. Elsevier, p.59.

Kong, G., Xu, D., Yang, J., Yin, X., Wang, T., Jiang, B. and Hu, Y. (2016). Belief rule-based inference for predicting trauma outcome. Knowledge-Based Systems, 95, pp.35-44.

Levy, M., Dellinger, R., Townsend, S., Linde-Zwirble, W., Marshall, J., Bion, J., Schorr, C., Artigas, A., Ramsay, G., Beale, R., Parker, M., Gerlach, H., Reinhart, K., Silva, E., Harvey, M., Regan, S. and Angus, D. (2010). The Surviving Sepsis Campaign: Results of an international guideline-based performance improvement program targeting severe sepsis*. Critical Care Medicine, 38(2), pp.367-374.

Ling, C., Huang, J. and Zhang, H. (2003). AUC: A Better Measure than Accuracy in Comparing Learning Algorithms. Advances in Artificial Intelligence, pp.329-341.

Liu, Z., Jerzy, K., Hua, X., Yuan, S., Dai, G. and Eugeniusz J., S. (2010). Mine safety and efficient exploitation facing challenges of the 21st century. Leiden, The Netherlands: CRC Press, p.153.

Lughofer, E. (2013). Evolving fuzzy systems - methodologies, advanced concepts and applications. [Place of publication not identified]: Springer-Verlag Berlin An, p.52.

Maimon, O. and Rokach, L. (2005). Decomposition Methodology for Knowledge Discovery and Data Mining: Theory and Applications. World Scientific, p.14.

Matignon, R. (2005). Neural network modeling using SAS Enterprise Miner. [Bloomington, IN]: Authorhouse, p.40.

Mani, S., Ozdas, A., Aliferis, C., Varol, H., Chen, Q., Carnevale, R., Chen, Y., Romano-Keeler, J., Nian, H. and Weitkamp, J. (2014). Medical decision support using machine learning for early detection of late-onset neonatal sepsis. Journal of the American Medical Informatics Association, 21(2), pp.326-336.

McLeod, S. (2017). Qualitative vs Quantitative Research. [online] Simplypsychology.org. Available at: https://www.simplypsychology.org/qualitative-quantitative.html [Accessed 21 Sep. 2018].

Mehta, R. (2017). Big Data Analytics with Java. Packt Publishing Ltd, p.161.

Michie, D., Spiegelhalter, D. and Taylor, C. (1994). Machine Learning, Neural and Statistical Classification. 1st ed.

Mittal, A. and Kassim, A. (2007). Bayesian Network Technologies: Applications and Graphical Models: Applications and Graphical Models. Hershey, PA: IGI Publishing, p.129.

Mohri, M., Rostamizadeh, A. and Talwalkar, A. (2012). Foundations of machine learning. Cambridge (EE. UU.): The MIT Press.

Molinaro, A., Simon, R. and Pfeiffer, R. (2005). Prediction error estimation: a comparison of resampling methods. Bioinformatics, 21(15), pp.3301-3307.

Moreira, J., Carvalho, A. and Horváth, T. (2018). A general introduction to data analytics. John Wiley & Sons, p.207.

Müller, A. and Guido, S. (2016). Introduction to Machine Learning with Python: A Guide for Data Scientists. p.27.

Nicolas, P. (2015). Scala for machine learning. Birmingham, UK: Packt Publishing, p.206.

Nilsson, N. (1984). Principles of artificial intelligence. Palo Alto, Calif.: Tioga publishing.

Oliver, P. (2010). Understanding the Research Process. London: Sage Publications, p.3.

Olson, D. and Delen, D. (2008). Advanced data mining techniques. Berlin: Springer, p.95.

Patil, P., Aghav, J. and Sareen, V. (2016). An Overview of Classi cation Algorithms and Ensemble Methods in Personal Credit Scoring. IJCST, 7(2), pp.183-188.

Pattnaik, P., Swetapadma, A. and Sarraf, J. (2018). Expert system techniques in biomedical science practice. IGI Global, p.xxi.

Punch, K. (2013). Introduction to social research. 3rd ed. SAGE.

Reinartz, T. (1999). Focusing Solutions for Data Mining: Analytical Studies and Experimental Results in Real-World Domains. Berlin: Springer Science & Business Media, p.131.

Richeldi, M. and Rossotto, M. (1995). Class-driven statistical discretization of continuous attributes (Extended abstract). Lecture Notes in Computer Science, pp.335-338.

Rokach, L. and Maimon, O. (2015). Data mining with decision trees. Singapore: World Scientific Pub. Co., pp.81-83.

Schweitzer, F. (2002). Modeling complexity in economic and social systems. New Jersey: World Scientific, p.56.

Sinha, A., Malo, P. and Deb, K. (2016). Evolutionary Bilevel Optimization: An Introduction and Recent Advances. Recent Advances in Evolutionary Multi-objective Optimization, pp.71-103.

Sinha, A., Malo, P. and Deb, K. (2018). A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications. IEEE Transactions on Evolutionary Computation, 22(2), pp.276-295.

Smithson, M. and Merkle, E. (2014). Generalized linear models for categorical and continuous limited dependent variables. Boca Raton: CRC Press, p.36.

Sun, R. (1995). Robust reasoning: integrating rule-based and similarity-based reasoning. Artificial Intelligence, 75(2), pp.241-295.

Suthaharan, S. (2016). Machine learning models and algorithms for big data classification. New York: Springer.

Tang, D., Yang, J., Chin, K., Wong, Z. and Liu, X. (2011). A methodology to generate a belief rule base for customer perception risk analysis in new product development. Expert Systems with Applications, 38(5), pp.5373-5383.

Tang, C., Middleton, P., Savkin, A., Chan, G., Bishop, S. and Lovell, N. (2010). Non-invasive classification of severe sepsis and systemic inflammatory response syndrome using a nonlinear support vector machine: a preliminary study. Physiological Measurement, 31(6), pp.775-793.

Taylor, R., Pare, J., Venkatesh, A., Mowafi, H., Melnick, E., Fleischman, W. and Hall, M. (2016). Prediction of In-hospital Mortality in Emergency Department Patients With Sepsis: A Local Big Data-Driven, Machine Learning Approach. Academic Emergency Medicine, 23(3), pp.269-278.

Tu, J. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. Journal of Clinical Epidemiology, 49(11), pp.1225-1231.

Tuffery, S. (2011). Data Mining and Statistics for Decision Making. John Wiley & Sons, p.477.

Vikhar, P. (2016). Evolutionary algorithms: A critical review and its future prospects. 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC).

Warhurst, G., Maddi, S., Dunn, G., Ghrew, M., Chadwick, P., Alexander, P., Bentley, A., Moore, J., Sharman, M., Carlson, G., Young, D. and Dark, P. (2014). Diagnostic accuracy of SeptiFast multi-pathogen real-time PCR in the setting of suspected healthcare-associated bloodstream infection. Intensive Care Medicine, 41(1), pp.86-93.

Wegner, T. (2010). Applied business statistics: Methods and Excel-Based Applications. Cape Town: Juta, p.504.

Welsh, A. (1996). Aspects of statistical inference. New York, N.Y: John Wiley & Sons, Inc, p.369.

Xu, D. (2011). An introduction and survey of the evidential reasoning approach for multiple criteria decision analysis. Annals of Operations Research, 195(1), pp.163-187.

Xu, X., Zheng, J., Yang, J., Xu, D. and Chen, Y. (2017). Data classification using evidence reasoning rule. Knowledge-Based Systems, 116, pp.144-151.

Yang, J. (2001). Rule and utility based evidential reasoning approach for multiattribute decision analysis under uncertainties. European Journal of Operational Research, 131(1), pp.31-61.

Yang, J. and Singh, M. (1994). An evidential reasoning approach for multiple-attribute decision making with uncertainty. IEEE Transactions on Systems, Man, and Cybernetics, 24(1), pp.1-18.

Yang, J. and Xu, D. (2013). Evidential reasoning rule for evidence combination. Artificial Intelligence, 205, pp.1-29.

Yang, J. and Xu, D. (2014). A Study on Generalising Bayesian Inference to Evidential Reasoning. Belief Functions: Theory and Applications, pp.180-189.

Yang, J. and Xu, D. (2017). Inferential modelling and decision making with data. 2017 23rd International Conference on Automation and Computing (ICAC).

Yang, J., Liu, J., Wang, J., Sii, H. and Wang, H. (2006). Belief rule-base inference methodology using the evidential reasoning Approach-RIMER. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 36(2), pp.266-285.

Yang, Y. and Webb, G. (2008). Discretization for naive-Bayes learning: managing discretization bias and variance. Machine Learning, 74(1), pp.39-74.

Zhang, C., Hu, Y., Chan, F., Sadiq, R. and Deng, Y. (2014). A new method to determine basic probability assignment using core samples. Knowledge-Based Systems, 69, pp.140-149.

Zhang, S., Zhang, C. and Yang, Q. (2003). Data preparation for data mining. Applied Artificial Intelligence, 17(5-6), pp.375-381.

Zhou, J. and Chen, F. (2018). Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent. Cham: Springer International Publishing, p.25.

# Appendices

## Appendix A:

## The Training Set of the First Fold of the Sepsis Data Set

| No. | CRP | IL6 | IL10 | PCT | WCC | Diagnosis |
|---|---|---|---|---|---|---|
| 1 | 80 | 17.66 | 4.1 | 0.12 | 13.2 | 1 |
| 2 | 58 | 724.73 | 63.41 | 1.26 | 15.09 | 1 |
| 3 | 86 | 173.19 | 139.13 | 1.13 | 12.2 | 1 |
| 4 | 193 | 43.27 | 10.28 | 2.15 | 1.9 | 1 |
| 5 | 217 | 122.79 | 4.03 | 0.34 | 12.2 | 1 |
| 6 | 374 | 260.54 | 4.62 | 7.18 | 9.3 | 1 |
| 7 | 516 | 1306.84 | 6.2 | 3.29 | 20.3 | 1 |
| 8 | 306 | 110.54 | 4.19 | 2.24 | 17.3 | 1 |
| 9 | 131 | 391.79 | 15.24 | 0.24 | 13.4 | 1 |
| 10 | 139 | 631.3 | 24.03 | 2.54 | 0.2 | 1 |
| 11 | 457 | 690.82 | 47.36 | 2.53 | 6.6 | 1 |
| 12 | 302 | 306.42 | 9.94 | 3.97 | 10.1 | 1 |
| 13 | 189 | 129.52 | 7.88 | 0.88 | 19.2 | 1 |
| 14 | 157.8 | 121.72 | 7.18 | 2.14 | 12.5 | 1 |
| 15 | 157 | 275.63 | 11.81 | 2.92 | 29.1 | 1 |
| 16 | 389 | 109.71 | 11.16 | 26.42 | 3 | 1 |
| 17 | 176 | 2134.32 | 33.13 | 8.45 | 9.3 | 1 |
| 18 | 305 | 796.35 | 31.38 | 15.86 | 24 | 1 |
| 19 | 458 | 210.31 | 13.68 | 16.15 | 12.7 | 1 |
| 20 | 201 | 226 | 12.41 | 20.84 | 16.7 | 1 |
| 21 | 206 | 217.99 | 20.38 | 7.68 | 35.8 | 1 |
| 22 | 135 | 815.49 | 53.16 | 0.79 | 0.5 | 1 |
| 23 | 267 | 101.9 | 345.81 | 3.52 | 6 | 1 |
| 24 | 176 | 403.8 | 99.51 | 0.55 | 12.3 | 1 |
| 25 | 158 | 619.45 | 120.1 | 123.86 | 32.5 | 1 |
| 26 | 287 | 544.91 | 88.17 | 163.22 | 13.2 | 1 |
| 27 | 30 | 27.36 | 1.94 | 0.29 | 5.4 | 2 |
| 28 | 21 | 2.72 | 1.64 | 0.17 | 4.4 | 2 |
| 29 | 13 | 7.7 | 3.37 | 0.05 | 9.9 | 2 |
| 30 | 11 | 19.78 | 6.28 | 0.45 | 10.6 | 2 |
| 31 | 10 | 4.04 | 1.8 | 0.41 | 19 | 2 |
| 32 | 35 | 4 | 12.48 | 0.05 | 3 | 2 |
| 33 | 31 | 13.13 | 10.02 | 0.06 | 9.8 | 2 |
| 34 | 22 | 5676.78 | 26.84 | 4.58 | 20.9 | 2 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 35 | 89 | 22.07 | 0.73 | 0.34 | 5.3 | 2 |
| 36 | 60 | 45.85 | 5.17 | 0.17 | 1 | 2 |
| 37 | 117 | 37.58 | 2.72 | 0.32 | 6.98 | 2 |
| 38 | 92 | 22.76 | 3.87 | 0.15 | 8.7 | 2 |
| 39 | 82 | 32.18 | 3.8 | 0.15 | 8.3 | 2 |
| 40 | 106.6 | 30.78 | 1 | 0.05 | 13.9 | 2 |
| 41 | 83 | 45.95 | 1.31 | 0.09 | 18.9 | 2 |
| 42 | 89 | 37.51 | 0.16 | 0.28 | 17.1 | 2 |
| 43 | 95 | 8.3 | 2.58 | 3.61 | 2.9 | 2 |
| 44 | 120 | 23.79 | 5.06 | 0.67 | 10.5 | 2 |
| 45 | 72 | 29.15 | 6.73 | 1.84 | 16 | 2 |
| 46 | 46 | 17.65 | 3.94 | 0.71 | 20.2 | 2 |
| 47 | 55 | 64.76 | 3.705 | 0.83 | 18.1 | 2 |
| 48 | 106 | 58.15 | 12.31 | 0.05 | 0.4 | 2 |
| 49 | 55 | 67.85 | 7.31 | 0.11 | 3.5 | 2 |
| 50 | 119.3 | 46.21 | 8.97 | 0.08 | 7.9 | 2 |
| 51 | 91 | 11.08 | 10.57 | 0.2 | 13.4 | 2 |
| 52 | 101 | 19.67 | 10.67 | 0.05 | 13.2 | 2 |
| 53 | 47 | 24.87 | 8.26 | 4.04 | 5.3 | 2 |
| 54 | 124 | 11.36 | 11.94 | 1.61 | 28.8 | 2 |
| 55 | 95 | 22.43 | 17.9 | 5.53 | 11.4 | 2 |
| 56 | 60 | 639.72 | 35.95 | 5.59 | 8.9 | 2 |
| 57 | 66 | 555.11 | 43.86 | 0.72 | 14.5 | 2 |
| 58 | 435 | 66.19 | 5.13 | 0.18 | 9.4 | 2 |
| 59 | 163 | 39.81 | 3.22 | 0.19 | 9.1 | 2 |
| 60 | 184.6 | 12.54 | 0.55 | 0.3 | 27.8 | 2 |
| 61 | 178 | 22.27 | 3.99 | 0.34 | 15.9 | 2 |
| 62 | 422 | 38.97 | 1.66 | 3.13 | 25.8 | 2 |
| 63 | 359 | 34.41 | 3.13 | 2.45 | 15.5 | 2 |
| 64 | 338 | 64.88 | 0.87 | 3.22 | 16.6 | 2 |
| 65 | 160 | 67.25 | 5.43 | 0.89 | 49.8 | 2 |
| 66 | 194 | 26.07 | 2.34 | 1.03 | 14.7 | 2 |
| 67 | 180 | 55.59 | 11.95 | 0.1 | 7.6 | 2 |
| 68 | 204.8 | 14.88 | 11.7 | 0.08 | 18.1 | 2 |
| 69 | 194 | 33.47 | 15.94 | 2.35 | 8.5 | 2 |
| 70 | 170 | 38.95 | 15.23 | 6.15 | 15.6 | 2 |
| 71 | 143 | 65.79 | 12.48 | 1.1 | 16.2 | 2 |
| 72 | 248 | 110.52 | 6.21 | 0.19 | 9.9 | 2 |
| 73 | 177 | 399.76 | 5.08 | 0.4 | 14.1 | 2 |
| 74 | 131 | 123.4 | 2.53 | 0.49 | 5.2 | 2 |
| 75 | 323 | 357.67 | 3.783 | 0.99 | 9.2 | 2 |
| 76 | 280 | 133.86 | 2.96 | 1.62 | 8.7 | 2 |
| 77 | 149 | 130.85 | 6.54 | 1.24 | 6.5 | 2 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 78 | 222.4 | 154.64 | 3.42 | 0.68 | 14.5 | 2 |
| 79 | 204 | 124.44 | 2.57 | 1.63 | 29 | 2 |
| 80 | 195 | 138.57 | 31.36 | 0.21 | 12.9 | 2 |
| 81 | 522 | 243.27 | 10.23 | 4.67 | 7.4 | 2 |
| 82 | 326 | 112.63 | 9.72 | 1.79 | 6 | 2 |
| 83 | 229 | 128.29 | 8.93 | 6.86 | 13 | 2 |
| 84 | 224 | 222.06 | 12.03 | 4.08 | 16.6 | 2 |
| 85 | 274 | 130.43 | 8.33 | 0.84 | 17.2 | 2 |
| 86 | 253 | 560.92 | 14.98 | 2.41 | 12.6 | 2 |
| 87 | 290 | 185.56 | 20.45 | 2.07 | 12.9 | 2 |
| 88 | 326 | 374.6 | 10.38 | 0.73 | 15.7 | 2 |
| 89 | 215 | 14124.19 | 45.05 | 38.44 | 2.2 | 2 |
| 90 | 364 | 163.76 | 12.55 | 17.34 | 35.2 | 2 |
| 91 | 690 | 153.5 | 13.468 | 46.62 | 20.7 | 2 |
| 92 | 125 | 19.82 | 2.04 | 0.17 | 15.6 | 1 |
| 93 | 66 | 5986.39 | 1230.3 | 7.63 | 22.8 | 1 |
| 94 | 46 | 4817.98 | 62.85 | 1.35 | 15.4 | 1 |
| 95 | 384 | 48.54 | 18.93 | 0.72 | 0.9 | 1 |
| 96 | 276 | 90.17 | 5.51 | 0.22 | 17 | 1 |
| 97 | 333 | 629.22 | 6.75 | 2.68 | 6.4 | 1 |
| 98 | 311 | 88.3 | 2.7 | 0.97 | 17.5 | 1 |
| 99 | 196 | 89.38 | 11.14 | 0.23 | 15.29 | 1 |
| 100 | 299 | 284.82 | 9.68 | 0.19 | 16.3 | 1 |
| 101 | 193 | 152.68 | 51.38 | 1.79 | 5.6 | 1 |
| 102 | 174 | 106.13 | 15.4 | 1.85 | 9.2 | 1 |
| 103 | 306 | 97.37 | 11.47 | 1.36 | 15.8 | 1 |
| 104 | 381 | 5533.12 | 24.88 | 4.45 | 27.7 | 1 |
| 105 | 133 | 325.22 | 8.83 | 0.8 | 19.6 | 1 |
| 106 | 229 | 286.43 | 10.3 | 0.93 | 12.3 | 1 |
| 107 | 404 | 293.95 | 30.02 | 69.47 | 3.6 | 1 |
| 108 | 128 | 3481.21 | 37.48 | 12.97 | 9.7 | 1 |
| 109 | 209 | 1751.1 | 42.89 | 8.4 | 14.1 | 1 |
| 110 | 230 | 394.52 | 9.08 | 10.04 | 22.4 | 1 |
| 111 | 275 | 7344.2 | 21.57 | 62.45 | 13.7 | 1 |
| 112 | 372 | 208 | 4563.87 | 0.65 | 0.1 | 1 |
| 113 | 271 | 1408.45 | 86.07 | 3.96 | 7.9 | 1 |
| 114 | 144 | 102.14 | 75.33 | 1.19 | 6.2 | 1 |
| 115 | 457 | 594.38 | 60.87 | 3.21 | 31.8 | 1 |
| 116 | 275 | 5038.86 | 366.85 | 200 | 17.63 | 1 |
| 117 | 336 | 193.61 | 120.57 | 10.24 | 12.4 | 1 |
| 118 | 2.9 | 8.93 | 5.27 | 0.33 | 0.1 | 2 |
| 119 | 11 | 0.82 | 0.19 | 0.05 | 5.4 | 2 |
| 120 | 39 | 25.38 | 5.41 | 0.13 | 7.6 | 2 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 121 | 40 | 2.42 | 3.8 | 0.05 | 9.4 | 2 |
| 122 | 4.2 | 14.73 | 0.87 | 0.08 | 14.5 | 2 |
| 123 | 5.7 | 6.49 | 6.03 | 0.14 | 14.5 | 2 |
| 124 | 31 | 36.15 | 8.88 | 0.05 | 1 | 2 |
| 125 | 20 | 71.89 | 13.07 | 0.47 | 9.6 | 2 |
| 126 | 16 | 262.79 | 16.92 | 0.75 | 17.4 | 2 |
| 127 | 70 | 18.37 | 3.23 | 0.15 | 1.7 | 2 |
| 128 | 47 | 42.83 | 5.76 | 0.05 | 0.3 | 2 |
| 129 | 61 | 20.29 | 3.66 | 0.27 | 6.7 | 2 |
| 130 | 60 | 29.57 | 2.62 | 0.22 | 6.6 | 2 |
| 131 | 107 | 58.64 | 1.61 | 0.33 | 23.5 | 2 |
| 132 | 116 | 3.98 | 2.02 | 0.37 | 14.4 | 2 |
| 133 | 41 | 5.27 | 2.96 | 0.45 | 18.9 | 2 |
| 134 | 68 | 13.98 | 6.59 | 0.77 | 5.7 | 2 |
| 135 | 68 | 7.03 | 5.28 | 0.48 | 8.9 | 2 |
| 136 | 76 | 21.22 | 3.37 | 2.21 | 13 | 2 |
| 137 | 118 | 17.33 | 1.79 | 0.56 | 15.4 | 2 |
| 138 | 52 | 5.81 | 3.53 | 1.47 | 31 | 2 |
| 139 | 64 | 67.24 | 7.7 | 0.26 | 0.2 | 2 |
| 140 | 76 | 22.11 | 7.29 | 0.31 | 7 | 2 |
| 141 | 65 | 67.09 | 8.7 | 0.27 | 21.77 | 2 |
| 142 | 103 | 70.68 | 14.96 | 0.08 | 11.6 | 2 |
| 143 | 87 | 66.64 | 46.61 | 1.35 | 5.9 | 2 |
| 144 | 71 | 33.48 | 10.62 | 0.49 | 0.3 | 2 |
| 145 | 110 | 39.9 | 48.09 | 1.14 | 11.7 | 2 |
| 146 | 47 | 314 | 25.54 | 1.08 | 9.9 | 2 |
| 147 | 52 | 179.06 | 9.46 | 1 | 15.3 | 2 |
| 148 | 125 | 155.98 | 18.63 | 1.43 | 13.5 | 2 |
| 149 | 138 | 29.23 | 3.406 | 0.19 | 9.5 | 2 |
| 150 | 245 | 14.42 | 1.33 | 0.45 | 20.6 | 2 |
| 151 | 241 | 69.99 | 2.275 | 0.18 | 18.4 | 2 |
| 152 | 194 | 31.05 | 4.004 | 0.34 | 19.1 | 2 |
| 153 | 143 | 41 | 1.87 | 1.73 | 16.2 | 2 |
| 154 | 206 | 23.95 | 1.87 | 0.72 | 12 | 2 |
| 155 | 154 | 51.46 | 3.5 | 4.23 | 13.9 | 2 |
| 156 | 314 | 44.08 | 4.71 | 3.39 | 16.1 | 2 |
| 157 | 129 | 63.07 | 4.381 | 2.12 | 11.7 | 2 |
| 158 | 147 | 23.66 | 8.29 | 0.18 | 7.7 | 2 |
| 159 | 234 | 66.93 | 7.36 | 0.28 | 18 | 2 |
| 160 | 305 | 13.81 | 17.74 | 0.91 | 6.4 | 2 |
| 161 | 270 | 70.51 | 16.42 | 1.14 | 12.7 | 2 |
| 162 | 328 | 71.25 | 38.7 | 2.38 | 22.5 | 2 |
| 163 | 376 | 99.53 | 1.68 | 0.24 | 9.6 | 2 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 164 | 307 | 369.12 | 4.381 | 0.37 | 12.9 | 2 |
| 165 | 318 | 142.8 | 2.33 | 1.29 | 0.2 | 2 |
| 166 | 273 | 147.38 | 4.86 | 1.82 | 9.9 | 2 |
| 167 | 220 | 111.27 | 2.353 | 2.86 | 10.2 | 2 |
| 168 | 196 | 147.25 | 2.43 | 1.44 | 8.8 | 2 |
| 169 | 348 | 435.27 | 3.82 | 4.31 | 22 | 2 |
| 170 | 428 | 106.6 | 4.07 | 2.08 | 24.1 | 2 |
| 171 | 207.5 | 392.57 | 9.54 | 0.39 | 17.2 | 2 |
| 172 | 200 | 265.31 | 8.69 | 0.62 | 8.4 | 2 |
| 173 | 197 | 139.49 | 15.72 | 0.49 | 9.7 | 2 |
| 174 | 246 | 265.91 | 45.05 | 4.77 | 20.6 | 2 |
| 175 | 436 | 1592.32 | 10.35 | 3.13 | 14.4 | 2 |
| 176 | 147 | 103.46 | 21.21 | 5.44 | 19.7 | 2 |
| 177 | 140 | 228.75 | 6.87 | 0.85 | 22.2 | 2 |
| 178 | 365 | 638.41 | 12.01 | 0.96 | 16.8 | 2 |
| 179 | 267 | 462.83 | 14.14 | 126.79 | 5.7 | 2 |
| 180 | 298 | 154.44 | 13.23 | 12.22 | 11.3 | 2 |
| 181 | 262 | 429.5 | 18.78 | 11.84 | 11.9 | 2 |
| 182 | 361 | 125.28 | 10.06 | 22.66 | 15.5 | 2 |
| 183 | 124 | 20.46 | 4.83 | 0.44 | 11 | 1 |
| 184 | 56 | 608.57 | 105.03 | 1.23 | 17.2 | 1 |
| 185 | 190 | 20.21 | 32.47 | 0.76 | 0.1 | 1 |
| 186 | 244 | 360.16 | 4.09 | 0.29 | 11.9 | 1 |
| 187 | 440 | 145.18 | 1.11 | 0.7 | 9 | 1 |
| 188 | 382 | 656.67 | 5.69 | 2.33 | 8.9 | 1 |
| 189 | 318 | 241.74 | 3.12 | 6.8 | 11.6 | 1 |
| 190 | 225.6 | 440.76 | 7.85 | 0.27 | 13.5 | 1 |
| 191 | 264 | 174.34 | 14.31 | 0.33 | 15.1 | 1 |
| 192 | 237 | 205.64 | 16.04 | 1.32 | 0 | 1 |
| 193 | 289 | 148.61 | 12.35 | 1.65 | 8 | 1 |
| 194 | 292 | 153.09 | 7.76 | 2.89 | 18.6 | 1 |
| 195 | 263 | 127.12 | 14.4 | 0.9 | 12.1 | 1 |
| 196 | 210 | 136.23 | 10.25 | 1.71 | 20.7 | 1 |
| 197 | 197 | 116.22 | 20.99 | 5.52 | 11.5 | 1 |
| 198 | 164 | 6256.87 | 27.35 | 16.7 | 0.1 | 1 |
| 199 | 356 | 1506.66 | 16.85 | 22.68 | 6.6 | 1 |
| 200 | 216 | 104.23 | 7.24 | 47.5 | 39.7 | 1 |
| 201 | 257 | 93.3 | 10.84 | 15.68 | 18.2 | 1 |
| 202 | 262 | 111.1 | 8.06 | 13.26 | 19.8 | 1 |
| 203 | 164 | 485.74 | 63.01 | 3.68 | 0.5 | 1 |
| 204 | 144 | 208 | 4489.32 | 4.28 | 10.9 | 1 |
| 205 | 243 | 113.7 | 308.96 | 1.35 | 16.2 | 1 |
| 206 | 229 | 1745.77 | 2593.81 | 6.25 | 28.1 | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 207 | 403 | 20971.01 | 2640.79 | 41.44 | 31.1 | 1 |
| 208 | 6.4 | 9.55 | 0.87 | 0.05 | 2.8 | 2 |
| 209 | 29 | 11.74 | 2.97 | 0.14 | 7.3 | 2 |
| 210 | 28 | 7.1 | 5.16 | 0.05 | 7.1 | 2 |
| 211 | 6 | 0.9 | 0.14 | 0.05 | 8.3 | 2 |
| 212 | 19 | 8.42 | 1.13 | 0.05 | 53.9 | 2 |
| 213 | 11 | 24.29 | 2.09 | 0.12 | 13.6 | 2 |
| 214 | 39 | 32.29 | 29.97 | 0.05 | 0.9 | 2 |
| 215 | 9 | 12.14 | 12.47 | 0.23 | 8.5 | 2 |
| 216 | 3.1 | 86.97 | 11.81 | 3.47 | 17.9 | 2 |
| 217 | 61 | 67.51 | 5.2 | 0.05 | 0.3 | 2 |
| 218 | 43 | 59.18 | 4.22 | 0.07 | 1 | 2 |
| 219 | 83 | 58.79 | 2.07 | 0.1 | 9.5 | 2 |
| 220 | 90 | 66.86 | 2.12 | 0.22 | 10.4 | 2 |
| 221 | 123 | 70.72 | 3.51 | 0.26 | 9.5 | 2 |
| 222 | 122 | 15.62 | 4.08 | 0.05 | 66 | 2 |
| 223 | 92 | 20.14 | 4.51 | 0.42 | 11.1 | 2 |
| 224 | 110 | 23.97 | 2.22 | 0.34 | 20.5 | 2 |
| 225 | 48 | 35.17 | 3.69 | 2.05 | 0.4 | 2 |
| 226 | 120 | 23.2 | 4.21 | 0.5 | 7 | 2 |
| 227 | 84 | 29.29 | 2.21 | 1.6 | 19.9 | 2 |
| 228 | 127 | 18.77 | 3.62 | 0.85 | 44.9 | 2 |
| 229 | 70 | 22.21 | 0.43 | 0.54 | 15.2 | 2 |
| 230 | 87 | 69.6 | 13.28 | 0.23 | 4.9 | 2 |
| 231 | 84 | 55.11 | 23.98 | 0.4 | 10.1 | 2 |
| 232 | 86 | 21.08 | 7.08 | 0.17 | 19 | 2 |
| 233 | 68 | 15.95 | 19.67 | 0.21 | 12.5 | 2 |
| 234 | 50 | 54.02 | 16.55 | 0.54 | 5.4 | 2 |
| 235 | 100 | 48.01 | 12.96 | 6.35 | 18.5 | 2 |
| 236 | 114 | 43.87 | 6.88 | 4.46 | 14.7 | 2 |
| 237 | 113 | 234.53 | 26.871 | 1.18 | 7.4 | 2 |
| 238 | 92 | 525.27 | 23.95 | 4.12 | 13.6 | 2 |
| 239 | 247 | 15.84 | 1.638 | 0.17 | 8.5 | 2 |
| 240 | 153 | 51.65 | 0.96 | 0.35 | 8.1 | 2 |
| 241 | 138 | 10.18 | 0.43 | 0.19 | 22.3 | 2 |
| 242 | 238 | 72.36 | 6.04 | 0.36 | 25.6 | 2 |
| 243 | 159 | 61.19 | 6.54 | 0.22 | 25.4 | 2 |
| 244 | 183 | 52.92 | 3.99 | 0.9 | 12.9 | 2 |
| 245 | 335 | 18.6 | 2.25 | 0.79 | 18.9 | 2 |
| 246 | 165 | 35.88 | 2.21 | 2.89 | 31.5 | 2 |
| 247 | 253 | 34.07 | 3.76 | 1.6 | 13.4 | 2 |
| 248 | 169 | 48.75 | 10.91 | 0.05 | 6.6 | 2 |
| 249 | 146 | 43.74 | 8.7 | 0.38 | 8.3 | 2 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 250 | 239 | 25.93 | 6.87 | 0.42 | 13.5 | 2 |
| 251 | 173 | 41.15 | 10.01 | 0.51 | 9.2 | 2 |
| 252 | 195 | 10.96 | 7.29 | 3.46 | 16.9 | 2 |
| 253 | 465 | 109.93 | 4.01 | 0.31 | 7.4 | 2 |
| 254 | 147 | 1001.76 | 2.57 | 0.23 | 7.6 | 2 |
| 255 | 273 | 628.5 | 5.668 | 0.19 | 14.5 | 2 |
| 256 | 315 | 446.24 | 5.99 | 1.16 | 2.8 | 2 |
| 257 | 208 | 738.43 | 6.162 | 0.96 | 8.8 | 2 |
| 258 | 203 | 313.81 | 4.28 | 3.95 | 7 | 2 |
| 259 | 349 | 592.78 | 1.91 | 0.57 | 9.9 | 2 |
| 260 | 236 | 106.46 | 6.12 | 1.24 | 13.6 | 2 |
| 261 | 137 | 192.4 | 31.67 | 0.34 | 23.8 | 2 |
| 262 | 251 | 189.09 | 10.51 | 0.24 | 26.7 | 2 |
| 263 | 173 | 110.04 | 35.31 | 2.64 | 7.9 | 2 |
| 264 | 217 | 141.26 | 8.3 | 0.51 | 7.1 | 2 |
| 265 | 265 | 194.95 | 7.59 | 0.73 | 11.2 | 2 |
| 266 | 343.4 | 271.56 | 43.42 | 4.79 | 11.1 | 2 |
| 267 | 468 | 235.68 | 7.56 | 3.08 | 15.3 | 2 |
| 268 | 372 | 906.38 | 11.43 | 1.44 | 11.2 | 2 |
| 269 | 431 | 362.32 | 9.2 | 7.55 | 29.3 | 2 |
| 270 | 240 | 2093.44 | 36.25 | 17.2 | 1.2 | 2 |
| 271 | 450 | 1057.01 | 28.32 | 200 | 12 | 2 |
| 272 | 256 | 1390.89 | 13.7 | 29.26 | 13.4 | 2 |

## Appendix B:

## The Belief Rule-base (the MAKER-based Model) for the Numerical Example in Chapter 6

| Rule No. | **If** the Values of Biomarkers of A Patient Are | | | | | **Then** the Probabilities of the Patient Having Sepsis or Not Are | |
|---|---|---|---|---|---|---|---|
| | CRP | IL6 | IL10 | PCT | WCC | Sepsis | Non-sepsis |
| 1 | 2.9 | 0.82 | 0.14 | 0.05 | 0 | 0.000385 | 0.999615 |
| 2 | 2.9 | 0.82 | 0.14 | 0.05 | 5.7066 | 0.005642 | 0.994358 |
| 3 | 2.9 | 0.82 | 0.14 | 0.05 | 66 | 0.008642 | 0.991358 |
| 4 | 2.9 | 0.82 | 0.14 | 9.8923 | 0 | 0.000152 | 0.999848 |
| 5 | 2.9 | 0.82 | 0.14 | 9.8923 | 5.7066 | 0.003131 | 0.996869 |
| 6 | 2.9 | 0.82 | 0.14 | 9.8923 | 66 | 0.004038 | 0.995962 |
| 7 | 2.9 | 0.82 | 0.14 | 200 | 0 | 0.587278 | 0.412722 |
| 8 | 2.9 | 0.82 | 0.14 | 200 | 5.7066 | 0.011127 | 0.988873 |
| 9 | 2.9 | 0.82 | 0.14 | 200 | 66 | 0.010971 | 0.989029 |
| 10 | 2.9 | 0.82 | 97.6625 | 0.05 | 0 | 0.004377 | 0.995623 |
| 11 | 2.9 | 0.82 | 97.6625 | 0.05 | 5.7066 | 0.051408 | 0.948592 |
| 12 | 2.9 | 0.82 | 97.6625 | 0.05 | 66 | 0.088019 | 0.911981 |
| 13 | 2.9 | 0.82 | 97.6625 | 9.8923 | 0 | 0.019699 | 0.980301 |
| 14 | 2.9 | 0.82 | 97.6625 | 9.8923 | 5.7066 | 0.019467 | 0.980533 |
| 15 | 2.9 | 0.82 | 97.6625 | 9.8923 | 66 | 0.027037 | 0.972963 |
| 16 | 2.9 | 0.82 | 97.6625 | 200 | 0 | 0.587593 | 0.412407 |
| 17 | 2.9 | 0.82 | 97.6625 | 200 | 5.7066 | 0.089497 | 0.910503 |
| 18 | 2.9 | 0.82 | 97.6625 | 200 | 66 | 0.089451 | 0.910549 |
| 19 | 2.9 | 0.82 | 4563.87 | 0.05 | 0 | 0.587674 | 0.412326 |
| 20 | 2.9 | 0.82 | 4563.87 | 0.05 | 5.7066 | 0.046369 | 0.953631 |
| 21 | 2.9 | 0.82 | 4563.87 | 0.05 | 66 | 0.046056 | 0.953944 |
| 22 | 2.9 | 0.82 | 4563.87 | 9.8923 | 0 | 0.587679 | 0.412321 |
| 23 | 2.9 | 0.82 | 4563.87 | 9.8923 | 5.7066 | 0.120671 | 0.879329 |
| 24 | 2.9 | 0.82 | 4563.87 | 9.8923 | 66 | 0.120759 | 0.879241 |
| 25 | 2.9 | 0.82 | 4563.87 | 200 | 0 | 0.587674 | 0.412326 |
| 26 | 2.9 | 0.82 | 4563.87 | 200 | 5.7066 | 0.045716 | 0.954284 |
| 27 | 2.9 | 0.82 | 4563.87 | 200 | 66 | 0.045398 | 0.954602 |
| 28 | 2.9 | 101.9637 | 0.14 | 0.05 | 0 | 0.072852 | 0.927148 |
| 29 | 2.9 | 101.9637 | 0.14 | 0.05 | 5.7066 | 0.045814 | 0.954186 |
| 30 | 2.9 | 101.9637 | 0.14 | 0.05 | 66 | 0.085296 | 0.914704 |
| 31 | 2.9 | 101.9637 | 0.14 | 9.8923 | 0 | 0.63938 | 0.36062 |
| 32 | 2.9 | 101.9637 | 0.14 | 9.8923 | 5.7066 | 0.116665 | 0.883335 |
| 33 | 2.9 | 101.9637 | 0.14 | 9.8923 | 66 | 0.16096 | 0.83904 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 34 | 2.9 | 101.9637 | 0.14 | 200 | 0 | 0.972972 | 0.027028 |
| 35 | 2.9 | 101.9637 | 0.14 | 200 | 5.7066 | 0.251805 | 0.748195 |
| 36 | 2.9 | 101.9637 | 0.14 | 200 | 66 | 0.225665 | 0.774335 |
| 37 | 2.9 | 101.9637 | 97.6625 | 0.05 | 0 | 0.691537 | 0.308463 |
| 38 | 2.9 | 101.9637 | 97.6625 | 0.05 | 5.7066 | 0.612189 | 0.387811 |
| 39 | 2.9 | 101.9637 | 97.6625 | 0.05 | 66 | 0.75006 | 0.24994 |
| 40 | 2.9 | 101.9637 | 97.6625 | 9.8923 | 0 | 0.983879 | 0.016121 |
| 41 | 2.9 | 101.9637 | 97.6625 | 9.8923 | 5.7066 | 0.743109 | 0.256891 |
| 42 | 2.9 | 101.9637 | 97.6625 | 9.8923 | 66 | 0.870792 | 0.129208 |
| 43 | 2.9 | 101.9637 | 97.6625 | 200 | 0 | 0.997178 | 0.002822 |
| 44 | 2.9 | 101.9637 | 97.6625 | 200 | 5.7066 | 0.914343 | 0.085657 |
| 45 | 2.9 | 101.9637 | 97.6625 | 200 | 66 | 0.904247 | 0.095753 |
| 46 | 2.9 | 101.9637 | 4563.87 | 0.05 | 0 | 0.587886 | 0.412114 |
| 47 | 2.9 | 101.9637 | 4563.87 | 0.05 | 5.7066 | 0.934717 | 0.065283 |
| 48 | 2.9 | 101.9637 | 4563.87 | 0.05 | 66 | 0.927295 | 0.072705 |
| 49 | 2.9 | 101.9637 | 4563.87 | 9.8923 | 0 | 0.58954 | 0.41046 |
| 50 | 2.9 | 101.9637 | 4563.87 | 9.8923 | 5.7066 | 0.996344 | 0.003656 |
| 51 | 2.9 | 101.9637 | 4563.87 | 9.8923 | 66 | 0.995868 | 0.004132 |
| 52 | 2.9 | 101.9637 | 4563.87 | 200 | 0 | 0.587862 | 0.412138 |
| 53 | 2.9 | 101.9637 | 4563.87 | 200 | 5.7066 | 0.907343 | 0.092657 |
| 54 | 2.9 | 101.9637 | 4563.87 | 200 | 66 | 0.897179 | 0.102821 |
| 55 | 2.9 | 20971.01 | 0.14 | 0.05 | 0 | 0.972525 | 0.027475 |
| 56 | 2.9 | 20971.01 | 0.14 | 0.05 | 5.7066 | 0.066417 | 0.933583 |
| 57 | 2.9 | 20971.01 | 0.14 | 0.05 | 66 | 0.087319 | 0.912681 |
| 58 | 2.9 | 20971.01 | 0.14 | 9.8923 | 0 | 0.99926 | 0.00074 |
| 59 | 2.9 | 20971.01 | 0.14 | 9.8923 | 5.7066 | 0.101359 | 0.898641 |
| 60 | 2.9 | 20971.01 | 0.14 | 9.8923 | 66 | 0.083837 | 0.916163 |
| 61 | 2.9 | 20971.01 | 0.14 | 200 | 0 | 0.968595 | 0.031405 |
| 62 | 2.9 | 20971.01 | 0.14 | 200 | 5.7066 | 0.341026 | 0.658974 |
| 63 | 2.9 | 20971.01 | 0.14 | 200 | 66 | 0.311119 | 0.688881 |
| 64 | 2.9 | 20971.01 | 97.6625 | 0.05 | 0 | 0.996807 | 0.003193 |
| 65 | 2.9 | 20971.01 | 97.6625 | 0.05 | 5.7066 | 0.680134 | 0.319866 |
| 66 | 2.9 | 20971.01 | 97.6625 | 0.05 | 66 | 0.75491 | 0.24509 |
| 67 | 2.9 | 20971.01 | 97.6625 | 9.8923 | 0 | 0.999864 | 0.000136 |
| 68 | 2.9 | 20971.01 | 97.6625 | 9.8923 | 5.7066 | 0.781121 | 0.218879 |
| 69 | 2.9 | 20971.01 | 97.6625 | 9.8923 | 66 | 0.858802 | 0.141198 |
| 70 | 2.9 | 20971.01 | 97.6625 | 200 | 0 | 0.994926 | 0.005074 |
| 71 | 2.9 | 20971.01 | 97.6625 | 200 | 5.7066 | 0.931557 | 0.068443 |
| 72 | 2.9 | 20971.01 | 97.6625 | 200 | 66 | 0.926228 | 0.073772 |
| 73 | 2.9 | 20971.01 | 4563.87 | 0.05 | 0 | 0.587861 | 0.412139 |
| 74 | 2.9 | 20971.01 | 4563.87 | 0.05 | 5.7066 | 0.940885 | 0.059115 |
| 75 | 2.9 | 20971.01 | 4563.87 | 0.05 | 66 | 0.9372 | 0.0628 |
| 76 | 2.9 | 20971.01 | 4563.87 | 9.8923 | 0 | 0.588664 | 0.411336 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 77 | 2.9 | 20971.01 | 4563.87 | 9.8923 | 5.7066 | 0.995915 | 0.004085 |
| 78 | 2.9 | 20971.01 | 4563.87 | 9.8923 | 66 | 0.995503 | 0.004497 |
| 79 | 2.9 | 20971.01 | 4563.87 | 200 | 0 | 0.587849 | 0.412151 |
| 80 | 2.9 | 20971.01 | 4563.87 | 200 | 5.7066 | 0.91665 | 0.08335 |
| 81 | 2.9 | 20971.01 | 4563.87 | 200 | 66 | 0.911737 | 0.088263 |
| 82 | 190.2799 | 0.82 | 0.14 | 0.05 | 0 | 0.345411 | 0.654589 |
| 83 | 190.2799 | 0.82 | 0.14 | 0.05 | 5.7066 | 0.036552 | 0.963448 |
| 84 | 190.2799 | 0.82 | 0.14 | 0.05 | 66 | 0.049088 | 0.950912 |
| 85 | 190.2799 | 0.82 | 0.14 | 9.8923 | 0 | 0.580788 | 0.419212 |
| 86 | 190.2799 | 0.82 | 0.14 | 9.8923 | 5.7066 | 0.044226 | 0.955774 |
| 87 | 190.2799 | 0.82 | 0.14 | 9.8923 | 66 | 0.060472 | 0.939528 |
| 88 | 190.2799 | 0.82 | 0.14 | 200 | 0 | 0.587786 | 0.412214 |
| 89 | 190.2799 | 0.82 | 0.14 | 200 | 5.7066 | 0.231375 | 0.768625 |
| 90 | 190.2799 | 0.82 | 0.14 | 200 | 66 | 0.2079 | 0.7921 |
| 91 | 190.2799 | 0.82 | 97.6625 | 0.05 | 0 | 0.949546 | 0.050454 |
| 92 | 190.2799 | 0.82 | 97.6625 | 0.05 | 5.7066 | 0.314671 | 0.685329 |
| 93 | 190.2799 | 0.82 | 97.6625 | 0.05 | 66 | 0.41478 | 0.58522 |
| 94 | 190.2799 | 0.82 | 97.6625 | 9.8923 | 0 | 0.987971 | 0.012029 |
| 95 | 190.2799 | 0.82 | 97.6625 | 9.8923 | 5.7066 | 0.374288 | 0.625712 |
| 96 | 190.2799 | 0.82 | 97.6625 | 9.8923 | 66 | 0.485895 | 0.514105 |
| 97 | 190.2799 | 0.82 | 97.6625 | 200 | 0 | 0.587844 | 0.412156 |
| 98 | 190.2799 | 0.82 | 97.6625 | 200 | 5.7066 | 0.863237 | 0.136763 |
| 99 | 190.2799 | 0.82 | 97.6625 | 200 | 66 | 0.849029 | 0.150971 |
| 100 | 190.2799 | 0.82 | 4563.87 | 0.05 | 0 | 0.587894 | 0.412106 |
| 101 | 190.2799 | 0.82 | 4563.87 | 0.05 | 5.7066 | 0.954234 | 0.045766 |
| 102 | 190.2799 | 0.82 | 4563.87 | 0.05 | 66 | 0.949694 | 0.050306 |
| 103 | 190.2799 | 0.82 | 4563.87 | 9.8923 | 0 | 0.589592 | 0.410408 |
| 104 | 190.2799 | 0.82 | 4563.87 | 9.8923 | 5.7066 | 0.997286 | 0.002714 |
| 105 | 190.2799 | 0.82 | 4563.87 | 9.8923 | 66 | 0.996962 | 0.003038 |
| 106 | 190.2799 | 0.82 | 4563.87 | 200 | 0 | 0.58787 | 0.41213 |
| 107 | 190.2799 | 0.82 | 4563.87 | 200 | 5.7066 | 0.820472 | 0.179528 |
| 108 | 190.2799 | 0.82 | 4563.87 | 200 | 66 | 0.828215 | 0.171785 |
| 109 | 190.2799 | 101.9637 | 0.14 | 0.05 | 0 | 0.607755 | 0.392245 |
| 110 | 190.2799 | 101.9637 | 0.14 | 0.05 | 5.7066 | 0.305586 | 0.694414 |
| 111 | 190.2799 | 101.9637 | 0.14 | 0.05 | 66 | 0.428551 | 0.571449 |
| 112 | 190.2799 | 101.9637 | 0.14 | 9.8923 | 0 | 0.798377 | 0.201623 |
| 113 | 190.2799 | 101.9637 | 0.14 | 9.8923 | 5.7066 | 0.568648 | 0.431352 |
| 114 | 190.2799 | 101.9637 | 0.14 | 9.8923 | 66 | 0.714976 | 0.285024 |
| 115 | 190.2799 | 101.9637 | 0.14 | 200 | 0 | 0.65311 | 0.34689 |
| 116 | 190.2799 | 101.9637 | 0.14 | 200 | 5.7066 | 0.207195 | 0.792805 |
| 117 | 190.2799 | 101.9637 | 0.14 | 200 | 66 | 0.499783 | 0.500217 |
| 118 | 190.2799 | 101.9637 | 97.6625 | 0.05 | 0 | 0.983723 | 0.016277 |
| 119 | 190.2799 | 101.9637 | 97.6625 | 0.05 | 5.7066 | 0.903934 | 0.096066 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 120 | 190.2799 | 101.9637 | 97.6625 | 0.05 | 66 | 0.935093 | 0.064907 |
| 121 | 190.2799 | 101.9637 | 97.6625 | 9.8923 | 0 | 0.96002 | 0.03998 |
| 122 | 190.2799 | 101.9637 | 97.6625 | 9.8923 | 5.7066 | 0.95534 | 0.04466 |
| 123 | 190.2799 | 101.9637 | 97.6625 | 9.8923 | 66 | 0.974974 | 0.025026 |
| 124 | 190.2799 | 101.9637 | 97.6625 | 200 | 0 | 0.777835 | 0.222165 |
| 125 | 190.2799 | 101.9637 | 97.6625 | 200 | 5.7066 | 0.904929 | 0.095071 |
| 126 | 190.2799 | 101.9637 | 97.6625 | 200 | 66 | 0.967355 | 0.032645 |
| 127 | 190.2799 | 101.9637 | 4563.87 | 0.05 | 0 | 0.993686 | 0.006314 |
| 128 | 190.2799 | 101.9637 | 4563.87 | 0.05 | 5.7066 | 0.979319 | 0.020681 |
| 129 | 190.2799 | 101.9637 | 4563.87 | 0.05 | 66 | 0.983169 | 0.016831 |
| 130 | 190.2799 | 101.9637 | 4563.87 | 9.8923 | 0 | 0.99973 | 0.00027 |
| 131 | 190.2799 | 101.9637 | 4563.87 | 9.8923 | 5.7066 | 0.997387 | 0.002613 |
| 132 | 190.2799 | 101.9637 | 4563.87 | 9.8923 | 66 | 0.99727 | 0.00273 |
| 133 | 190.2799 | 101.9637 | 4563.87 | 200 | 0 | 0.587852 | 0.412148 |
| 134 | 190.2799 | 101.9637 | 4563.87 | 200 | 5.7066 | 0.97173 | 0.02827 |
| 135 | 190.2799 | 101.9637 | 4563.87 | 200 | 66 | 0.977654 | 0.022346 |
| 136 | 190.2799 | 20971.01 | 0.14 | 0.05 | 0 | 0.805105 | 0.194895 |
| 137 | 190.2799 | 20971.01 | 0.14 | 0.05 | 5.7066 | 0.336976 | 0.663024 |
| 138 | 190.2799 | 20971.01 | 0.14 | 0.05 | 66 | 0.551817 | 0.448183 |
| 139 | 190.2799 | 20971.01 | 0.14 | 9.8923 | 0 | 0.543805 | 0.456195 |
| 140 | 190.2799 | 20971.01 | 0.14 | 9.8923 | 5.7066 | 0.600541 | 0.399459 |
| 141 | 190.2799 | 20971.01 | 0.14 | 9.8923 | 66 | 0.77827 | 0.22173 |
| 142 | 190.2799 | 20971.01 | 0.14 | 200 | 0 | 0.136863 | 0.863137 |
| 143 | 190.2799 | 20971.01 | 0.14 | 200 | 5.7066 | 0.318533 | 0.681467 |
| 144 | 190.2799 | 20971.01 | 0.14 | 200 | 66 | 0.547879 | 0.452121 |
| 145 | 190.2799 | 20971.01 | 97.6625 | 0.05 | 0 | 0.990087 | 0.009913 |
| 146 | 190.2799 | 20971.01 | 97.6625 | 0.05 | 5.7066 | 0.931917 | 0.068083 |
| 147 | 190.2799 | 20971.01 | 97.6625 | 0.05 | 66 | 0.965597 | 0.034403 |
| 148 | 190.2799 | 20971.01 | 97.6625 | 9.8923 | 0 | 0.861349 | 0.138651 |
| 149 | 190.2799 | 20971.01 | 97.6625 | 9.8923 | 5.7066 | 0.960368 | 0.039632 |
| 150 | 190.2799 | 20971.01 | 97.6625 | 9.8923 | 66 | 0.987335 | 0.012665 |
| 151 | 190.2799 | 20971.01 | 97.6625 | 200 | 0 | 0.307991 | 0.692009 |
| 152 | 190.2799 | 20971.01 | 97.6625 | 200 | 5.7066 | 0.935368 | 0.064632 |
| 153 | 190.2799 | 20971.01 | 97.6625 | 200 | 66 | 0.974648 | 0.025352 |
| 154 | 190.2799 | 20971.01 | 4563.87 | 0.05 | 0 | 0.988476 | 0.011524 |
| 155 | 190.2799 | 20971.01 | 4563.87 | 0.05 | 5.7066 | 0.972522 | 0.027478 |
| 156 | 190.2799 | 20971.01 | 4563.87 | 0.05 | 66 | 0.98123 | 0.01877 |
| 157 | 190.2799 | 20971.01 | 4563.87 | 9.8923 | 0 | 0.999507 | 0.000493 |
| 158 | 190.2799 | 20971.01 | 4563.87 | 9.8923 | 5.7066 | 0.995673 | 0.004327 |
| 159 | 190.2799 | 20971.01 | 4563.87 | 9.8923 | 66 | 0.995534 | 0.004466 |
| 160 | 190.2799 | 20971.01 | 4563.87 | 200 | 0 | 0.587844 | 0.412156 |
| 161 | 190.2799 | 20971.01 | 4563.87 | 200 | 5.7066 | 0.963471 | 0.036529 |
| 162 | 190.2799 | 20971.01 | 4563.87 | 200 | 66 | 0.976728 | 0.023272 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 163 | 690 | 0.82 | 0.14 | 0.05 | 0 | 0.994237 | 0.005763 |
| 164 | 690 | 0.82 | 0.14 | 0.05 | 5.7066 | 0.015692 | 0.984308 |
| 165 | 690 | 0.82 | 0.14 | 0.05 | 66 | 0.023481 | 0.976519 |
| 166 | 690 | 0.82 | 0.14 | 9.8923 | 0 | 0.999771 | 0.000229 |
| 167 | 690 | 0.82 | 0.14 | 9.8923 | 5.7066 | 0.021153 | 0.978847 |
| 168 | 690 | 0.82 | 0.14 | 9.8923 | 66 | 0.030448 | 0.969552 |
| 169 | 690 | 0.82 | 0.14 | 200 | 0 | 0.587701 | 0.412299 |
| 170 | 690 | 0.82 | 0.14 | 200 | 5.7066 | 0.152895 | 0.847105 |
| 171 | 690 | 0.82 | 0.14 | 200 | 66 | 0.134767 | 0.865233 |
| 172 | 690 | 0.82 | 97.6625 | 0.05 | 0 | 0.998851 | 0.001149 |
| 173 | 690 | 0.82 | 97.6625 | 0.05 | 5.7066 | 0.208573 | 0.791427 |
| 174 | 690 | 0.82 | 97.6625 | 0.05 | 66 | 0.255116 | 0.744884 |
| 175 | 690 | 0.82 | 97.6625 | 9.8923 | 0 | 0.999952 | 4.78E-05 |
| 176 | 690 | 0.82 | 97.6625 | 9.8923 | 5.7066 | 0.270409 | 0.729591 |
| 177 | 690 | 0.82 | 97.6625 | 9.8923 | 66 | 0.351452 | 0.648548 |
| 178 | 690 | 0.82 | 97.6625 | 200 | 0 | 0.587831 | 0.412169 |
| 179 | 690 | 0.82 | 97.6625 | 200 | 5.7066 | 0.806365 | 0.193635 |
| 180 | 690 | 0.82 | 97.6625 | 200 | 66 | 0.783847 | 0.216153 |
| 181 | 690 | 0.82 | 4563.87 | 0.05 | 0 | 0.587881 | 0.412119 |
| 182 | 690 | 0.82 | 4563.87 | 0.05 | 5.7066 | 0.904612 | 0.095388 |
| 183 | 690 | 0.82 | 4563.87 | 0.05 | 66 | 0.892555 | 0.107445 |
| 184 | 690 | 0.82 | 4563.87 | 9.8923 | 0 | 0.589831 | 0.410169 |
| 185 | 690 | 0.82 | 4563.87 | 9.8923 | 5.7066 | 0.994947 | 0.005053 |
| 186 | 690 | 0.82 | 4563.87 | 9.8923 | 66 | 0.994221 | 0.005779 |
| 187 | 690 | 0.82 | 4563.87 | 200 | 0 | 0.587853 | 0.412147 |
| 188 | 690 | 0.82 | 4563.87 | 200 | 5.7066 | 0.673492 | 0.326508 |
| 189 | 690 | 0.82 | 4563.87 | 200 | 66 | 0.677878 | 0.322122 |
| 190 | 690 | 101.9637 | 0.14 | 0.05 | 0 | 0.295757 | 0.704243 |
| 191 | 690 | 101.9637 | 0.14 | 0.05 | 5.7066 | 0.185142 | 0.814858 |
| 192 | 690 | 101.9637 | 0.14 | 0.05 | 66 | 0.27664 | 0.72336 |
| 193 | 690 | 101.9637 | 0.14 | 9.8923 | 0 | 0.715158 | 0.284842 |
| 194 | 690 | 101.9637 | 0.14 | 9.8923 | 5.7066 | 0.327258 | 0.672742 |
| 195 | 690 | 101.9637 | 0.14 | 9.8923 | 66 | 0.36346 | 0.63654 |
| 196 | 690 | 101.9637 | 0.14 | 200 | 0 | 0.899136 | 0.100864 |
| 197 | 690 | 101.9637 | 0.14 | 200 | 5.7066 | 0.090682 | 0.909318 |
| 198 | 690 | 101.9637 | 0.14 | 200 | 66 | 0.110602 | 0.889398 |
| 199 | 690 | 101.9637 | 97.6625 | 0.05 | 0 | 0.946372 | 0.053628 |
| 200 | 690 | 101.9637 | 97.6625 | 0.05 | 5.7066 | 0.815642 | 0.184358 |
| 201 | 690 | 101.9637 | 97.6625 | 0.05 | 66 | 0.892554 | 0.107446 |
| 202 | 690 | 101.9637 | 97.6625 | 9.8923 | 0 | 0.925821 | 0.074179 |
| 203 | 690 | 101.9637 | 97.6625 | 9.8923 | 5.7066 | 0.895763 | 0.104237 |
| 204 | 690 | 101.9637 | 97.6625 | 9.8923 | 66 | 0.917381 | 0.082619 |
| 205 | 690 | 101.9637 | 97.6625 | 200 | 0 | 0.961594 | 0.038406 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 206 | 690 | 101.9637 | 97.6625 | 200 | 5.7066 | 0.729681 | 0.270319 |
| 207 | 690 | 101.9637 | 97.6625 | 200 | 66 | 0.832643 | 0.167357 |
| 208 | 690 | 101.9637 | 4563.87 | 0.05 | 0 | 0.994804 | 0.005196 |
| 209 | 690 | 101.9637 | 4563.87 | 0.05 | 5.7066 | 0.969331 | 0.030669 |
| 210 | 690 | 101.9637 | 4563.87 | 0.05 | 66 | 0.970189 | 0.029811 |
| 211 | 690 | 101.9637 | 4563.87 | 9.8923 | 0 | 0.999778 | 0.000222 |
| 212 | 690 | 101.9637 | 4563.87 | 9.8923 | 5.7066 | 0.997247 | 0.002753 |
| 213 | 690 | 101.9637 | 4563.87 | 9.8923 | 66 | 0.997051 | 0.002949 |
| 214 | 690 | 101.9637 | 4563.87 | 200 | 0 | 0.587854 | 0.412146 |
| 215 | 690 | 101.9637 | 4563.87 | 200 | 5.7066 | 0.956968 | 0.043032 |
| 216 | 690 | 101.9637 | 4563.87 | 200 | 66 | 0.958458 | 0.041542 |
| 217 | 690 | 20971.01 | 0.14 | 0.05 | 0 | 0.055135 | 0.944865 |
| 218 | 690 | 20971.01 | 0.14 | 0.05 | 5.7066 | 0.183821 | 0.816179 |
| 219 | 690 | 20971.01 | 0.14 | 0.05 | 66 | 0.36706 | 0.63294 |
| 220 | 690 | 20971.01 | 0.14 | 9.8923 | 0 | 0.024516 | 0.975484 |
| 221 | 690 | 20971.01 | 0.14 | 9.8923 | 5.7066 | 0.3919 | 0.6081 |
| 222 | 690 | 20971.01 | 0.14 | 9.8923 | 66 | 0.550473 | 0.449527 |
| 223 | 690 | 20971.01 | 0.14 | 200 | 0 | 0.049723 | 0.950277 |
| 224 | 690 | 20971.01 | 0.14 | 200 | 5.7066 | 0.102182 | 0.897818 |
| 225 | 690 | 20971.01 | 0.14 | 200 | 66 | 0.174508 | 0.825492 |
| 226 | 690 | 20971.01 | 97.6625 | 0.05 | 0 | 0.571554 | 0.428446 |
| 227 | 690 | 20971.01 | 97.6625 | 0.05 | 5.7066 | 0.840769 | 0.159231 |
| 228 | 690 | 20971.01 | 97.6625 | 0.05 | 66 | 0.929877 | 0.070123 |
| 229 | 690 | 20971.01 | 97.6625 | 9.8923 | 0 | 0.06993 | 0.93007 |
| 230 | 690 | 20971.01 | 97.6625 | 9.8923 | 5.7066 | 0.9435 | 0.0565 |
| 231 | 690 | 20971.01 | 97.6625 | 9.8923 | 66 | 0.98077 | 0.01923 |
| 232 | 690 | 20971.01 | 97.6625 | 200 | 0 | 0.095133 | 0.904867 |
| 233 | 690 | 20971.01 | 97.6625 | 200 | 5.7066 | 0.818333 | 0.181667 |
| 234 | 690 | 20971.01 | 97.6625 | 200 | 66 | 0.932999 | 0.067001 |
| 235 | 690 | 20971.01 | 4563.87 | 0.05 | 0 | 0.990379 | 0.009621 |
| 236 | 690 | 20971.01 | 4563.87 | 0.05 | 5.7066 | 0.968364 | 0.031636 |
| 237 | 690 | 20971.01 | 4563.87 | 0.05 | 66 | 0.974095 | 0.025905 |
| 238 | 690 | 20971.01 | 4563.87 | 9.8923 | 0 | 0.999588 | 0.000412 |
| 239 | 690 | 20971.01 | 4563.87 | 9.8923 | 5.7066 | 0.996005 | 0.003995 |
| 240 | 690 | 20971.01 | 4563.87 | 9.8923 | 66 | 0.995825 | 0.004175 |
| 241 | 690 | 20971.01 | 4563.87 | 200 | 0 | 0.587845 | 0.412155 |
| 242 | 690 | 20971.01 | 4563.87 | 200 | 5.7066 | 0.956892 | 0.043108 |
| 243 | 690 | 20971.01 | 4563.87 | 200 | 66 | 0.965653 | 0.034347 |